EDB Postgres™ Advanced Server Release Notes

Version 11

August 12, 2019

EDB Postgres Advanced Server Release Notes, Version 11
by EnterpriseDB® Corporation
Copyright © 2018 - 2019 EnterpriseDB Corporation. All rights reserved.

Table of Contents

1 Introduction

With this release of EDB Postgres Advanced Server 11, EnterpriseDB continues its leadership as the only worldwide company to deliver innovative and low cost open source derived database solutions with commercial quality, ease of use, compatibility, scalability, and performance for small or large-scale enterprises. EDB Postgres Advanced Server 11 is built on the open source PostgreSQL 11.

EDB Postgres Advanced Server 11 adds a number of new features that will please developers and DBAs alike, including:

- Data redaction
- Autonomous transactions
- Performance diagnostics
- Various performance improvements to OCI dblink

These release notes are applicable to the Advanced Server version 11 release.

1.1 Installers and Documentation

EDB Postgres Advanced Server v11 is packaged and delivered as interactive installers for Windows; visit the EnterpriseDB website:

https://www.enterprisedb.com/advanced-downloads

RPM Packages are available for Linux from:

https://yum.enterprisedb.com/

If you need to request repository access, visit:

https://www.enterprisedb.com/repository-access-request

Documentation is provided on the EnterpriseDB website. Visit:

https://www.enterprisedb.com/resources/product-documentation

1.1.1 Supported Platforms

EDB Postgres Advanced Server v11 supports 64 bit Linux and Windows server platforms.

The Advanced Server 11 RPM packages are supported on the following platforms:

64 bit Linux:

- Red Hat Enterprise Linux (x86_64) 6.x and 7.x
- CentOS (x86_64) 6.x and 7.x
- PPC-LE 8 running RHEL or CentOS 7.x
- SLES 12

The Advanced Server 11 native packages (expected in early December 2018) are supported on the following platforms:

64 bit Linux:

- Debian 9
- Ubuntu 18.04 LTS

Graphical installers are supported on the following 64 bit Windows platforms:

- Windows Server 2016
- Windows Server 2012 R2 Server

Note: Connectors Installer will be supported on Windows 7, 8, & 10

Additional information about supported platforms is available on the EnterpriseDB website:

http://www.enterprisedb.com/ppas-platform-support

1.1.2 Component Certification

The following components have been tested with EBD Postgres Advanced Server V11:

- Procedural Language Packs – PL/Perl 5.26, PL/Python 3.6, PL/TCL 8.6
- CloneSchema 1.8
- pgAgent 4.0
- Slony 2.2.7
- Connectors 11.0.1
- JDBC 42.2.5.1, ODBC 10.03.0000.02, .NET 4.0.2.1, OCI 11.0.1.1
- pgAdmin 4 Client 2.0
- pgBouncer 1.9.0.1
- pgPool-II & pgPool-II Extensions 3.7.5
- MTK 52.0.0
- EDBPlus 37.0.0
- PostGIS 2.5.x

2 EDB Postgres Advanced Server v11 Features

EnterpriseDB features added for Advanced Server Version 11 include:

- Advanced Server no longer creates the `dbo` system catalog in its databases.
- Advanced Server now supports data redaction. Data redaction is a technique for limiting the exposure of sensitive data to certain users. Data redaction results in the alteration of the displayed data to such users. This is accomplished by defining redaction policies for tables with sensitive data.
- You can use the `edb_filter_log.redact_password_commands` extension to instruct the server to redact stored passwords from the log file.
- Advanced Server now supports EDB wait states, which is a background worker that probes each running session at a regular interval collecting information such as the database connection, the logged in user, the running query, and the wait events.
- Advanced Server now permits an infinite number of tries for custom plan generation if you set the `edb_custom_plan_tries` parameter to `-1`.
- The output format of the `version()` function has been changed to appear more consistent with the PostgreSQL community version output.
- Advanced Server now supports SPL standalone procedure overloading. Note that this feature is not compatible with Oracle databases.
- Advanced Server now supports the `PRAGMA AUTONOMOUS_TRANSACTION` directive within any SPL block to provide the autonomous transaction capability.
- Advanced server now offers performance improvements to libpq dblink and the OCI.

For information about Advanced Server features that are compatible with Oracle databases, see the following guides:

- Database Compatibility for Oracle Developer's Guide
- Database Compatibility for Oracle Developers Reference Guide
- Database Compatibility for Oracle Developers Tools and Utilities Guide
- Database Compatibility for Oracle Developers Built-in Package Guide

2.1 Community PostgreSQL 11 Updates

Advanced Server 11 integrates all of the community PostgreSQL 11 features. To review a complete list of changes to the community PostgreSQL project and the contributors names, see the *PostgreSQL 11 Release Notes* at:

https://www.postgresql.org/docs/11/static/release-11.html.

The following updates are available in PostgreSQL 11:

Partitioning Updates

- Allow faster partition elimination during query processing; this speeds access to partitioned tables with many partitions.
- Allow partition elimination during query execution.
- Allow the creation of partitions based on hashing a key.
- Allow updated rows to automatically move to new partitions based on the new row contents.
- Allow partitioned tables to have a default partition.
- Allow UNIQUE indexes on partitioned tables if the partition key guarantees uniqueness.
- Allow indexes on a partitioned table to be automatically created in any child partitions. The new command ALTER INDEX ATTACH PARTITION allows indexes to be attached to partitions. This does not behave as a global index since the contents are private to each index. WARN WHEN USING AN EXISTING INDEX?
- Allow foreign keys on partitioned tables.
- Allow INSERT, UPDATE, and COPY on partitioned tables to properly route rows to foreign partitions.
- This is supported by postgres_fdw foreign tables.
- Allow FOR EACH ROW triggers on partitioned tables.
- Allow equality joins between partitioned tables with identically partitioned child tables to join the child tables directly.
- Perform aggregation on each partition, and then merge the results.
- Allow postgres_fdw to push down aggregates to foreign tables that are partitions.

Parallel Queries

- Allow btree indexes to be built in parallel.
- Allow hash joins to be performed in parallel using a shared hash table.
- Allow UNION to run each SELECT in parallel if the individual SELECTs cannot be parallelized.
- Allow partition scans to more efficiently use parallel workers.
- Allow LIMIT to be passed to parallel workers.

- Allow single-evaluation queries, e.g. WHERE clause aggregate queries, and functions in the target list to be parallelized.
- Add server option parallel_leader_participation to control if the leader executes subplans.
- Allow parallelization of commands CREATE TABLE .. AS, SELECT INTO, and CREATE MATERIALIZED VIEW.
- Improve performance of sequential scans with many parallel workers.
- Add reporting of parallel worker sort activity to EXPLAIN.

Indexes

- Allow indexes to INCLUDE columns that are not part of the unique constraint but are available for index-only scans.
- This is also useful for including columns that don't have btree support.
- Remember the highest btree index page to optimize future monotonically increasing index additions.
- Allow entire hash index pages to be scanned.
- Previously for each hash index entry, we need to refind the scan position within the page. This cuts down on lock/unlock traffic.
- Add predicate locking for hash, GiST and GIN indexes.
- Allow heap-only-tuple (HOT) updates for expression indexes when the values of the expressions are unchanged.

SP-Gist

- Add TEXT prefix operator ^@ which is supported by SP-GiST.
- Allow polygons to be indexed with SP-GiST.
- Allow SP-GiST to use lossy representation of leaf keys.

Optimizer

- Improve the selection of the optimizer statistics' most-common-values.
- Improve selectivity estimates for >= and <= when the constants are not common values.
- Optimize var = var to var IS NOT NULL where equivalent.
- Improve row count optimizer estimates for EXISTS and NOT EXISTS queries.
- Add optimizer selectivity costs for HAVING clauses.

General Performance

- Add Just-in-Time (JIT) compilation of some parts of query plans to improve execution speed.
- Allow bitmap scans to perform index-only scans when possible.
- Update the free space map during vacuum.
- Allow vacuum to avoid unnecessary index scans.

- Improve performance of committing multiple concurrent transactions.
- Reduce memory usage for queries using set-returning functions in their target lists.
- Allow postgres_fdw to push UPDATEs and DELETEs using joins to foreign servers.

Monitoring

- Show memory usage in log_statement_stats, log_parser_stats, log_planner_stats, log_executor_stats.
- Add pg_stat_activity.backend_type now shows the type of background worker.
- Add bgw_type to the background worker C structure. This is displayed to the user in pg_stat_activity.backend_type and ps output.
- Have log_autovacuum_min_duration log skipped tables that are concurrently being dropped.

Information Schema

- Add information_schema columns related to table constraints and triggers.

Authentication

- Allow the server to specify more complex LDAP specifications in search+bind mode.
- Allow LDAP authentication to use ldaps.
- Improve LDAP logging of errors.

Permissions

- Add default roles which control file system access.
- Allow access to file system functions to be controlled by GRANT/REVOKE permissions, rather than superuser checks.
- Use GRANT/REVOKE to control access to lo_import() and lo_export().
- Compile-time option ALLOW_DANGEROUS_LO_FUNCTIONS has been removed.
- Use view owner not session owner when preventing non-password access to postgres_fdw tables.
- Fix invalid locking permission check in SELECT FOR UPDATE on views.

Server Configuration

- Add server setting ssl_passphrase_command to allow supplying of the passphrase for SSL key files.
- Add storage parameter toast_tuple_target to control the minimum length before TOAST storage will be considered for new rows.

- Allow server options related to memory and file sizes to be specified as number of bytes.

Write-Ahead Log (WAL)

- Allow the WAL file size to be set via initdb.
- No longer retain WAL that spans two checkpoints.
- Fill the unused portion of force-switched WAL segment files with zeros for improved compressibility.

Base Backup and Streaming Replication

- Replicate TRUNCATE activity when using logical replication.
- Pass prepared transaction information to logical replication subscribers.
- Exclude unlogged tables, temporary tables, and pg_internal.init files from streaming base backups.
- Allow heap pages checksums to be checked during streaming base backup.
- Allow replication slots to be advanced programmatically, rather than be consumed by subscribers.
- Add timeline information to the backup_label file.
- Add host and port connection information to the pg_stat_wal_receiver system view.

Window Functions

- Add window function features to complete SQL:2011 compliance.

Utility Commands

- Allow ALTER TABLE to add a column with a non-null default without a table rewrite.
- Allow views to be locked by locking the underlying tables.
- Allow ALTER INDEX to set statistics-gathering targets for expression indexes.
- In psql, \d+ now shows the statistics target for indexes.
- Allow multiple tables to be specified in one VACUUM or ANALYZE command. Also, if any table mentioned in VACUUM uses a column list, then the ANALYZE keyword must be supplied; previously, ANALYZE was implied in such cases.
- Add parenthesized options syntax to ANALYZE. This is similar to the syntax supported by VACUUM.
- Add CREATE AGGREGATE option to specify the behavior of the aggregate finalization function.

Data Types

- Allow the creation of arrays of domains.
- Support domains over composite types.
- Add casts from jsonb scalars to numeric and boolean data types.

Functions

- Add SHA-2 family of hash functions; specifically, sha224(), sha256(), sha384(), sha512() were added.
- Add support for 64-bit non-cryptographic hash functions.
- Allow to_char() and to_timestamp() to specify the time zone's hours and minutes from UTC.
- Improve the speed of aggregate computations.
- Add text search function websearch_to_tsquery() that supports a query syntax similar to that used by web search engines.
- Add function json(b)_to_tsvector() to create a text search query for matching JSON/JSONB values.

Server-Side Languages

- Add SQL-level procedures, which can start and commit their own transactions. They are created with the new CREATE PROCEDURE command and invoked via CALL. The new ALTER/DROP ROUTINE commands allows altering/dropping of procedures, functions, and aggregates.
- Add transaction control to PL/pgSQL, PL/Perl, PL/Python, PL/Tcl, and SPI server-side languages.
- Add the ability to define PL/pgSQL record types as not null, constant, or with initial values.
- Allow PL/pgSQL to handle changes to composite types (e.g. record, row) that happen between the first and later function executions in the same session. Previously such circumstances generated errors.
- Add extension jsonb_plpython to transform JSONB to/from PL/Python types.
- Add extension jsonb_plperl to transform JSONB to/from PL/Perl types.

Client Interface

- Change libpq to disable compression by default.
- Add DO CONTINUE action to the ECPG WHENEVER statement.
- Add ecpg mode to enable Oracle Pro*C handling of char arrays.
- This mode is enabled with -C.

Client Applications

psql

- Add psql command \gdesc to display the column names and types of the query output.
- Add psql variables to report query activity and errors.
- Allow psql to test for the existence of a variable.
- Add PSQL_PAGER to control psql's pager.
- Have psql \d+ always show the partition information.
- Have psql report the proper user name before the password prompt.
- Allow quit and exit to exit psql when used in an empty buffer.
- Have psql hint at using control-D when \q is entered alone on a line but ignored.
- Improve tab-completion for ALTER INDEX RESET/SET.
- Add infrastructure to allow psql to customize tab completion queries based on the server version.
- Previously tab completion queries could fail.

pgbench

- Add pgbench expressions support for NULLs, booleans, and some functions and operators.
- Add \if conditional support to pgbench.
- Allow the use of non-ASCII characters in pgbench variable names.
- Add pgbench option --init-steps to control the initialization steps performed.
- Add an approximated Zipfian-distributed random generator to pgbench.
- Allow the random seed to be set in pgbench.
- Allow pgbench to do exponentiation with pow() and power().
- Add hashing functions to pgbench.
- Make pgbench statistics more accurate when using --latency-limit and --rate.

Server Applications

- Add an option to pg_basebackup that creates a named replication slot. The option --create-slot creates the named replication slot (--slot) when the WAL streaming method (--wal-method=stream) is used.
- Allow initdb to set group read access to the data directory.
- Add pg_verify_checksums tool to verify database checksums while offline.
- Allow pg_resetwal to change the WAL segment size via --wal-segsize.
- Add long options to pg_resetwal and pg_controldata.
- Add pg_receivewal option --no-sync to prevent synchronous WAL writes, for testing.
- Add pg_receivewal option --endpos to specify when WAL receiving should stop.
- Allow pg_ctl to send the SIGKILL signal to processes.
- Reduce the number of files copied by pg_rewind.

- Prevent pg_rewind from running as root.

pg_dump, pg_dumpall, pg_restore

- Add pg_dumpall option --encoding to control encoding.
- Add pg_dump option --load-via-partition-root to force loading of data into the partition's root table, rather than the original partitions.
- Add an option to suppress dumping and restoring comments.

Source Code

- Add support for large pages on Windows.
- Add support for ARMv8 hardware CRC calculations.
- Convert documentation to DocBook XML.
- Use stdbool.h to define type bool on platforms where it's suitable.
- Add ability to use channel binding when using SCRAM authentication.
- Overhaul the way system tables are defined for bootstrap use.
- Allow background workers to attach to databases that normally disallow connections.
- Speed up lookups of built-in function names matching OIDs.
- Speed up construction of query results.
- Improve access speed to system caches.
- Add a generational memory allocator which is optimized for serial allocation/deallocation.
- Make the computation of system column pg_class.reltuples consistent.
- Update to use perltidy version.

Additional Modules

- Allow extension pg_prewarm to restore the previous shared buffer contents on startup.
- Add pg_trgm function strict_word_similarity() to compute the similarity of whole words.
- Allow creation of indexes on citext extension columns that can be used by LIKE comparisons.
- Allow btree_gin to index bool, bpchar, name and uuid data types.
- Allow cube and seg extensions using GiST indexes to perform index-only scans.
- Allow retrieval of negative cube coordinates using the ~> operator.
- Add Vietnamese letter detection to the unaccent extension.
- Enhance amcheck to check that each heap tuple has an index entry.
- Have adminpack use the new default file system access roles.
- Increase pg_stat_statement's query id to 64 bits.
- Install errcodes.txt to provide access to the error codes reported by PostgreSQL.
- Prevent extensions from creating custom server variables that take a quoted list of values.

- Removed contrib/start-scripts/osx.
- Removed the chkpass extension.

2.2 Deprecated Features

Please note that the following items will be deprecated:

- The PL/Java package is deprecated in EDB Postgres Advanced Server 11 and will be unavailable in EDB Postgres Advanced Server 12.

- Advanced Server no longer supports the Infinite Cache feature. All related components have been removed such as the functions, scripts, configuration parameters, and columns from statistical tables and views.

2.3 How to Report Problems

To report any issues you are having please contact EnterpriseDB's technical support staff:

- Email: support@enterprisedb.com

- Phone: +1-732-331-1320 or 1-800-235-5891 (US Only)