



EDB Postgres Backup and Recovery 2.0

Release Notes

June 5, 2017

**EDB Postgres Backup and Recovery Version 2.0 Release Notes
by EnterpriseDB Corporation
Copyright © 2017 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 **www**.enterprisedb.com

Table of Contents

[Introduction](#)

[EDB Postgres Backup and Recovery 2.0 Features](#)

[Installers and Documentation](#)

[Platform Support and System Requirements](#)

[How to Report Problems](#)

1 Introduction

EDB Backup and Recovery simplifies the backup and recovery procedures and reduces the chance for human error by automating many of the tasks that were taken care of by hand or by custom developed scripts.

EDB Backup and Recovery has been built on top of `pg_basebackup`, allowing for complete hot physical backups of your PostgreSQL or EDB Postgres Advanced Server database. It is a centralized server that can connect to and backup multiple local or remote Postgres databases.

In previous releases, EDB introduced capabilities such as policy enforced management and naming of the backup files, compressed storage, and low disk space alerts.

Backup and Recovery 2.0 adds the most commonly requested feature, block level incremental backup. Block level incremental backup supports the needs of customers with large databases to be able to take backups in a more efficient way by increasing the speed at which backups can be taken and minimizing the amount of space required.

Notice: The names for some of EDB's products have been changed. The product formerly referred to as 'EDB Backup and Recovery Tool (BART)' is now referred to as 'EDB Postgres Backup and Recovery'. As such, you may find the term 'BART' referred to throughout the software, documentation and release notes.

2 EDB Postgres Backup and Recovery 2.0 Features

The major highlight of this release is block level incremental backup

- Block-level incremental backups are now supported.
 - A block-level incremental backup saves a copy of the file blocks from tables, indexes, sequences, and other database objects that have been modified since a previous parent backup.
 - For more information, see Section 2.2 of the Backup and Recovery documentation.
 - The BART configuration file now contains the `allow_incremental_backups` parameter, which determines whether or not an incremental backup is allowed to be taken for specific database servers.
 - For more information, see Section 4.2.5 of the Backup and Recovery documentation
 - A new WAL scanner program (`bart-scanner`) is introduced to support

block level incremental backup. The bart-scanner scans incoming WAL files to identify and record which blocks have been modified in the relation files. These recorded changes are then used to locate the modified blocks to copy and save when an incremental backup is performed.

- For more information, see Section 5.5 of the Backup and Recovery documentation.
 - The BACKUP subcommand now supports the --parent option to specify the backup identifier of a previous full base backup or incremental backup that is to be used to set the starting point of changes that are to be saved in the incremental backup.
 - For more information, see Section 5.4.3 of the Backup and Recovery documentation.
 - Retention policies apply to incremental backups based on the parent, full base backup of the backup chain.
 - For more information, see Section 5.2.5 of the Backup and Recovery documentation.
- A new BART CHECK-CONFIG subcommand now provides validation of the BART configuration file and proper setup of the database servers.
 - This command checks that proper settings of parameters in both the global section and the individual database server sections. Using the CHECK-CONFIG subcommand helps avoidance of unexpected errors before starting the BART backup process.
 - For more information, see sections 5.1 and 5.4.1 of the Backup and Recovery documentation
- New BART configuration file options
 - All parameter names in the BART configuration file now accept only the underscore character (_) instead of the hyphen character (-) to separate multiple words in the parameter name. For example, bart-host is now bart_host.
 - For the list of configuration parameters, see sections 4.1 and 4.2.5 of the Backup and Recovery documentation.
 - The BART configuration file now contains the cluster_owner parameter, which must specify the Linux operating system user account that owns the directory of the database cluster to be saved in a backup.
 - For more information, see Section 4.2.5 of the Backup and Recovery documentation
 - The BART configuration file now contains the thread_count parameter, which sets the number of block copier threads for collecting modified blocks from a database server on which an incremental backup is taken.
 - For more information, see Section 4.2.5 of the Backup and Recovery documentation.
 - The BART configuration file now contains the batch_size parameter, which specifies the number of blocks of memory used for collecting modified blocks from a database server on which an incremental backup is taken.

- For more information, see Section 4.2.5 of the Backup and Recovery documentation.
 - The BART configuration file now contains the `scanner_logfile` parameter, which specifies the log file to record output from the WAL scanner program `bart-scanner`. This separates the WAL scanner results from all other BART subcommands that are invoked by the `bart` program.
 - For more information, see Section 4.1 of the Backup and Recovery documentation.
- Full base backups of database clusters with user-defined tablespaces can now be taken in plain text format as well as tar format.
 - In previous BART versions, this could only be accomplished in tar format. The backup of the tablespaces is stored in the same BART backup catalog subdirectory along with the backup of the main database cluster. Transaction log streaming (BART configuration file parameter `xlog_method = stream`) can now be enabled on full base backups for database clusters containing user-defined tablespaces when the backup is performed in plain text format.
 - For more information, see Section 4.2.4 of the Backup and Recovery documentation.

3 Installers and Documentation

EDB Postgres Backup Recovery 2.0 (`edb-bart20`) is packaged and delivered as an RPM via the EDB `yum.enterprisedb.com` system.

Please contact your EnterpriseDB Account Manager or sales@enterprisedb.com to request the location and credentials for the EDB Yum Repository if you do not already have them.

For more information, see Section 3 of the Backup and Recovery documentation.

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

4 Platform Support and System Requirements

EDB Backup and Recovery 2.0 supports usage for Postgres database server version 9.5 or later. Postgres database server version 9.4 or earlier is not supported by Backup and Recovery 2.0.

EDB Postgres Backup and Recovery is supported on RHEL/CentOS/OEL 7 & 6

platforms.

Additional details on supported platforms is available on the EnterpriseDB website:

<https://www.enterprisedb.com/services/edb-supported-products-and-platforms>

5 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)