Postgres Plus® Advanced Server 9.3

Release Notes

July 31, 2013

Postgres Plus Advanced Server, Version 9.3 Release Notes
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 latest release of Postgres Plus Advanced Server, 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. The major highlights of this release are:

- Integration of all PostgreSQL v9.3 features, including:
  - Materialized views
  - Parallel pg_dump
  - Parallel pg_upgrade
  - Auto-updateable views
  - Enhanced JSON support
  - Writeable foreign tables

- 10th generation of Oracle compatibility, with updated support for Oracle-compatible built-in packages:
  - DBMS_RANDOM
  - DBMS_LOCK
  - DBMS_CRYPTO
  - DBMS_SCHEDULER
  - DBMS_MVIEW
  - UTL_ENCODE
  - UTL_HTTP
  - UTL_URL
  - Support for Oracle compatible exceptions in UTL_FILE package

  Expanded Oracle compatibility includes extended support for constructor methods, functions (REGEXP_INSTR, REGEXP_SUBSTR, REGEXP_COUNT), and EDB*Loader functionality.

Please Note: These release notes apply to all supported platforms.

2 PostgreSQL v9.3 Feature Highlights

2.1 PostgreSQL Merge Version

This version of Postgres Plus Advanced Server contains the merge of community PostgreSQL v9.3.Beta2, released June 27, 2013. To read the PostgreSQL release notes visit:

http://www.postgresql.org/docs/devel/static/release-9-3.html

Highlights of the release are listed below.

2.1.1 Lateral Sub-query Support

The SQL standard `LATERAL` keyword allows `SELECT` sub-queries in `FROM` clauses or set returning functions in `FROM` clauses to reference or parameterize items from the encompassing `SELECT`. Lateral subqueries can reference columns of tables defined outside the subquery at the same level, i.e. laterally. For example, a `LATERAL` subquery in a `FROM` clause could reference tables defined in the same `FROM` clause. Prior to v9.3 only the columns of tables defined above subqueries are recognized.

2.1.2 pg_terminate_backend your own queries

`pg_terminate_backend` allows you to stop backend processes running your queries.

2.1.3 Auto-Updatable Views

Many DML statements (e.g. `INSERT`, `UPDATE`, and `DELETE`) can now be successfully executed against views without creating `INSTEAD OF` triggers or `INSTEAD` rules. Auto-updatable views are “simple views” as specified in the SQL-92 rules. Executing DML against a view now successfully operates on the view's parent table without additional coding. This feature significantly reduces the programming burden on DBAs and developers for qualifying views.

2.1.4 Materialized Views

This foundational work for materialized views allows the creation of views that contain a persistent copy of the underlying table data, based on a view rule. This can help improve performance in some applications similar to data warehousing (only the view data is processed instead of the entire underlying table) and also provide convenience for remote clients by simplifying queries.

A materialized view has a rule like a normal view and a heap as well as other physical properties like a table. The rule is only used to populate the table and materialized view references in queries refer to the materialized data. Currently data is only populated on demand by the `CREATE MATERIALIZED VIEW` and `REFRESH MATERIALIZED VIEW` statements. The other supported commands are `ALTER MATERIALIZED VIEW` and `DROP MATERIALIZED VIEW`.

2.1.5 Support for CREATE RECURSIVE VIEW

This additional feature in support of the SQL standard simplifies the creation of views whose data population is based on recursive functions. This makes a developers work simpler in creating recursion based views and subsequently easier to understand by others by reducing the amount of coding involved.

2.1.6 Reduced lock contention and deadlocks around Foreign Key locks

Reducing an over-aggressive lock strength required by foreign key checks now reduces the number of deadlock patterns during concurrent transactions improving concurrency.

2.1.7 Optimize Referential Integrity Checks Involving NULLs

This optimization prevents Foreign Key triggers from firing unnecessarily during an `UPDATE` if the foreign-key column(s) contain any `NULL`s.

2.1.8 FOR KEY SHARE and FOR NO KEY UPDATE

New lock types used in the implementation of the reduced lock contention around FK locks.

2.1.9 Additional ALTER statement enhancements

`ALTER TYPE … ADD VALUE` now supports `IF NOT EXISTS` syntax

`ALTER ROLE ALL SET` allows you to set values that apply to all users in the database.

`ALTER RULE … RENAME` allows you to rename a rule without recreating it.

2.1.10 Event Triggers

These are triggers that fire on the execution of DDL (as opposed to DML) such as the `CREATE`, `ALTER` or `DROP` commands.

2.1.11 Expanded JSON Support

Four new JSON related functions have been added that simplify creating and emitting JSON formatted data stored in the database:

1. `json_agg(any record)` – converts any record into json formatted data

2. `to_json(any value)` – converts any value into json formatted data

3. `hstore_to_json(hstore)` – casts the hstore value into json formatted data (requires the hstore contrib module)

4. `hstore_to_json_loose(hstore)` – casts the hstore value into json formatted data with alternate data representations (requires the hstore contrib module) -> json

Four new JSON related operators have been added to simplify manipulating JSON data:

1. - returns the field value from a column stored in JSON format as valid JSON.

2. `->>` - returns the field value from a column stored in JSON format as plain text.

3. `#>` - returns the value from an element in an array stored in JSON format as valid JSON without the need for a field identifier-index descriptor.

4. `#>>` - returns the value from an element in an array stored in JSON format as plain text without the need for a field identifier-index descriptor.

Nine new JSON functions have been added for working with JSON data:

1. `json_each()` - converts JSON data into key-value records returning JSON formatted values.

2. `json_each_text()` - converts JSON data into key-value records returning plain text values.

3. `json_extract_path()` – returns a field value from a set of keys performing the same function as the operator "->".

4. `json_extract_path_text()` – returns a field value from a set of keys performing the same function as the operator "->>".

5. `json_object_keys()` - return the set of keys for a JSON object but only on the outermost object.

6. `json_populate_record()` - used to cast a single JSON record into a given user defined type.

7. `json_populate_recordset()` - used to cast a multiple JSON records into a given user defined type.

8. `json_array_length()` - returns the number of elements in a JSON array.

9. `json_array_elements()` - returns the elements in a JSON array as records.

JSON Parser API

The JSON parser has been converted into a recursive descent parser, and exposed for use by other modules such as extensions. The API provides hooks for all the significant parser events such as the beginning and end of objects and arrays, and provides functions to handle these hooks allowing for fairly simple construction of a wide variety of JSON processing functions. A set of new basic processing functions and operators was also added, which use this API, including operations to extract array elements, object fields, get the length of arrays and the set of keys of a field, deconstruct an object into a set of key/value pairs, and create records from JSON objects and arrays of objects.

2.1.12 New Array handling functions

Two new functions have been added for manipulating arrays: `array_remove()` and `array_replace()`. These functions allow users to delete or replace array elements based on their values matching a given search value.

2.1.13 Extended large object access to 4TB

The previous limit was 2GB.

2.1.14 Allow Standby Replicas to Follow a Different WAL Timeline

In multi-replica configurations using streaming replication, it is possible for replication to not start because a recently promoted replica's WAL timeline (i.e. the new master) differs from another joining replica. A new replication command, `TIMELINE_HISTORY`, allows joining replicas to ask the primary for any timeline history files that are missing from the standby. In addition, `START_REPLICATION` now takes a `TIMELINE` parameter, to specify exactly which timeline to stream WAL from.

2.1.15 PG_ISREADY utility

`PG_ISREADY` is a new command-line utility that tests whether a server is ready to accept connections.

2.1.16 Background Worker Processes

This new feature allows PostgreSQL to start additional processes that are bound to the running PostgreSQL instance – when PostgreSQL starts, so will the background process and when PostgreSQL is shutdown, the background process is also shut down. The background process also has access to PostgreSQL shared buffers. This feature makes it easy to ensure one or more programs are run whenever PostgreSQL runs without using scripts of other external means.

2.1.17 COPY FREEZE mode for Bulk Loads

This dramatically reduces the total I/O cost of loading data into an empty (just-created or just-truncated) table. Prior to v9.3, PostgreSQL can end up writing the table four times (initial write, add hint bits, freeze, WAL records for the freeze). This enhancement cuts out the last three of the four writes.

2.1.18 COPY support for piping data to/from an external program

The `COPY` command now has the added flexibility to exchange data with an external program with support in both the `psql` client and backend syntax.

2.1.19 postgres-fdw extension

A new foreign data wrapper called `postgres_fdw` is now included with PostgreSQL. This wrapper allows you to query other PostgreSQL instances and return results to the calling local server.

2.1.20 Support for Writable Foreign Tables

This foundation work extends contrib/postgres_fdw providing basic support that allows updates against remote Postgres servers. Future work will add performance improvements such as pushing the processing of `WHERE` clauses down to the foreign server when possible and returning a minimal result set.

2.1.21 pg_xlogdump contrib module

`pg_xlogdump` displays the WAL (write ahead log) in a human-readable format for a PostgreSQL database cluster. The utility is designed to enhance a users ability to debug server issues and can only be run by the user who installed the server, because it requires read-only access to the data directory.

2.1.22 pg_basebackup generates replica configuration

Added `-R` or `-write-recovery.conf` parameter to generate basic recovery file for a standby database.

2.1.23 Simplified OS shared memory configuration

Use POSIX shared memory: no more SHMMAX

2.1.24 Parallel pg_dump

Dumping a database can now be done faster with the new parallel processing feature. The parallel option is selected by including the `-j` / `--jobs` command line parameter of pg_dump. This option only works when specifying the directory-format archive as output (i.e. using the `-F d` command line option). Depending on the structure of your database and I/O capacity, this new option can significantly reduce the time required to dump a database.

2.1.25 pg_upgrade Optimizations and Performance Enhancements

pg_upgrade is designed to convert an existing database to a newer version of the database in a small time frame without the need for time-consuming dump and restore operations. A number of significant improvements have been made to pg_upgrade's binary upgrade process:

- A 2-4x improvement in link mode upgrade speed
- Utilization of multiple CPUs in parallel when dumping/restoring multiple schemas in link mode providing significant performance improvements for databases with many schemas (refer to the –jobs command option)
- Tablespace parallel copy/link during upgrade for user data and index files provides faster completion for large databases.

3 Performance Enhancements

This section details performance enhancements that are new in Postgres Plus Advanced Server 9.3.

3.1 Improved Partitioning

Advanced Server 9.3 demonstrates `SELECT` and `INSERT` performance gains of up to 170 times faster for tables with large partition counts. Postgres Plus Advanced Server 9.3 has broken previously existing barriers, and tables can now have thousands of partitions with a very limited performance overhead.

4 New Oracle Compatibility Features

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.

For more information about Oracle Compatible features, see the Postgres Plus Advanced Server Oracle Compatibility Developer's Guide, available from the EnterpriseDB website at:

http://www.enterprisedb.com/products-services-training/products/documentation

4.1 Materialized Views

Foundation support for Oracle compatible syntax of materialized views has been introduced. The follow `CREATE MATERIALIZED VIEW` syntax is supported:

```
CREATE MATERIALIZED VIEW name
   [build_clause][create_mv_refresh] AS subquery
```

Where *build_clause* is:

```
BUILD {IMMEDIATE | DEFERRED}
```

Where *create_mv_refresh* is:

```
REFRESH [COMPLETE] [ON DEMAND]
```

Additional support for materialized views is provided by `DBMS_MVIEW`, also new in Advanced Server 9.3.

4.2 Oracle Compatible Built-In Package Support

Advanced Server 9.3 includes expanded support for functions and procedures in the following Oracle-compatible packages:

4.2.1 DBMS_RANDOM

This package generates random numbers. Supported functions and procedures include: `INITIALIZE`, `NORMAL`, `RANDOM`, `SEED`, `STRING`, `TERMINATE`, and `VALUE`.

4.2.2 DBMS_LOCK

PPAS 9.3 includes support for the `DBMS_LOCK.SLEEP` procedure, which is commonly used when migrating Oracle applications. `DBMS_LOCK.SLEEP` temporarily suspends a session based on the amount of time passed to the procedure.

4.2.3 DBMS_CRYPTO

PPAS 9.3 adds support for an Oracle compatible package that encrypts and decrypts data. The following DBMS_CRYPTO subprograms have been implemented: DECRYPT, ENCRYPT, HASH, MAC, RANDOMBYTES, RANDOMINTEGER, and RANDOMNUMBER.

4.2.4 DBMS_SCHEDULER

This package provides a way to schedule and manage database jobs. The following DBMS_SCHEDULER subprograms have been implemented: CREATE_JOB, CREATE_PROGRAM, CREATE_SCHEDULE, DEFINE_PROGRAM_ARGUMENT, DISABLE, DROP_JOB, DROP_PROGRAM, DROP_PROGRAM_ARGUMENT, DROP_SCHEDULE, ENABLE, EVALUATE_CALENDAR_STRING, RUN_JOB, SET_JOB_ARGUMENT_VALUE.

4.2.5 DBMS_MVIEW

This package provides an Oracle compatible way to manage the new Postgres Plus Advanced Server materialized view feature. PPAS 9.3 includes the following DBMS_MVIEW subprograms: GET_MV_DEPENDENCIES, REFRESH, REFRESH_ALL_MVIEWS, REFRESH_DEPENDENT.

4.2.6 UTL_ENCODE

This package provides a way to encode and decode data. PPAS 9.3 includes the following UTL_ENCODE procedures and functions: BASE_64_DECODE, BASE_64_ENCODE, MIMEHEADER_DECODE, MIMEHEADER_ENCODE, QUOTED_PRINTABLE_DECODE, QUOTED_PRINTABLE_ENCODE, TEXT_DECODE, TEXT_ENCODE, UUDECODE, UUENCODE.

4.2.7 UTL_HTTP

This package provides a method to retrieve data from URLs. The following UTL_HTTP functions have been implemented in this release: REQUEST, REQUEST_PIECES.

4.2.8 UTL_URL

This package is used to escape illegal characters within a URL. PPAS 9.3 implements the ESCAPE and UNESCAPE functions.

4.2.9 Support for Oracle compatible exceptions in UTL_FILE package

Additional Oracle compatible exceptions have been added to the UTL_FILE package, thus reducing the manual recoding effort when migrating from Oracle.

4.3 Support for Oracle compatible constructor support for object types

Advanced Server 9.3 allows you to add custom constructor methods when defining object types. PPAS also includes support for the NEW keyword, allowing you to explicitly invoke an object constructor to create a new object.

4.4 Support for Function: REGEXP_INSTR

REGEXP_INSTR finds the occurrence of the pattern in the source string when starting at start_position using the match_pattern.

```
REGEXP_INSTR(<source_string>, <pattern>
[[, <start_position>][,<occurrence>][, <return_option>]
[, <match_parameter>][,<sub_expression>]])
```

4.5 Support for Function: REGEXP_SUBSTR

REGEXP_SUBSTR searches a string for the specified pattern:

```
REGEXP_SUBSTR(<source_string>,<pattern>[[,<start_position>]
[,<occurrence>][,<match_parameter>][,<sub_expression>]])
```

4.6 Support for Function: REGEXP_COUNT

REGEXP_COUNT searches for a string and returns the number of occurrences.

```
REGEXP_COUNT(<source_string>,<pattern>,[[,<start_position>[,
<match_parameter>]])
```

4.7 Support for Package Program Attributes

Advanced Server now supports the STRICT, LEAKPROOF, COST, ROWS, and SET attributes for package functions and procedures.

4.8 EDB*Loader Enhancements

EDB*Loader is a high-speed bulk data loader with parallel processing and many features not available using the Postgres COPY command.

4.8.1 ROWS Parameter

EDB*Loader used to process an entire data file as a single transaction. PPAS 9.3 has added the ROWS option, which performs commits to the database batches of the size specified using the `ROWS` parameter. This allows EDB*Loader to use fewer system resources, and adds options for handling problem data.

4.8.2 Additional Error Codes

EDB*Loader now supports two additional error codes in addition to `0 - Success` and `1 - Failure`. Execution now also emits `2 - Warning` and `3 - Fatal`.

4.8.3 Stream output files to client

EDB*Loader can now stream its output files from the server to the client; i.e. the Log, Bad, and Discard files. This allows users to view the state of these files in process, saves space on the server, and preserves results up to any point where an error may occur that aborts the load.

4.8.4 Empty string handling GUC

PPAS 9.3 includes a new GUC (`edbldr.empty_csv_field`) to handle `NULL` or empty string values in input file when loading data using EDB*Loader. Legal values are `null`, `empty_string`, and `pgsql`. This GUC allows you to choose between Oracle and PostgreSQL behavior.

5 Client Connectors

5.1 JDBC

The version of the JDBC connector packaged with Postgres Plus Advanced Server is based on pgJDBC 9.2-1002. Two versions are available, one built against JDK 1.4 (`edb-jdbc14.jar`) and one built against JDK 1.6 (`edb-jdbc16.jar`).

This component is optionally installed with StackBuilder Plus, and works in all supported environments.

5.2 ODBC

The version of the ODBC connector packaged with Postgres Plus Advanced Server is based on psqlODBC-09.02.0100.

This component is optionally installed with StackBuilder Plus, and works in all supported environments.

5.3 .NET

The version of the .NET connector packaged with Postgres Plus Advanced Server is based on Npgsql-2.0.12.

This component is optionally installed with StackBuilder Plus, and works in the Windows supported environments.

6 Supporting Tools and Utilities

The Advanced Server 9.3 installer also installs the tools and utilities listed below:

6.1 Postgres Enterprise Manager (PEM) Client

The PEM client can be used stand-alone as a database administration and management tool as well as the client for the Postgres Enterprise Manager monitoring system. The administration and management features of PEM are designed to answer the needs of all users, from writing simple SQL queries to developing complex databases. The graphical interface supports all PostgreSQL features and makes administration easy. The application also includes a syntax highlighting SQL editor, a server-side code editor, an SQL/batch/shell job scheduling agent, support for the Slony-I replication engine and much more. Server connection may be made using TCP/IP or Unix Domain Sockets (on *nix platforms), and may be SSL encrypted for security. No additional drivers are required to communicate with the database server. PEM also contains a SQL Profiler that makes it easy for developers to find slow running queries and optimize them.

PEM is optionally installed by StackBuilder Plus, and runs natively in Linux, Windows and Mac environments. Install PEM into one of these environments to access the database server installed on Solaris.

The PEM client is based on pgAdmin 1.16.2.

6.2 EDB*Plus

EDB*Plus is Postgres Plus Advanced Server's Oracle-like command line tool for communicating with the database server. It is similar to Oracle's SQL*Plus and understands most SQL*Plus commands. It also supports spooling, running scripts using @, set commands, column formatting, etc. and can define variables in `login.sql`.

The utility is installed by the Postgres Plus Advanced Server meta-installer and runs natively in all supported environments.

6.3 Migration Toolkit

The Migration Toolkit provides fast, flexible and customized database migration from Oracle, SQL Server, Sybase, and MySQL to PostgreSQL and Postgres Plus Advanced Server in online and offline modes. It is a powerful tool for moving schema, tables, constraints, data, stored procedures, triggers, and additional objects from other databases to Postgres Plus Advanced Server automatically.

The utility is installed by the Postgres Plus Advanced Server meta-installer and runs natively in all supported environments.

6.4 pgpool-II

The Postgres Plus Advanced Server installer includes and installs pgpool II v3.2.4. pgpool-II is middleware that works between PostgreSQL database servers and PostgreSQL database clients. Only the following pgpool features are supported: connection pooling and load balancing.

This component is optionally installed with StackBuilder Plus and works in all supported environments except the Windows platform.

6.5 Slony

Slony v 2.1.3 is packaged with Postgres Plus Advanced Server and contains the following changes:

- Fixed a bug in `MOVE SET` that could cause data on other nodes to get out of sync.
- Removed the '`might be unsupported'` when working with PG 9.2.
- Fixed duplicate key detection on `sl_nodelock`
- Added --`with-pgport` configuration option

For a complete list of Slony bug fixes and more information about Slony replication, please visit: http://slony.info/

This component is optionally installed with StackBuilder Plus, and works in all supported environments.

6.6 PostGIS

The version of PostGIS shipped in StackBuilder Plus is v2.1.0, Beta 3.

6.7 PL/Java

The version of PL/Java packaged with Postgres Plus Advanced Server is: v1.4.3. This component is installed with the database server and it works in the Linux and Windows environments.

6.8 PL/Python

The version of PL/Python packaged with Postgres Plus Advanced Server was built against ActivePython v3.2.2. This component is installed with the database server, and works in all supported environments.

6.9 pgSNMPd

The version of pgSNMPd packaged with Postgres Plus Advanced Server is v1.0. This component is installed with the database server and it works in a Linux environment.

6.10iCache (Infinite Cache)

The version of iCache packaged with Postgres Plus Advanced Serveris based on memcached v1.4.5. This component is optionally installed with StackBuilder Plus, and works in all supported environments except the Windows platforms.

6.11pgAgent

The version of pgAgent packaged with Postgres Plus Advanced Server is 3.3.0. This component is optionally installed with StackBuilder Plus and it works in all supported environments.

6.12PgBouncer

The version of PgBouncer packaged with Postgres Plus Advanced Server is 1.5. This component is optionally installed with StackBuilder Plus and it works in all supported environments.

7 Installers

After initially installing Advanced Server 9.3 with the Advanced Server meta-installer, you can use StackBuilder Plus or Update Monitor to keep the installation up-to-date.

7.1 The Advanced Server Meta-Installer

The Advanced Server meta-installer utilizes BitRock installer technology and provides the following features:

- Easy point-and-click installation procedures
- Common installer technology for all platforms
- Silent installation option (typically used by ISVs)
- Installation options for users with limited privileges (e.g. non-root Linux users and non-administrator Windows users)

7.2 StackBuilder Plus

StackBuilder Plus is distributed with Advanced Server. StackBuilder Plus provides a wide array of complimentary components for Advanced Server. StackBuilder Plus also provides Update Monitor to notify you when packages you have installed have updates available, and helps you download and install the updates.

StackBuilder Plus is offered as an option to run at the end of your Advanced Server installation, or can be accessed through the Advanced Server system menu.

7.3 Update Monitor

Update Monitor will notify you when new updates are available for any component you have installed, including the database server or StackBuilder Plus modules. You can use Update Monitor to access StackBuilder Plus to download and install a component and read the release notes. Update Manager is automatically installed in your desktop system tray (you will see the blue elephant icon), and automatically alerts you to available updates from EnterpriseDB.

7.4 Product Keys for Localized Language Installations

If you wish to install Advanced Server in localized Japanese, Korean, Traditional Chinese, Simplified Chinese, or a Central American or South American language you will need to enter a *Product Key* when the installer executes. Product keys can be obtained from distributors in each country as follows:

Locale Code	Country Locale	Distributor	Contact	Contact Email
ja_jp	Japanese	EDB - Japan	Yuji Fujita	yuji.fujita@enterprisedb.com
ja_jp	Japanese	COMTEC Inc.	Hiroaki Tanaka	htanaka@ct-net.co.jp
ja_jp	Japanese	SIOS Technology, Inc	Noriko Daitoku	tdaitoku@sios.com
ja_jp	Japanese	K.K. Ashisuto	Yoko Takase	edb_sal@ashisuto.co.jp
ja_jp	Japanese	SRA OSS, Inc. Japan	Kazuhiko Hamada	edb@sraoss.co.jp
ja_jp	Japanese	FUJITSU Social Science Laboratory Limited	Atsushi Ogasawara	ssl-info@cs.jp.fujitsu.com
zh_tw	Chinese Taiwan (ROC)	EDB - Japan	Yuji Fujita	yuji.fujita@enterprisedb.com
zh_cn	Chinese (PRC)	EDB - Japan	Yuji Fujita	yuji.fujita@enterprisedb.com
zh_hk	Chinese (Hong Kong S.A.R)	EDB - Japan	Yuji Fujita	yuji.fujita@enterprisedb.com
ko_kr	Korean	Daou Tech, Inc.	K.I Lee	kilee@daou.co.kr
es_ar	Argentina - Spanish	Genup-IT	Fernando Maidana	fmaidana@genup-it.com
Pt_br	Brazil - Portuguese	4Linux	Flavio Gurgel	contato@4linux.com.br
pt_br	Brazil - Portuguese	Tecnisys	Rogerio Carvalho	Rogerio.carvalho@tecnisys.com.br
es_bo	Bolivia - Spanish	CommIT	Oxiel Contreras	oxiel.contreras@commit.com.bo
Es_bo	Bolivia - Spanish	iTEAM	Marco Orellana	marco.orellana@iteam.com.bo
es_cl	Chile - Spanish	SEIS, SA	Hector Barrios	hector.barrios@seissoft.cl
es_co	Colombia - Spanish	SUMMAN	Felipe Posada	fposada@summan.com
es_co	Colombia - Spanish	Tayronaweb	Rafael Cortez	rafael.cortes@tayronaweb.com
es_ec	Ecuador - Spanish	Software Libre Andino	Juan Anamaria	juanam@softwarelibreandino.com
es_gt	Guatemala - Spanish	Systemshause Westfalia	Esteban Calderon	esteban.calderon@westfalia-it.com
es_hn	Honduras - Spanish	Systemshause Westfalia	Esteban Calderon	esteban.calderon@westfalia-it.com
es_mx	Mexico - Spanish	TEAM	Claudia Garcia	cgarcia@teamnet.com.mx
es_ni	Nicaragua - Spanish	Systemshause	Esteban Calderon	esteban.calderon@westfalia

		Westfalia		-it.com
es_pe	Peru - Spanish	Software Libre Andino	Juan Anamaria	juanam@softwarelibreandino.com
es_py	Paraguay - Spanish	Smartechpy	Charles Charotti	ccharotti@gmail.com
es_sv	El Salvador - Spanish	Systemshause Westfalia	Esteban Calderon	esteban.calderon@westfalia-it.com
es_uy	Uruguay - Spanish	Ideasoft	Enrique Tucci	etucci@ideasoft.biz
es_ve	Venezuela - Spanish	HIA Technology de Venezuela	Ernesto Lozano	elozano@hiatechnology.com.ve

This list of distributors possessing installation product keys is also available at:

http://www.enterprisedb.com/product-keys

7.5 Internationalization / Localization

Features from v9.1 and prior versions are localized in Postgres Plus Advanced Server v9.3 into the following languages:

- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

8 Service Pack Maintenance

A number of maintenance items (bug fixes and enhancements) have been added to Postgres Plus Advanced Server for release 9.3. Some of the more interesting ones are noted below:

8.1 Database Server

- Advanced Server 9.3 supports the `dml` keyword for the `edb_audit_statement` configuration parameter (30828).

- Fixed a bug to prevent the user from dropping the partition or sub-partition key column in a partitioned table (20806).

- Fixed a bug in copy handling for partitioned table with a VPD policy. The bug was causing `pg_dump` to emit the same record multiple times when VPD is used on a partitioned table. More generally, it caused `COPY` to behave differently on a partitioning root depending on whether or not VPD is present, which is not desirable (30562).

- Fixed a bug in OCI database links which was causing the server to crash when doing an `INSERT` or `UPDATE` in a `VARCHAR` column of a remote table using OCI link and the value is being selected from a local PPAS table (31046).

- Advanced Server now prevents a user from adding a column to a partition backing table - you can still add a column to the partitioning root, but not to a backing table (30904).

- Advanced Server prevents a user from changing the data type of a partitioning or subpartitioning key column (30904).

- When invoked using a connector, the PL/PGSQL functions containing out parameters were being resolved as SPL out parameter style regardless of the db_dialect value, this was causing an error since in postgres style the `OUT` and `INOUT` parameter values are part of the result tuple (30334).

- Fixed a bug in pg_dump to restore permission's on schema (previously, the dump failed) (9255).

- VPD policy should only be attached to table. Attaching VPD policy to a view or index will result in an error (30805, 30795, 30743, 30582).

- Revise the `DECODE()` function for better Oracle compatibility by making it type-aware. Previously the `DECODE()` function would always return a

`VARCHAR`. The return type is now chosen using an algorithm more compatible with Oracle (30123).

➔ Allow `ROLLBACK` as a separate value for `edb_audit_statement`. This will allow the user to choose whether they want the `ROLLBACK` statement to appear in the audit log (30467).

➔ Allow some options to be set on package functions and procedures. We now support `STRICT`, `LEAKPROOF`, `COST`, `ROWS`, and `SET` on package functions and procedures (30783).

➔ Advanced Server does not allow synonyms to point to package objects. Synonyms for a package are not supported in Advanced Server. Previously we allowed synonyms to be created for package objects, which was not compatible with Oracle (30282).

➔ Corrected EDB*Loader behavior so that it applies validity checks e.g. numeric bounds, string length checking etc properly in case of expressions (23103).

➔ `UPDATE` statement now enforces the VPD policy when the column mentioned in `sec_relevant_cols` (parameter defined by `dbms_rls.add_policy` procedure) is part of `UPDATE` Statement. VPD should check against selected columns as well as modified ones. This is a security issue (30423).

➔ Fixed a bug when sub-queries are used in the target list of a hierarchical query (23338).

➔ Advanced Server already supported `NULL` in the `DEFAULT` or `MAXVALUE` partition, but this release adds Oracle-compatible support for directing `NULL`s to a particular list partition (30528).

➔ The server no longer crashes when `OUT` parameters are used in a cursor declaration (30497).

➔ Added new GUC (`edbldr.empty_csv_field`) to handle `NULL` or empty string values in input file when loading data using EDB*Loader. Legal values are `null`, `empty_string`, and `pgsql`. This is to address the case when user is trying to load data using EDB*Loader to a table column with `NOT NULL` constraints. The GUC is added to choose between Oracle and pgsql behavior. In pgsql mode, an empty field is treated as a null if the raw field contains no characters, but as an empty string if it contains a pair of delimiters with nothing in between (30556).

➔ Do not reset `CURRENT_USER` when releasing a savepoint. Doing a `RELEASE SAVEPOINT` in a transaction was wrongly resetting `current_user` (23439).

- Include user-defined types in Oracle-compatible types views (21928).

- Fix a server crash when using composite type as `OUT/INOUT` parameter in an SPL package (30207).

- Exclude inaccessible objects from Oracle-compatible system views. The user should only see the accessible objects in `user_*` and `all_*` views (30279).

- Make sure VPD policies are enforced when data is loaded through EDB*Loader. This is a security issue (30290).

- Fixed a memory leak that occurs when rotating an audit file that is already in use. (23332).

- All optional parameters supported by EDB*Loader should be logged. Previously it was only logging `skip` and `errors` parameters (23396).

- Fix a bug in EDB*Loader to support overlapping positioning specified in the control file. The bug was resulting in an error if the positioning was not specified in a specific order (23068).

- Warnings were generated when creating a package that have user-defined exception in it. This has been fixed (30020).

- Fix a bug in EDB*Loader for handling `NULL` when fixed format is used in the control file (21471).

- When EDB*Loader truncates a table, it uses the `DROP_CASCADE` option, potentially causing unrelated tables to get truncated. It should use `DROP_RESTRICT` instead (23320).

- Fix the warnings when non-spl functions are used in an SPL block and `edb_stmt_level_tx` GUC is turned on (19733).

- Doing a `COMMIT` inside an SPL block with multiple sub-transactions was producing unexpected results. This has been fixed (23181).

- Log records to the discard file only if no table accepts them. This was not happening when multiple tables are used in the control file (17420).

- Enforce proper preservation of object type OIDs across pg_upgrade. In binary upgrade mode dumping of OID was missing in case of Oracle style object types (23288).

- Teach pg_dump not to query partitioning tables on pre-9.1 servers. This was causing an issue when taking a dump of pre-9.1 server using the pg_dump of an older version (22723).

- Fix type OID preservation logic for collection and composite types. This was causing an issue when doing a upgrade from pre-9.2 server to 9.2, it was running into an OID conflict since the OID for collection and composite type wasn't preserved (23273).

- Previously, the `COPY` command was not working for synonym defined for objects with Oracle database links (17069).

- Revamp the EDB-specific extensions to `CREATE OR REPLACE VIEW`. This allows `CREATE OR REPLACE VIEW` to preserve properties such as column privileges, security labels, triggers, rules, etc. when a view is replaced (21507).

- Advanced Server no longer crashes if an SPL commit is used with cursors inside an SPL block (22832).

- Fixed `DBMS_PIPE` crash when `NULL` messages are used in pack/unpack operations (22961).

- `ppas-agent-9.2` service takes too long to start when there are multiple plugins added to preload libraries (22873).

- Fixed server crash on `prepare as select func()` with extended protocol (22707).

- Previously, Advanced Server logged cancelled queries as failed connection attempts - this is fixed in 9.3 (21057).

- Fixed SPL crash when using `RAISE` with invalid error code (22630).

- Revoke default permissions on `UTL_TCP` implementation functions. This is a security issue (21038).

- Fix pg_dump to dump casts involving packages. This was causing pg_dump operation to fail if table created in a user schema and package in public schema using that table (21837).

- Fix a bug encountered when creating large packages. In this particular case, the package contained a cursor with more then 250 rows - this was causing a `row too big` error (22249).

- ➔ If you launch the server with `log_connections=on`, for each connection that's made (even with psql), you get a message in the log claiming it's a replication connection. This has been fixed (22054).

8.2 Migration Toolkit

- ➔ Migration Toolkit now supports migration of tables in which a `COMMENT` statement includes an escape sequence (30890).
- ➔ Migration Toolkit now supports migration of comments that include a single-quote (') (20675).

8.3 Connectors

- ➔ Advanced Server 9.3 adds support for `OCI_DESCRIBE_ONLY` (30822).
- ➔ `IN` parameter handling has been updated to a correct segmentation fault caused by `OCIbind` pointer (23306).
- ➔ `OCIBindByPos` now handles `NULL` values correctly (30821).
- ➔ Advanced Server 9.3's ODBC connector adds support for the `gssapi for GSS request` option for Windows (30688).
- ➔ Advanced Server 9.3's ODBC connector is now merged with `psqlODBC-09.02.0100` (30760).
- ➔ PPAS 9.3 is updated to correct a problem in ecpg (both Advanced Server and community PostgreSQL) that initiated a core dump (21886).
- ➔ PPAS 9.3 corrects a problem with ecpg in which ecpg was returning invalid values for length, precision and scale (20315)
- ➔ Advanced Server 9.3's .NET connector is now merged with `npgsql-2.0.12` (24413).

8.4 EDB*Plus

- ➔ PPAS 9.3 updates EDB*Plus to return the result of a `DESC` command in order by `ordinal_position` (20689).
- ➔ EDB*Plus now exhibits /`NOLOG` behavior that is compatible with SQL*Plus (23156).

8.5 PEM Client

- The PEM Client no longer crashes if you rename the function name (from the Properties dialog) (30580).

- Scheduled tasks are successfully removed from PEM client (30896).

- The following fixes are coming in from the recent merge with pgAdmin 1.16.2:

  - Fix the help path on the import dialogue and improve the error handling

  - Fix path the the Search Objects help doc

  - Fix UTF-8 display for guru hints

  - Prevent a crash when creating a stored procedure in PPAS

  - Disable the Favourites menu items if there is no favourites file path

  - Avoid a possible crash on Linux when using the "Script" options of the Query Tool

  - Fix comments on constraints

  - Fix schema prefixes in cast definitions in the browser.

  - Treat sequences as first class objects in the Grant Wizard. Support USAGE and remove obsolete RULE permissions

  - Improve the copy handling on the Edit Grid so that it works more consistently and predictably

  - Fix comments on columns.

  - Remove the "Apply" buttons from the function, view package and external table dialogues. Their use is discouraged, and the code is fragile, complex and very buggy.

  - Fix a crash that could occur if the browser fails to detect that an object has been changed by another session.

  - Don't prompt the user for a password if they're using a client certificate.

  - Ensure global backups use the mintenance database to avoid access issue with postgres or template1.

- Fix SSL certificate authentication.

- Ensure the Query Tool's maximum column length setting is honoured.

- Handle default privileges to PUBLIC correctly.

- Fix the editing of pg_hba.conf files

- Fix the display of extension owner names.

- Fix a bug that could cause a NOTICE to be displayed when rendering FTS Configurations in the tree.

- Don't try to display dependents or dependencies for pgAgent jobs, schedules or steps.

- Fix the handling of expiry times when modifying roles.

- Prevent foreign keys being recreated unnecessarily when modifying tables

- Prevent Cmd+S adding characters to the Query Tool when the Save button is disabled.

- Fix handling of Greenplum partitions

- Fix the database SQL when resetting a configuration parameter.

- Refresh the toolbar buttons when a list has been updated.

- Fix the "NO INHERIT" query.

- Fix the change of the superuser attribute.

- Prevent a crash when using Cmd-A on a file loaded into the Query Tool when opened with Cmd-E on Mac.

- Fix the queries used to get object comments to allow for duplicate OIDs that may be present following use of pg_upgrade.

9 Documentation Updates

For the latest versions of the Postgres Plus Advanced Server guides, please visit:

http://www.enterprisedb.com/documentation

Please note that Postgres Plus Advanced Server subscription holders can also access PDF versions of Advanced Server by logging into the EnterpriseDB website, and visiting the customer portal at:

http://www.enterprisedb.com/support

For complete installation information for Advanced Server 9.3 Beta 1, please see the Postgres Plus Advanced Server Installation Guide.

The updated Advanced Server Oracle Compatibility Developers Guide contains an additional 159 pages documenting the Oracle-compatible features found in Advanced Server 9.3.

The Postgres Plus Advanced Server Guide has been updated to include information about using EDB*Loader, new Advanced Server configuration parameters, and an up-to-date PL/Debugger section.

The Postgres Plus ODBC Connector Guides has been updated to reflect recent changes to the ODBC connector.

10 Upgrade Paths

You can use pg_upgrade to facilitate an upgrade from PPAS 9.2 to 9.3. pg_upgrade allows you to update your installation in a matter of minutes (for most users) without the downtime and additional planning that used to be required when using the traditional dump and restore method.

Usage details for pg_upgrade can be found in the Postgres Plus Advanced Server Installation Guide available on the EnterpriseDB web site at:

http://www.enterprisedb.com/documentation/english

11 Platform Support and System Requirements

Postgres Plus Advanced Server v9.3 is supported on the following production level platforms:

Advanced Server 9.3 is supported on the following platforms.

32 bit Windows:

- Windows Server 2008 R1

64 bit Windows:

- Windows 2012
- Windows Server 2008 R1
- Windows Server 2008 R2

32 bit Linux:

- CentOS 6.x
- Red Hat Enterprise Linux 6.x
- SLES 11.x
- Ubuntu 12.04 LTS

64 bit Linux:

- CentOS 6.x
- Red Hat Enterprise Linux 6.x
- Ubuntu 12.04 LTS

Solaris:

- Solaris SPARC 64, v10.x and v11.x

Postgres Plus Advanced Server 9.3 may work on the following OS platforms to varying degrees in non-production environments. EnterpriseDB will address issues in these environments on a best effort basis, where best effort is defined as a reasonable response to a request that can be achieved within the context of the technology and platforms available, and prevailing business conditions.

32 bit Windows:

- Windows 8
- Windows 7
- Windows Vista
- Windows 2003

64 bit Windows:

- Windows 8
- Windows 7
- Windows Vista

32 and 64 bit Linux:

- Fedora 18.x
- Fedora Core 6
- OpenSuSE 12.x
- Ubuntu 13.04

EnterpriseDB will address issues in these environments on a best effort basis, where best effort is defined as a reasonable response to a request that can be achieved within the context of the technology / platforms available, and prevailing business conditions.

To inquire about operating system support, contact us, by:

- Email: sales-us@enterprisedb.com or sales-intl@enterprisedb.com
- Phone: +1-781-357-3390 or 1-877-377-4352
- Web: http://www.enterprisedb.com/general-inquiry-form

System Requirements

Minimum hardware requirements for running Postgres Plus Advanced Server are:

- a 1 GHz processor
- 1 GB of RAM
- 512 MB of HDD

12 Known Issues

Postgres Plus Advanced Server provides support for PL/Perl version 5.14x. Using a more recent version of PL/Perl may cause server crashes.

The PostGIS installer for Windows 32 is not available for Beta 1; it is expected to be available for Beta 2 release.

13 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)