EDB Postgres™ Advanced Server

Version 10.1.5

November 14, 2017

EDB Postgres Advanced Server Release Notes, Version 10.1.5
by EnterpriseDB® Corporation

EnterpriseDB Corporation, 34 Crosby Drive Suite 100, Bedford, MA 01730, USA
T +1 781 357 3390 **F** +1 978 589 5701 **E** info@enterprisedb.com **www**.enterprisedb.com

Table of Contents

1 Introduction

With this release of EDB Postgres Advanced Server 10.0, 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 10.0 is built on the open source PostgreSQL 10.0, which introduces an impressive number of improvements that enable databases to scale up and scale out in more efficient ways. PostgreSQL 10.0 introduces Native Partitioning, Logical Replication, SCRAM Authentication, additional Parallel Query capabilities as well as a host of other new features and capabilities.

EDB Postgres Advanced Server 10.0 adds a number of new features that will delight developers and DBAs alike, including:

- Enhanced Partitioning features such as Hash Partitioning, Partition Management and additional Partition Pruning for performance.
- Customizable WAL Segment Size for performance optimization.
- New EDB Audit features.
- Additional Oracle compatibility.

These release notes are applicable to the 10.1.5 release on November 14, 2017.

2 EDB Postgres Advanced Server v10.0 Features

EnterpriseDB features added for Advanced Server Version 10 include:

- Improved Auditing:
    - Configurable by DDL and DML statement types
    - Configurable by User and Database
    - Audit logs can now be directed to syslog
- EDB Clone Schema:
    - Allows you to make a copy of an existing schema
- Customizable WAL Segment Size:
    - Option for initdb
    - Provides the capability to specify the WAL Segment size
- Automatic Prewarm
- The14th Generation of compatibility with the Oracle Database including the following:
    - Support for IN OUT Parameters for EXECUTE IMMEDIATE
    - OCI Aggregate Pushdown

For a summary of the newly added features that are compatible with Oracle databases, see the following documents:

- 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

This version also integrates all of the PostgreSQL v10.0 features, including:

- Declarative Partitioning
    - Replaces the use of table inheritance
    - Provides better performance
    - Support for list and range partitioning
    - Existing EPAS Partitioning has been fully integrated
- Logical Replication
    - Based on Logical Decoding
    - Adds configurable replication at table level granularity
- Parallel Query Phase 2
    - Parallel Index Scan
    - Parallel Bitmap Heap Scan
    - Parallel Merge Join
    - Parallel subplans
    - Extended language support for Parallel query
- SCRAM Authentication
- Durable Hash Indexes
- Executor Speedups
- ICU Collation Support
- FDW Aggregate Pushdown
- Improved Wait Event visibility
- Extended Statistics
- Support NEW and OLD tuplestores in AFTER triggers

2.1 Platform Support and System Requirements

EDB Postgres Advanced Server v10.0 supports 64 bit Linux and Windows server platforms. The Advanced Server 10 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

The Advanced Server 10 graphical (or interactive) installers are supported on the following platforms:

64 bit Linux:

- Red Hat Enterprise Linux 6.x and 7.x
- CentOS 6.x and 7.x
- Oracle Enterprise Linux 6.x and 7.x
- Ubuntu 14.04 LTS and 16.04 LTS
- Debian 7 and 8
- SELinux Enterprise 12.x

64 bit Windows:

- Windows Server 2016
- Windows Server 2012 R2 Server

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

Details on supported platforms is available on the EnterpriseDB website:

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

New platforms will be added as they become available. The next scheduled platform for v10.0 will be SLES 12.0.

2.2 Installers and Documentation

EDB Postgres Advanced Server v10.1.5 is packaged and delivered as a series of interactive installers available on the EnterpriseDB website. Visit:

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

RPM Packages are available for download from:

http://yum.enterprisedb.com/

Documentation is provided on the EnterpriseDB website. Visit:

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

2.3 Test Matrix

The following components have been tested with EBD Postgres Advanced Server V10.0

- Procedural Language Packs – PL/Perl 5.24, PL/Python 3.4, PL/TCL 8.6
- pgAgent 3.4.1
- Slony 2.2.6
- Connectors 10.0.0
- JDBC 9.4-1208, ODBC 9.05.0400, .NET 3.0.5, OCL 9.6.0.0
- pgAdmin 4 Client 2.0
- pgBouncer 1.7.2.1
- pgPool-II & pgPool-II Extensions 3.6.7
- MTK 51.0.0
- EDBPlus 36.0.0
- PostGIS support will be added when PostGIS versions supporting both 9.6 and 10.1 are made available

2.4 Incompatibilities

PostgeSQL 10.0 contains a number of changes that may affect compatibility with previous releases. They are published in the *PostgreSQL 10.0 Release Notes* - https://www.postgresql.org/docs/10/static/release-10.html - and listed here for convenience.

- Hash indexes must be rebuilt after pg_upgrade-ing from any previous major PostgreSQL version (Mithun Cy, Robert Haas, Amit Kapila)

  Major hash index improvements necessitated this requirement. pg_upgrade will create a script to assist with this.

- Rename write-ahead log directory pg_xlog to pg_wal, and rename transaction status directory pg_clog to pg_xact (Michael Paquier)

  Users have occasionally thought that these directories contained only inessential log files, and proceeded to remove write-ahead log files or transaction status files manually, causing irrecoverable data loss. These name changes are intended to discourage such errors in future.

- Rename SQL functions, tools, and options that reference “xlog” to “wal” (Robert Haas)

  For example, pg_switch_xlog() becomes pg_switch_wal(), pg_receivexlog becomes pg_receivewal, and --xlogdir becomes --waldir. This is for consistency with the change of the pg_xlog directory name; in general, the “xlog” terminology is no longer used in any user-facing places.

- Rename WAL-related functions and views to use lsn instead of location (David Rowley)

  There was previously an inconsistent mixture of the two terminologies.

- Change the implementation of set-returning functions appearing in a query's SELECT list (Andres Freund)

  Set-returning functions are now evaluated before evaluation of scalar expressions in the SELECT list, much as though they had been placed in a LATERAL FROM-clause item. This allows saner semantics for cases where multiple set-returning functions are present. If they return different numbers of rows, the shorter results are extended to match the longest result by adding nulls. Previously the results were cycled until they all terminated at the same time, producing a number of rows equal to the least common multiple of the functions' periods. In addition, set-returning functions are now disallowed within CASE and COALESCE constructs.

- When ALTER TABLE ... ADD PRIMARY KEY marks columns NOT NULL, that change now propagates to inheritance child tables as well (Michael Paquier)

- Prevent statement-level triggers from firing more than once per statement (Tom Lane)

Cases involving writable CTEs updating the same table updated by the containing statement, or by another writable CTE, fired BEFORE STATEMENT or AFTER STATEMENT triggers more than once. Also, if there were statement-level triggers on a table affected by a foreign key enforcement action (such as ON DELETE CASCADE), they could fire more than once per outer SQL statement. This is contrary to the SQL standard, so change it.

- Move sequences' metadata fields into a new pg_sequence system catalog (Peter Eisentraut)

A sequence relation now stores only the fields that can be modified by nextval(), that is last_value, log_cnt, and is_called. Other sequence properties, such as the starting value and increment, are kept in a corresponding row of the pg_sequence catalog. ALTER SEQUENCE updates are now fully transactional, implying that the sequence is locked until commit. The nextval() and setval() functions remain nontransactional.

The main incompatibility introduced by this change is that selecting from a sequence relation now returns only the three fields named above. To obtain the sequence's other properties, applications must look into pg_sequence. The new system view pg_sequence can also be used for this purpose; it provides column names that are more compatible with existing code.

The output of psql's \d command for a sequence has been redesigned, too.

- Make pg_basebackup stream the WAL needed to restore the backup by default (Magnus Hagander)

This changes pg_basebackup's -X/--xlog-method default to stream. An option value none has been added to reproduce the old behavior. The pg_basebackup option -x has been removed (instead, use -X fetch).

- Change how logical replication uses pg_hba.conf (Peter Eisentraut)

In previous releases, a logical replication connection required the replication keyword in the database column. As of this release, logical replication matches a normal entry with a database name or keywords such as all. Physical replication continues to use the replication keyword. Since built-in logical replication is new in this release, this change only affects users of third-party logical replication plugins.

- Make all pg_ctl actions wait for completion by default (Peter Eisentraut)

Previously some pg_ctl actions didn't wait for completion, and required the use of -w to do so.

- Change the default value of the log_directory server parameter from pg_log to log (Andreas Karlsson)

- Add configuration option ssl_dh_params_file to specify file name for custom OpenSSL DH parameters (Heikki Linnakangas)

This replaces the hardcoded, undocumented file name dh1024.pem. Note that dh1024.pem is no longer examined by default; you must set this option if you want to use custom DH parameters.

- Increase the size of the default DH parameters used for OpenSSL ephemeral DH ciphers to 2048 bits (Heikki Linnakangas)

The size of the compiled-in DH parameters has been increased from 1024 to 2048 bits, making DH key exchange more resistant to brute-force attacks. However, some old SSL implementations, notably some revisions of Java Runtime Environment version 6, will not accept DH parameters longer than 1024 bits, and hence will not be able to connect over SSL. If it's necessary to support such old clients, you can use custom 1024-bit DH parameters instead of the compiled-in defaults.

- Remove the ability to store unencrypted passwords on the server (Heikki Linnakangas)

The password_encryption server parameter no longer supports off or plain. The UNENCRYPTED option is no longer supported in CREATE/ALTER USER ... PASSSWORD. Similarly, the --unencrypted option has been removed from createuser. Unencrypted passwords migrated from older versions will be stored encrypted in this release. The default setting for password_encryption is still md5.

- Add min_parallel_table_scan_size and min_parallel_index_scan_size server parameters to control parallel queries (Amit Kapila, Robert Haas)

These replace min_parallel_relation_size, which was found to be too generic.

- Don't downcase unquoted text within shared_preload_libraries and related server parameters (QL Zhuo)

These settings are really lists of file names, but they were previously treated as lists of SQL identifiers, which have different parsing rules.

- Remove sql_inheritance server parameter (Robert Haas)

Changing this setting from the default value caused queries referencing parent tables to not include child tables. The SQL standard requires them to be included, however, and this has been the default since PostgreSQL 7.1.

- Allow multi-dimensional arrays to be passed into PL/Python functions, and returned as nested Python lists (Alexey Grishchenko, Dave Cramer, Heikki Linnakangas)

This feature requires a backwards-incompatible change to the handling of arrays of composite types in PL/Python. Previously, you could return an array of composite values by writing, e.g., [[col1, col2], [col1, col2]]; but now that is interpreted as a two-dimensional array. Composite types in arrays must now be written as Python tuples, not lists, to resolve the ambiguity; that is, write [(col1, col2), (col1, col2)] instead.

- Remove PL/Tcl's “module” auto-loading facility (Tom Lane)

This functionality has been replaced by new server parameters pltcl.start_proc and pltclu.start_proc which are easier to use and more similar to features available in other PLs.

- Remove pg_dump/pg_dumpall support for dumping from pre-8.0 servers (Tom Lane)

Users needing to dump from pre-8.0 servers will need to use dump programs from PostgreSQL 9.6 or earlier. The resulting output should still load successfully into newer servers.

- Remove support for floating-point timestamps and intervals (Tom Lane)

This removes configure's --disable-integer-datetimes option. Floating-point timestamps have few advantages and have not been the default since PostgreSQL 8.3.

- Remove server support for client/server protocol version 1.0 (Tom Lane)

This protocol hasn't had client support since PostgreSQL 6.3.

- Remove contrib/tsearch2 module (Robert Haas)

This module provided compatibility with the version of full text search that shipped in pre-8.3 PostgreSQL releases.

- Remove createlang and droplang command-line applications (Peter Eisentraut)

These had been deprecated since PostgreSQL 9.1. Instead, use CREATE EXTENSION and DROP EXTENSION directly.

- Remove support for version-0 function calling conventions (Andres Freund)

Extensions providing C-coded functions must now conform to version 1 calling conventions. Version 0 has been deprecated since 2001.

2.5 Deprecated features

Please note that the following items will be deprecated and will no longer be provided in EDB Postgres Advanced Server 11:

- Linux Graphical Installers
- Infinite Cache
- EDB Slony

2.6 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)