



# EDB Postgres Advanced Server 9.6 BETA

**Release Notes**

**November 7, 2016**

**EDB Postgres Advanced Server, Version 9.6 BETA Release Notes  
by EnterpriseDB Corporation  
Copyright © 2016 EnterpriseDB Corporation. All rights reserved.**

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](mailto:info@enterprisedb.com) **www**.enterprisedb.com

# Table of Contents

[Introduction](#)

[EDB Postgres Advanced Server v9.6 BETA Features](#)

[Known Issues in 9.6 BETA](#)

[Installers and Documentation](#)

[Platform Support and System Requirements](#)

[Incompatibilities](#)

[How to Report Problems](#)

# 1 Introduction

With this BETA release of EDB Postgres Advanced Server 9.6, 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 9.6 BETA is built on the open source PostgreSQL 9.6, which introduces an impressive number of improvements that enable databases to scale up and scale out in more efficient ways. PostgreSQL 9.6 also includes new monitoring capabilities that offer more visibility into the internal operations of the system such as vacuum progress monitoring and exposure of low level wait states in `pg_stat_statements`.

EDB Postgres Advanced Server 9.6 BETA adds a number of new features that will delight developers and DBAs alike, including:

- `DBMS_AQ` - Advanced Queuing provides database-integrated message queuing functionality. This feature is compatible with Oracle's implementation.
- Nested Sub Procedures - For Stored Procedure developers, this allows a named procedure to be defined and used inside another procedure or function. This feature is compatible with Oracle's implementation.
- DATABASE LINKS performance improvements with join & sort pushdown while using `postgres_fdw` (instead of `libpq`) for Postgres, and `oci` for Oracle.
- Advanced Server Partitioning improvements (Fast pruning for varchar, smallint, prepared statements)
- EDB\*Loader - 5 new features based on prioritized requests from users to offer more flexibility in the bulk loading process. (`NULLIF`, `SELECT EXPRESSIONS`, `Datatype(length)`, `BOUNDFILLER`, Column name in when clause)
- Oracle-Style Parallel Hints, Parallel Clause (Safe/Unsafe) for Procedures and Functions
- Support for `REFERENCING OLD AS old NEW AS new` clause in triggers.

These release notes are applicable to the 9.6.1.4 BETA release on November 7, 2016.

## 2 EDB Postgres Advanced Server v9.6 BETA Features

The major highlights of this release are:

- Integration of all PostgreSQL v9.6 features including the following:
  - Phase 1 introduction of parallelism
    - Designates a group of background worker processes to perform a scan in parallel for read only transactions

EnterpriseDB Corporation. All Rights Reserved.

- Provides better performance particularly for large table scans
  - Sequential scans, nested loops, hash joins, and aggregations can be parallelized
- Vacuum improvements
  - A timeout limit configuration parameter is now provided for snapshots that permits dead tuple removal by vacuuming after the limit is reached. This eliminates table bloating as the space could not be recycled during long running queries.
  - New bit in the freeze map identifies pages containing only frozen tuples and thus, lets the vacuum skip such frozen pages
- Synchronous replication now supports multiple simultaneous synchronous standby servers
- Full-text searching improvement for “phrases” (referred to as lexemes) whereby you can specify which lexemes are adjacent to each other in a given order, or a specified distance between the lexemes
- PostgreSQL foreign data wrapper (postgres\_fdw) pushdown processing to the remote foreign server
  - Pushdown processing to the remote server minimizes I/O
  - For joins
  - For sorting
  - For UPDATE and DELETE
- Monitoring improvements
  - VACUUM Progress Checker - Progress reporting for vacuum operations by the pg\_stat\_progress\_vacuum view
  - pg\_stat\_activity improvements - Waits for lightweight locks and buffer pins are now shown in pg\_stat\_activity. (Previously only heavyweight locks were shown.)
- See <https://www.postgresql.org/docs/9.6/static/release-9-6.html> for more information.
- Advanced Queuing Packages DBMS\_AQADM and DBMS\_AQ:
  - Advanced queuing provides database-integrated message queuing so that business applications can communicate with each other with producer applications enqueueing messages and consumer applications dequeuing messages.
  - Package DBMS\_AQADM provides the following functionality:
    - CREATE\_QUEUE\_TABLE: Creates a new queue table that can physically hold any number of queues
    - CREATE\_QUEUE: Creates a new queue in an existing queue table
    - DROP\_QUEUE: Drops an existing queue
    - DROP\_QUEUE\_TABLE: Drops an existing queue table
    - ALTER\_QUEUE: Modifies an existing queue
    - ALTER\_QUEUE\_TABLE: Modifies an existing queue table
    - START\_QUEUE: Enables enqueueing and/or dequeuing in an existing queue

- **STOP\_QUEUE**: Disables enqueue and/or dequeue in an existing queue
  - Package **DBMS\_AQ** provides the following functionality:
    - **ENQUEUE**: Posts a message to a queue. Messages can optionally be delayed so that they are not available for dequeuing for a certain number of seconds. Messages can optionally have an expiration time limit so that they will expire and move to the exception queue if they are not dequeued after a certain number of seconds.
    - **DEQUEUE**: Retrieves a message from a queue if one is available and optionally waits for one to become available
    - **REGISTER**: Registers a callback procedure that will be invoked in a background worker when messages are enqueued
    - **UNREGISTER**: Unregisters a callback procedure previously registered with **REGISTER**
  - See chapters 3.2 and 3.3 of the Database Compatibility for Oracle Developers Built-in Package Guide for more information.
- **Stored Procedure Language (SPL) Nested Subprocedures and Subfunctions**:
  - A named procedure or function declared and used within a standalone SPL procedure, function, anonymous block, trigger, package, or parent subprocedure/subfunction program
  - Defined in the declaration part of the enclosing parent program
  - Consists of the basic SPL (PL/SQL) block structure that can be called with IN, IN OUT, or OUT parameters
  - Procedures/subprocedures are invoked as individual statements and functions/subfunctions are expressions that return a value
  - See Chapter 3.2.6 of the Database Compatibility for Oracle Developer's Guide for more information.
- **Performance Improvements**:
  - **Parallelism**:
    - **Parallel Clause** to enable parallel safe, unsafe, or restricted mode within SPL standalone functions and procedures as well as functions and procedures within package bodies. See chapters 2.3.20, 2.3.24, and 2.3.25 of the Database Compatibility for Oracle Developers Reference Guide for more information.
    - **Parallelism Hints** to force parallel query scans. See Chapter 2.4.7 of the Database Compatibility for Oracle Developer's Guide for more information.
  - **Database link (dblink) pushdown**:
    - Database links created with the **CREATE DATABASE LINK** command that use the **oci** link for remote Oracle database connection, or the new **postgres\_fdw** link for remote Postgres database connection employ pushdown to the remote foreign servers.
    - Pushdown processing to the remote server minimizes I/O for joins, sorting, and UPDATE/DELETE statements

- See Chapter 2.3.18 of the Database Compatibility for Oracle Developers Reference Guide for more information.
  - Partitioned Tables:
    - Fast Pruning supports additional datatypes (varchar, smallint)
    - Fast Pruning supports prepared statements
    - Exchange Partition Improvements (compatible with Oracle behavior)
    - See Chapter 10.2 of the Database Compatibility for Oracle Developer's Guide for more information
- Improvements to EDB\*Loader
  - Support of the NULLIF clause to set a column to null if it meets the condition
  - Support for use of a SQL SELECT statement to return a column value
  - Support of the BOUNDFILLER clause, which disallows the field from directly loading column data, but permits its usage within a subsequent field expression to load other table columns
  - Specification of the field length in the data type parameter for the field definition. This enables defining the layout of fields over the data file without having to specify the exact starting position and ending position of each field.
  - Support of the field name in the WHEN condition clause to specify which table to load a data record based upon the value defined by the field name instead of having to specify the exact starting position and ending position of the data field.
  - See Chapter 3.3 of the Database Compatibility for Oracle Developers Tools and Utilities Guide for more information
- 13th Generation of compatibility with the Oracle Database including the following, some of which have been described in the previous bullet points: (Remember, you don't have to be an Oracle user to use the following features. Most are simply great database enhancements beyond what PostgreSQL offers.)
  - Support for advanced queuing with packages DBMS\_AQADM and DBMS\_AQ
  - Support for nested subprocedures and subfunctions
  - Support for parallelism
  - Support for the additional EDB\*Loader functionality
  - Support for REFERENCING OLD AS ... NEW AS ... clause in the CREATE TRIGGER command
  - Support of the ALL\_DIRECTORIES and DBA\_DIRECTORIES views
  - For a summary of all such newly added features compatible with Oracle databases, see Chapter 1.1 in 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

### 3 Known Issues in 9.6 BETA

The following known issues are applicable to the 9.6.1.4 BETA release on Nov 7, 2016.

- edbldr WHEN clause having columnname in condition behaves differently than sqlldr and edbldr with positional operands - (RM 39005)
- SPL debugger : On Windows pldbg\_wait\_for\_breakpoint() - (RM 39004)
- DBMS\_AQ: In Oracle message can be dequeued on the basis of correlation - (RM 38358)
- Error log "cache lookup failure" under certain query conditions - (RM 38099)
- Result node getting generated twice if ROWNUM is part of projection as well as where clause - (RM 37255)
- Inconsistency in behavior of PPAS and ORACLE for dbms\_scheduler.create\_job name - (RM 35768)
- build failure for ECPG tests - (RM 35541)
- DBMS\_AQ: dbms\_aqadm procedures allow SQL injection - (RM 38171)
- DBMS\_AQ: Queue message callback failed - (RM 38197)

### 4 Installers and Documentation

EDB Postgres Advanced Server v9.6.1.4 BETA is packaged and delivered as a series of interactive installers available on the Beta Programs download site -

<http://www.enterprisedb.com/downloads/beta-programs>

Beta Documentation is also provided on the same page.

### 5 Platform Support and System Requirements

EDB Postgres Advanced Server v9.6 supports 64 bit Linux and Windows server platforms. This includes the following:

Interactive Installers:

RHEL / CentOS / OEL 6, 7

Ubuntu 14.04, Debian 7.6, 8, SLES 11, 12

Windows 2012 R2, 2008 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>

## 6 Incompatibilities

PostgreSQL 9.6 contains a number of changes that may affect compatibility with previous releases. They are published in the PostgreSQL 9.6 Release Notes - <https://www.postgresql.org/docs/9.6/static/release-9-6.html> - and listed here for convenience.

- Improve the `pg_stat_activity` view's information about what a process is waiting for
- In `to_char()`, do not count a minus sign (when needed) as part of the field width for time-related fields
- Make `extract()` behave more reasonably with infinite inputs (Vitaly Burovoy)
- Remove PL/pgSQL's "feature" that suppressed the innermost line of CONTEXT for messages emitted by RAISE commands
- Fix the default text search parser to allow leading digits in email and host tokens
- Extend contrib/unaccent's standard unaccent.rules file to handle all diacritics known to Unicode, and to expand ligatures correctly
- Remove the long-deprecated CREATEUSER/NOCREATEUSER options from CREATE ROLE and allied commands
- Treat role names beginning with `pg_` as reserved
- Change a column name in the `information_schema.routines` view from `result_cast_character_set_name` to `result_cast_char_set_name`
- `psql`'s `-c` option no longer implies `--no-psqlrc`
- Improve `pg_restore`'s `-t` option to match all types of relations, not only plain tables
- Change the display format used for NextXID in `pg_controldata` and related places
- Update extension functions to be marked parallel-safe where appropriate

## 7 How to Report Problems

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

- Email: [support@enterprisedb.com](mailto:support@enterprisedb.com)

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