



# **Backup and Recovery**

*Release 2.5.1*

**QuickStart Guide**

Oct 15, 2019

---

# Contents

---

<b>1</b>	<b>Installation</b>	<b>2</b>
<b>2</b>	<b>Configuring BART</b>	<b>4</b>
<b>3</b>	<b>Configuring a Database Server</b>	<b>8</b>
3.1	Authorizing SSH/SCP Access . . . . .	8
3.2	Setting up a Replication Database User . . . . .	9
3.3	Enabling WAL Archiving . . . . .	9
3.4	Verifying Configuration Setting . . . . .	9
<b>4</b>	<b>Taking a Backup</b>	<b>11</b>

This document provides shortcuts that allow you to install and configure BART, and take a full and incremental backup of a database server quickly. Please note that your system may have requirements that are not addressed in this document. For detailed information about BART installation and configuration, see BART Installation and Upgrade Guide available at:

<https://www.enterprisedb.com/edb-docs/>

BART is supported on the following platforms (64 bit only):

- CentOS 6.x or 7.x
- RHEL 6.x or 7.x
- PPC-LE 8 running RHEL or CentOS 7.x
- Ubuntu 18.04 (Bionic)
- Debian 9.x (Stretch)

BART works with the following database versions:

- Advanced Server versions 9.5, 9.6, 10, 11, and 12.
- PostgreSQL versions 9.5, 9.6, 10, 11, and 12.

Before installing BART, ensure that your repository configuration allows access to the EDB repository. For information about requesting credentials, visit:

<https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%202004-09-2019.pdf>

1. Then, use `yum` to create the repository configuration file:

```
yum -y install https://yum.enterprisedb.com/edb-repo-rpms/edb-repo-latest.noarch.rpm
```

2. Use your choice of editor to modify the repository configuration file (named `edb.repo`, located in `/etc/yum.repos.d`) ensuring that the value of the `enabled` parameter is 1, and replacing the `username` and `password` placeholders in the `baseurl` specification with the name and password of a registered EnterpriseDB user.
3. Before installing other software, use `yum` to install the `epel-release` package:

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

---

**Note:** You must enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum.repos.d`).

---

4. Then, use `yum` to install an Advanced Server or PostgreSQL database on any server on which an incremental backup will be restored. The host of the BART server is not required to have an installation of Postgres, but must include a copy of the following:
  - Postgres `libpq` library
  - Postgres `pg_basebackup` utility program
  - Boost Libraries version 1.48 and 1.53 (for RHEL/CentOS 6 and RHEL/CentOS 7 only).
5. Enable and activate Secure Shell (SSH) and Secure Copy (SCP) client programs on the BART host as well as on the remote database server hosts. The BART host and target database server host must accept a password-less SSH/SCP login connection.

After meeting the prerequisites, you can install the BART RPM package directly from the EnterpriseDB yum repository with the following command:

```
yum install edb-bart
```

Repeat the installation process described in this section to install BART 2.4 on all remote hosts where incremental backups are to be restored.

BART is installed in the `/usr/edb/bart` directory location referred to as *BART\_HOME*.

---

## Configuring BART

---

To configure the BART host and each database server that is to be managed by BART you must:

1. Establish the BART user account and ensure it runs the `bart` and the `bart-scanner` program.
2. Use the `bart.cfg.sample` file to create the `bart.cfg` file. The BART configuration file is located in `BART_HOME/etc/bart.cfg`:

```
cp bart.cfg.sample bart.cfg
```

3. Set the environment variable for the BART user account. If the `libpq` library does not reside in the default installation location, you must add the `libpq` library to the `LD_LIBRARY_PATH` environment variable and place the following settings in the BART user account's profile so they take effect upon login:

```
# .bash\_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
. ~/.bashrc
fi

# User specific environment and startup programs

export LD_LIBRARY_PATH=/usr/edb/as11/lib:$LD_LIBRARY_PATH
```

4. Set the following parameters in the `[BART]` section of the BART configuration file:

**bart\_host (required)**

Specify this parameter value in the form of `<bart_user>@<bart_host_address>`.

**backup\_path (required)**

Create the BART backup catalog and specify the path to the file system parent directory where BART database server backups and archived WAL files are stored. Ensure the BART user account owns the location specified in the `backup_path` parameter.

**pg\_basebackup\_path (required)**

Specify the path to the `pg_basebackup` program on the BART host.

**xlog\_method (optional)**

Set to `fetch` (default) to collect the transaction log files after the backup is completed. Set to `stream` to stream the transaction log in parallel with the full backup creation.

**retention\_policy (optional)**

Specify when an active backup should be marked as obsolete when the `MANAGE` subcommand is used. Specify `max_number BACKUPS` (default setting), `max_number DAYS`, `max_number WEEKS`, or `max_number MONTHS` where `max_number` is a positive integer.

**wal\_compression (optional)**

Set to `enabled` to compress the archived WAL files in `gzip` format when the `MANAGE` subcommand is invoked. Please note: The `gzip` compression program must be in the BART user account's `PATH`.

**copy\_wals\_during\_restore (optional)**

Set to `enabled` to copy the archived WAL files from the BART backup catalog to the `restore_path/archived_wals` directory prior to the database server archive recovery.

**logfile (optional)**

Use `logfile` to specify the path to the location to which output from the `bart` program is written. The log file will be created the first time you invoke the `bart` command using the sample configuration file value.

**scanner\_logfile (optional)**

Use `scanner_logfile` to specify the path to the location to which output from the `bart-scanner` program is written. The scanner log file will be created the first time you invoke the `bart` command using the sample configuration file value.

**thread\_count (optional)**

Specify the number of worker threads to copy blocks from the database server to the BART backup catalog when the `BACKUP` subcommand is invoked for incremental backups.

When taking a full backup, if the thread count is 1, then the `pg_basebackup` utility is used to take the full backup unless the `--no-pg_basebackup` option is specified with the `BACKUP` subcommand.

**batch\_size (optional)**

Specify the number of blocks of memory used for copying modified blocks from the database server to the BART backup catalog when the `BACKUP` subcommand is invoked for incremental backups. The maximum permitted value is 131072 ( $131072 * 8192 = 1 \text{ GB}$ ). The minimum permitted value is 1 ( $1 * 8192 = 8192 \text{ bytes}$ ).

**scan\_interval (optional)**

Specify the number of seconds before forcing a scan of the WAL files in the archive directory of the BART backup catalog. The default value is 0, which means no brute-force scanning will be started.

**mbm\_scan\_timeout (optional)**

Specify the number of seconds to wait for MBM files before timing out; the default value is 20 seconds. The `mbm_scan_timeout` parameter value must be greater than 0. The `mbm_scan_timeout` parameter is applicable only for incremental backup.

**workers (optional)**

Specify the number of parallel worker processes required to stream the modified blocks of an incremental backup to the restore host. The default value is 1.

5. Invoke the `CHECK-CONFIG` subcommand omitting the `-s` option to check the parameter settings in the BART configuration file including `bart_host`, `backup_path`, and `pg_basebackup_path`.
6. Set the following parameters for each database server in the server section of the BART configuration file. The parameter setting in the server section overrides the setting in the global [BART] section for that particular database server.

**[ServerName] (required)**

Specify a database server name.

**backup\_name (optional)**

Specify user-friendly name for the backups of the database server.

**host (required)**

Specify the IP address of the database server to be configured for backup.

**port (optional)**

Specify the port number identifying the database server instance to be backed up. The default is port 5444.

**user (required)**

Specify the replication database user name used by BART to establish the connection to the database server for full backups.

**archive\_command (optional)**

When the `INIT` subcommand is used, the content and variables specified in the BART `archive_command` result in the `archive_command` string to be generated into the Postgres `archive_command` parameter in the `postgresql.auto.conf` file. The following information applies only to the BART `archive_command` parameter.

Enclose the command string within single quotes (').

If the `archive_command` parameter is omitted, it still results in its usage by the `INIT` subcommand with a setting of `'scp %p %h:%a/%f'` where `%p` is the path of the file to archive used by the Postgres archiving process, `%h` is replaced by the `bart_host` parameter setting, `%a` is replaced by the BART archive path, and `%f` is the archived file name used by the Postgres archiving process.

**cluster\_owner (required)**

Specify the Linux operating system user account that owns the database cluster. This is typically `enterprisedb` for Advanced Server clusters installed in compatible mode, or `postgres` for PostgreSQL or Advanced Server clusters installed in the PostgreSQL compatible mode.

**remote\_host (optional)**

Specify this parameter value in the form of `<remote_user>@<remote_host_address>` where `remote_user` is the user account on the target database server host and `remote_host_address` is the IP address of the remote host.

**tablespace\_path (optional)**

Specify the path to which tablespaces are to be restored in the `OID = tablespace_path; OID = tablespace_path...` format. If the backup is to be restored to a remote host



(specified by the `remote_host` parameter), then the tablespace must exist on the remote host.

**allow\_incremental\_backups (optional)**

Set to `enabled` to permit incremental backups.

**description (optional)**

Specify the database server description.

For detailed information about configuring the optional parameters in the server section, refer to the BART Installation and Upgrade Guide available at:

<https://www.enterprisedb.com/edb-docs/>

- `Retention_policy`
- `xlog_method`
- `wal_compression`
- `copy_wals_during_restore`
- `thread_count`
- `batch_size`
- `scan_interval`
- `mbm_scan_timeout`
- `workers`

---

## Configuring a Database Server

---

To configure a database server, you must:

1. Authorize SSH/SCP access to the server.
2. Create and configure a replication database user.
3. Enable WAL archiving of the server.
4. Verify the server configuration settings.

---

**Note:** You must authorize SSH/SCP access and set up a replication database user before restarting the database server with WAL archiving enabled.

---

### 3.1 Authorizing SSH/SCP Access

The following example enables SSH/SCP access on a CentOS 6.x host; similar (platform-specific) steps will apply to other platforms/versions.

1. First, enable the usage of public key authentication:
  - (a) In the SSH server daemon configuration file `/etc/ssh/sshd_config`, ensure that the following parameter is set to `yes` and is not commented out:

```
PubkeyAuthentication yes
```
  - (b) Reload the configuration file using the `service sshd reload`, `service sshd stop`, `service sshd start`, or `service sshd restart` command.

---

**Note:** If you get any SSH or SCP errors, examine the log file (`/var/log/secure`).

---

2. Then, execute the following command to create a password-less connection:

```
ssh-copy-id target_user@host_address
```

For more information about how to generate the authorized public key, see the *Authorized Public Keys Generation* section of the BART Installation and Upgrade Guide available at:

<https://www.enterprisedb.com/edb-docs>

## 3.2 Setting up a Replication Database User

To set up a replication database user:

1. Choose a database user to serve as the *replication database user* (a superuser) for each Postgres database server to be managed by BART.
2. Modify the `pg_hba.conf` file to allow the replication database user to access the `template1` database. Include the replication database user in the `pg_hba.conf` file as a `replication` connection if `pg_basebackup` is to be used for taking any backups.
3. Specify the replication database user for the database server in the BART configuration file in the `user` parameter.

## 3.3 Enabling WAL Archiving

To enable WAL archiving, set the following parameters in the `postgresql.conf` file for any database server for which BART is to perform a backup. Set:

- `wal_level` to `archive` for Postgres 9.5 or to `replica` for Postgres 9.6 or later.
- `archive_mode` to `on`.
- `archive_command` to copy the WAL files to the BART backup catalog.
- `max_wal_senders` to a value high enough to leave at least one session available for the backup. If the `xlog_method=stream` parameter setting is to be used by this server, the `max_wal_senders` setting must account for an additional session for transaction log streaming.

---

**Note:** Run the `INIT` subcommand with the `-o` option to override any existing `archive_command` setting in the `postgresql.conf` or the `postgresql.auto.conf` file.

---

1. After verifying that the full path of the BART backup catalog has been created, restart the database server to initiate WAL archiving.
2. Start the WAL scanner by executing the following command:

```
./bart-scanner
```

## 3.4 Verifying Configuration Setting

- Use the `CHECK-CONFIG` subcommand with the `-s` option to verify the parameter settings in the database server configuration for which the `-s` option is specified.

```
bart CHECK-CONFIG [ -s <server_name> ]
```

In addition, the following `postgresql.conf` parameters for the database server must be properly set and activated for certain processes:

- The `cluster_owner` parameter must be set to the user account owning the database cluster directory.
- A password-less SSH/SCP connection must be set between the BART user and the user account specified by the `cluster_owner` parameter.
- The BART `user` parameter must specify a database superuser.
- The `pg_hba.conf` file must contain a replication entry for the database superuser specified by the BART `user` parameter.
- The `archive_mode` parameter in the `postgresql.conf` file must be enabled.
- The `archive_command` parameter in the `postgresql.auto.conf` or the `postgresql.conf` file must be set.
- The `allow_incremental_backups` parameter in the BART configuration file must be enabled for database servers for which incremental backups are to be taken.
- Archiving of WAL files to the BART backup catalog must be in process.
- The WAL scanner program must be running.

---

## Taking a Backup

---

This section provides information about creating a full or incremental backup of a database server. For detailed information about taking a full backup, incremental backup, point-in-time recovery and restore process, see BART User Guide available at:

<https://www.enterprisedb.com/edb-docs>

The syntax of the `BACKUP` subcommand is:

```
bart BACKUP -s { server_name | all }
[ -F { p | t } ]
[ -z ] [ -c compression_level ]
[ --parent { backup_id | backup_name } ]
[ --backup-name backup_name ]
[ --thread-count number_of_threads ]
[ { --with-pg_basebackup | --no-pg_basebackup } ]
[ --check ]
```

---

**Note:** While a `BACKUP` subcommand is in progress, no other processes may run in parallel.

---

Along with the `BACKUP` subcommand, you can:

- Specify the `-s` option and replace the *server\_name* with the server name to be backed up (it must be configured in the BART configuration file). Specify `all` to take a backup of all servers. This argument is mandatory.

The backup is saved in the *backup\_path/server\_name/backup\_id* directory.

Specify the following options only if required. If you do not specify any of the following options, the backup is created using the default settings.

- Specify the `-F p` option to create a backup in the plain text format and `-F t` to create a backup in tar format (default). If the transaction log streaming method is used, then the `-F p` option must be specified.
- Specify the `-z` option to use `gzip` compression on the tar file output using the default compression level. This option is applicable only for the tar format.

- Specify the `-c` option to apply the gzip compression level on the tar file output, and replace *compression\_level* with the digit 1 through 9, with 9 being the best compression (applicable only for the tar format).
- If you want to take an incremental backup, specify the option `--parent` and replace *backup\_id* with the backup identifier of a parent backup or replace *backup\_name* with the parent backup name. Incremental backup can only be taken in the plain text format (`-F p`). Specify the option `--check` before taking an incremental backup to verify if the required MBM files are present in the BART backup catalog. The `--parent` option must be specified when the `--check` option is used.
- Specify the option `--backup-name` and replace *backup\_name* with the user-friendly name assigned to the backup.
- Specify the option `-thread count` and replace *number\_of\_threads* with the number of worker threads to run in parallel to copy blocks for incremental backups
- Specify the option `--with-pg_basebackup` to use `pg_basebackup` to take a full backup. The number of thread counts in effect is ignored as given by the `thread_count` parameter in the BART configuration file.

---

**Note:** If the thread count in effect is greater than 1, then the `pg_basebackup` utility is not used to take the full backup unless the `--with-pg_basebackup` option is specified with the `BACKUP` subcommand.

---

- Specify the option `--no pg_basebackup` to not use `pg_basebackup` to take a full backup.

The following example creates a full backup in the default tar format with gzip compression. Note that checksums are generated for the full backup and user-defined tablespaces for the tar format backup.

```
[edb@localhost bin]$ ./bart BACKUP -s hr -z
INFO:  DebugTarget - getVar(checkDiskSpace.bytesAvailable)
INFO:  new backup identifier generated 1567591909098
INFO:  creating 5 harvester threads
NOTICE: all required WAL segments have been archived
/home/edb/bkup_new/hr/1567591909098
INFO:  backup completed successfully
INFO:
BART VERSION: 2.5
BACKUP DETAILS:
BACKUP STATUS: active
BACKUP IDENTIFIER: 1567591909098
BACKUP NAME: none
BACKUP PARENT: none
BACKUP LOCATION: /home/edb/bkup_new/hr/1567591909098
BACKUP SIZE: 13.91 MB
BACKUP FORMAT: tar.gz
BACKUP TIMEZONE: America/New_York
XLOG METHOD: fetch
BACKUP CHECKSUM(s): 0
TABLESPACE(s): 3
Oid      Name      Location
16387   test1    /home/edb/tbl1
16388   test2    /home/edb/tbl2
16389   test3    /home/edb/tbl3

START WAL LOCATION: 0000000100000000000000025
STOP WAL LOCATION: 0000000100000000000000026
BACKUP METHOD: streamed
BACKUP FROM: master
START TIME: 2019-09-04 06:11:49 EDT
```

```
STOP TIME: 2019-09-04 06:11:53 EDT
TOTAL DURATION: 4 sec(s)
```

The following example shows an incremental backup taken by specifying the `--parent` option. The option `-F p` must be specified as well for plain text format.

```
[edb@localhost bin]$ ./bart BACKUP -s hr -F p --parent hr_full_1 --backup-name
hr_incr_1
INFO: DebugTarget - getVar(checkDiskSpace.bytesAvailable)
INFO: checking /home/edb/bkup_new/hr/archived_wals for MBM files from 0/20000028 to
0/22000000
INFO: new backup identifier generated 1566899827751
INFO: creating 5 harvester threads
NOTICE: all required WAL segments have been archived
INFO: backup completed successfully
INFO:
BART VERSION: 2.5
BACKUP DETAILS:
BACKUP STATUS: active
BACKUP IDENTIFIER: 1566899827751
BACKUP NAME: hr_incr_1
BACKUP PARENT: 1566899819709
BACKUP LOCATION: /home/edb/bkup_new/hr/1566899827751
BACKUP SIZE: 7.19 MB
BACKUP FORMAT: plain
BACKUP TIMEZONE: America/New_York
XLOG METHOD: fetch
BACKUP CHECKSUM(s): 0
TABLESPACE(s): 0
START WAL LOCATION: 0000000100000000000000022
STOP WAL LOCATION: 0000000100000000000000023
BACKUP METHOD: streamed
BACKUP FROM: master
START TIME: 2019-08-27 05:57:07 EDT
STOP TIME: 2019-08-27 05:57:08 EDT
TOTAL DURATION: 1 sec(s)
```