How to Enable Audit Logging with EDB Postgres™ Advanced Server

Version 9.5

May 2, 2016

How to Enable Audit Logging with EDB Postgres Advanced Server
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

Notice: The names for EDB's products have changed.

> *The product formerly referred to as Postgres Plus Advanced Server is now referred to as EDB Postgres Advanced Server (Advanced Server).*
>
> *The product formerly referred to as Postgres Enterprise Manager (PEM) is now referred to as EDB Postgres Enterprise Manager (EDB Enterprise Manager).*

Until a new version of this documentation is published, wherever you see an earlier version of a product name, you may substitute it with the current name. Name changes in software and software outputs will be phased in over time.

EDB Postgres Advanced Server adds extended functionality to the open-source PostgreSQL database. The extended functionality supports database administration, enhanced SQL capabilities, database and application security, performance monitoring and analysis, and application development utilities.

Advanced Server allows database and security administrators, auditors, and operators to track and analyze database activities using audit logs. Audit logs can be configured to include:

- When a role establishes a connection to an Advanced Server database.
- When a role disconnects from an Advanced Server database.
- When database objects are created, modified or deleted from a database.
- When queries are performed against the database.
- When failed authentication attempts occur.
- When failed attempts to perform unauthorized activities occur.

This tutorial documents how to create and review an audit log. The path names and commands referenced in the examples are for Advanced Server hosts that reside on a CentOS 6.5 host – you may have to modify paths and commands for your configuration.

1.1 Typographical Conventions Used in this Guide

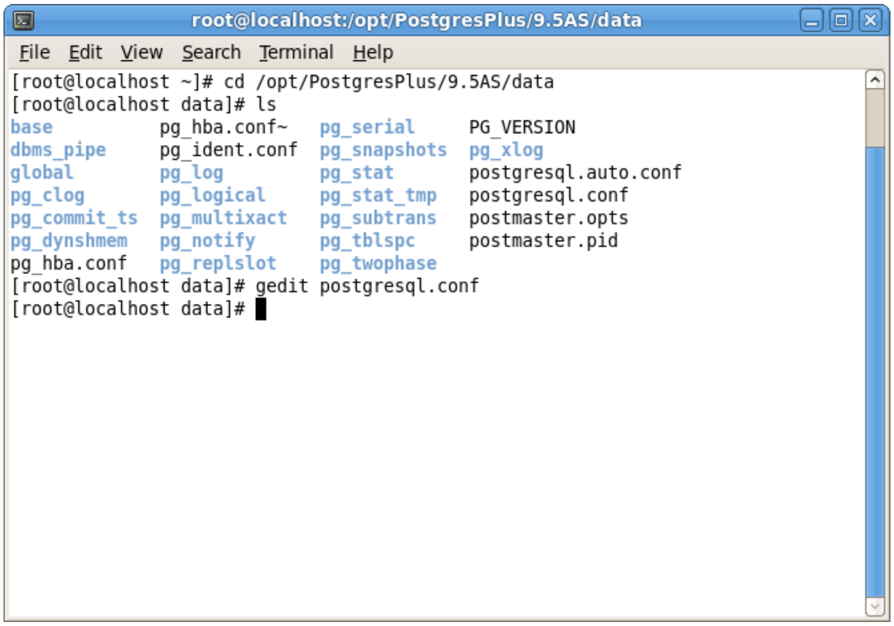
Certain typographical conventions are used in this manual to clarify the meaning and usage of various commands, statements, programs, examples, etc. This section provides a summary of these conventions.

In the following descriptions a *term* refers to any word or group of words that are language keywords, user-supplied values, literals, etc. A term's exact meaning depends upon the context in which it is used.

- *Italic font* introduces a new term, typically, in the sentence that defines it for the first time.
- `Fixed-width (mono-spaced) font` is used for terms that must be given literally such as SQL commands, specific table and column names used in the examples, programming language keywords, etc. For example, `SELECT * FROM emp;`
- *`Italic fixed-width font`* is used for terms for which the user must substitute values in actual usage. For example, `DELETE FROM` *`table_name`*;
- A vertical pipe | denotes a choice between the terms on either side of the pipe. A vertical pipe is used to separate two or more alternative terms within square brackets (optional choices) or braces (one mandatory choice).
- Square brackets [ ] denote that one or none of the enclosed terms may be substituted. For example, `[ a | b ]` means choose one of "`a`" or "`b`" or neither of the two.
- Braces {} denote that exactly one of the enclosed alternatives must be specified. For example, `{ a | b }` means exactly one of "`a`" or "`b`" must be specified.
- Ellipses ... denote that the preceding term may be repeated. For example, `[ a | b ] ...` means that you may have the sequence, "`b a a b a`".

2 Modifying the postgresql.conf File

If the server is configured to save audit logs, auditing will start each time the server starts. Configuration parameters that reside in the `postgresql.conf` file specify the auditing behavior of the server. The `postgresql.conf` file is located in the `data` directory under your PostgreSQL installation (see Figure 2.1).


root@localhost:/opt/PostgresPlus/9.5AS/data
File Edit View Search Terminal Help

```
[root@localhost ~]# cd /opt/PostgresPlus/9.5AS/data
[root@localhost data]# ls
base           pg_hba.conf~   pg_serial      PG_VERSION
dbms_pipe      pg_ident.conf  pg_snapshots   pg_xlog
global         pg_log         pg_stat        postgresql.auto.conf
pg_clog        pg_logical     pg_stat_tmp    postgresql.conf
pg_commit_ts   pg_multixact   pg_subtrans    postmaster.opts
pg_dynshmem    pg_notify      pg_tblspc      postmaster.pid
pg_hba.conf    pg_replslot    pg_twophase
[root@localhost data]# gedit postgresql.conf
[root@localhost data]# 
```

Figure 2.1 – Opening the postgresql.conf file.

If you've performed your installation with the graphical installer on Linux, the `postgresql.conf` file is located in:

```
/opt/PostgresPlus/9.5AS/data
```

If you've performed your installation with the graphical installer on Windows, the `postgresql.conf` file is located in:

```
C:\ProgramFiles\PostgresPlus\9.5AS\data
```

If you've used an RPM installer to install Advanced Server, the `postgresql.conf` file is located in:

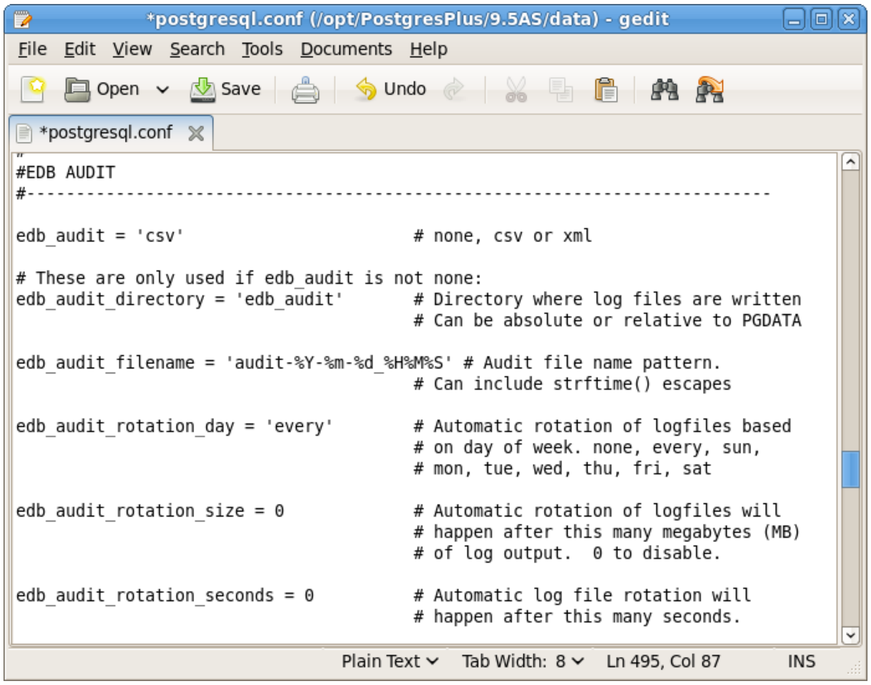
```
/var/lib/ppas/9.5/data
```

2.1 Enabling Auditing

To enable auditing, open the `postgresql.conf` file with an editor of your choice. To enable a parameter, remove the pound sign (#) from in front of the parameter, and specify a parameter setting to the right of the equal sign (=).

To enable auditing, modify the `postgresql.conf` file, changing the following parameters in the `EDB Audit` section (see Figure 2.2):

- To enable database auditing, remove the # sign in front of `edb_audit` and change the `'none'` value to `'csv'`. This specifies that the log will be written to a .csv file format.
- Remove the # sign in front of `edb_audit_directory`. Accept the default directory, or you can change where the `edb_audit_directory` locates files by changing `'edb_audit'` to a file name of your choice ('*path*').
- Remove the # sign in front of `edb_audit_filename` to accept the default filename which includes a timestamp. The escape sequences (`%Y`, `%m`, etc.) will be replaced by the appropriate current values according to the system date and time.
- Remove the # sign in front of `edb_audit_rotation_day`, `edb_audit_rotation_size`, and `edb_audit_rotation_seconds` to control audit log file rotation.



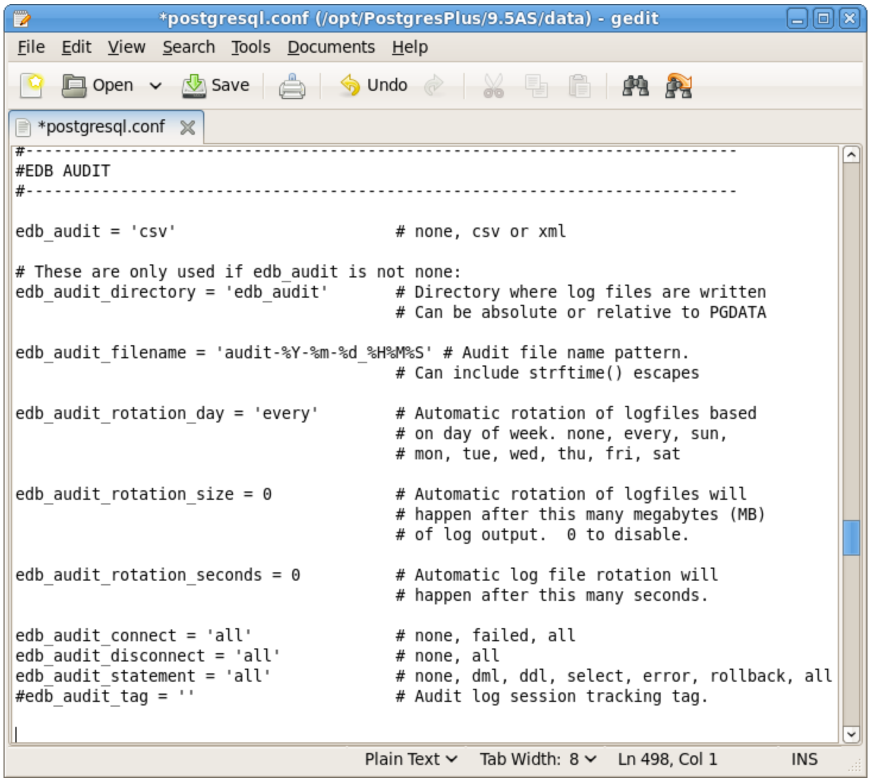
```
#EDB AUDIT
#------------------------------------------------------------------------------

edb_audit = 'csv'                       # none, csv or xml

# These are only used if edb_audit is not none:
edb_audit_directory = 'edb_audit'       # Directory where log files are written
                                        # Can be absolute or relative to PGDATA

edb_audit_filename = 'audit-%Y-%m-%d_%H%M%S' # Audit file name pattern.
                                        # Can include strftime() escapes

edb_audit_rotation_day = 'every'        # Automatic rotation of logfiles based
                                        # on day of week. none, every, sun,
                                        # mon, tue, wed, thu, fri, sat

edb_audit_rotation_size = 0             # Automatic rotation of logfiles will
                                        # happen after this many megabytes (MB)
                                        # of log output.  0 to disable.

edb_audit_rotation_seconds = 0          # Automatic log file rotation will
                                        # happen after this many seconds.
```

Figure 2.2 – Modifying the postgresql.conf file.

2.2 Specifying What to Log

Use parameters within the `EDB AUDIT` section of the `postgresql.conf` file to specify the type of information included in the audit logs. The following example logs all attempts at connecting and disconnecting, and all statements performed against the database. (see Figure 2.3):

- Remove the pound sign in front of `edb_audit_connect` and change the value to `'all'` to specify that all connection attempts will be logged.
- Remove the pound sign in front of `edb_audit_disconnect` and change the value to `'all'` to specify that all disconnection attempts will be logged.
- Remove the pound sign in front of `edb_audit_statement` and change the value to `'all'` to specify that all queries (`SELECT` statements), all DML statements (`INSERT, UPDATE, DELETE`), and all DDL statements (`CREATE, DROP, ALTER`) will be logged.


*postgresql.conf (/opt/PostgresPlus/9.5AS/data) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
*postgresql.conf
Plain Text
Tab Width: 8
Ln 498, Col 1
INS

```
#---------------------------------------------------------------------------
#EDB AUDIT
#---------------------------------------------------------------------------

edb_audit = 'csv'                       # none, csv or xml

# These are only used if edb_audit is not none:
edb_audit_directory = 'edb_audit'       # Directory where log files are written
                                        # Can be absolute or relative to PGDATA

edb_audit_filename = 'audit-%Y-%m-%d_%H%M%S' # Audit file name pattern.
                                        # Can include strftime() escapes

edb_audit_rotation_day = 'every'        # Automatic rotation of logfiles based
                                        # on day of week. none, every, sun,
                                        # mon, tue, wed, thu, fri, sat

edb_audit_rotation_size = 0             # Automatic rotation of logfiles will
                                        # happen after this many megabytes (MB)
                                        # of log output.  0 to disable.

edb_audit_rotation_seconds = 0          # Automatic log file rotation will
                                        # happen after this many seconds.

edb_audit_connect = 'all'               # none, failed, all
edb_audit_disconnect = 'all'            # none, all
edb_audit_statement = 'all'             # none, dml, ddl, select, error, rollback, all
#edb_audit_tag = ''                     # Audit log session tracking tag.
```

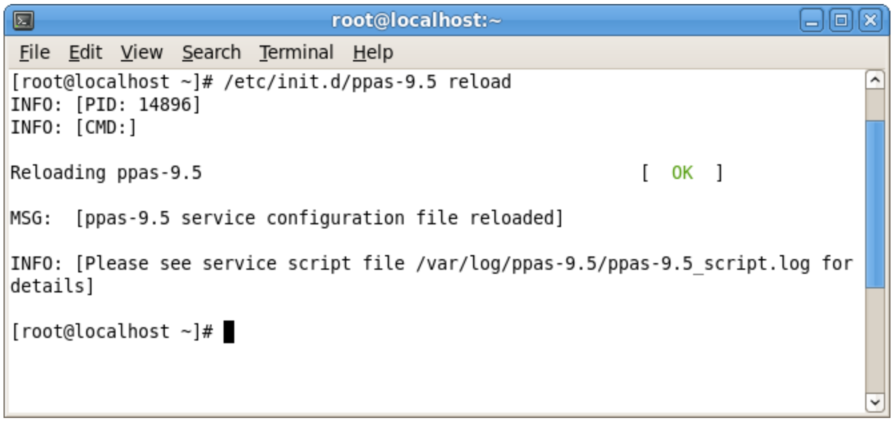
Figure 2.3 – Logging parameters.

Please note that the above parameters settings serve as a convenient example, but may store more information than required. You should evaluate your own auditing needs, and adjust the parameter settings accordingly to avoid storing excessively large audit files. To keep audit logs manageable, consistent log file maintenance and rotation is essential.

For more information about EDB audit logging parameters, see the Appendix.

2.3 Saving and Applying Modifications

To instruct the server to apply the changes that you've made to the auditing parameters, you must reload the server parameters. Reloading configuration parameters does not require Advanced Server users to log out of their current Advanced Server sessions. Save the `postgresql.conf` file, and reload the configuration parameters at the command line (see Figure 2.4), or use the `Reload Configuration` option on the menu bar (see Figure 2.5).



```
[root@localhost ~]# /etc/init.d/ppas-9.5 reload
INFO: [PID: 14896]
INFO: [CMD:]

Reloading ppas-9.5                                        [  OK  ]

MSG:  [ppas-9.5 service configuration file reloaded]

INFO: [Please see service script file /var/log/ppas-9.5/ppas-9.5_script.log for
details]

[root@localhost ~]# 
```

Figure 2.4 – Reloading Advanced Server with the service command.

To reload the system configuration parameter values through the `Applications` (or Windows `Start`) menu, navigate through `Postgres Plus Advanced Server` menu to the `Expert Configuration` menu, and select `Reload Configuration`.

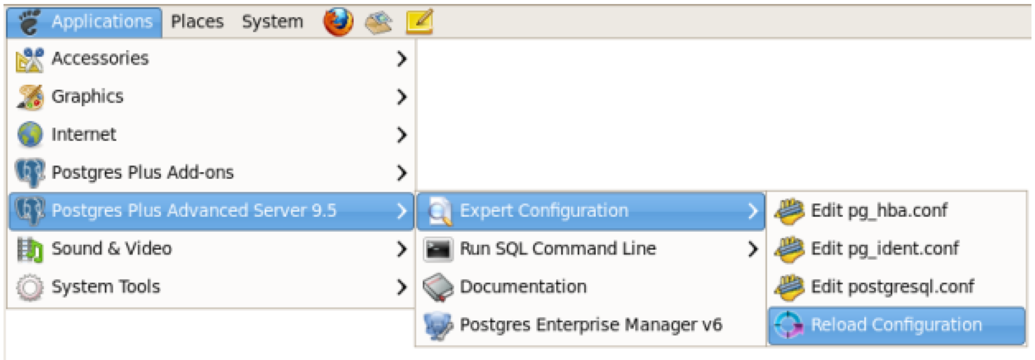


Figure 2.5 – Reloading configuration parameter values through the Applications menu.

If prompted, provide a password to reload the parameters.

3 Viewing Audit Logs

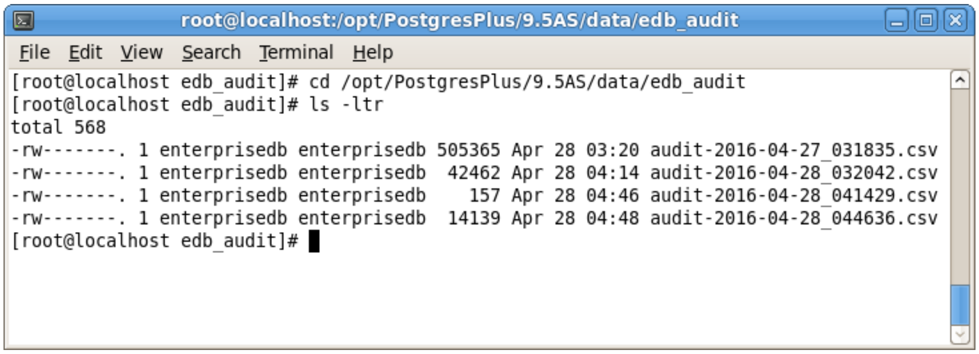
You can view the audit logs with your choice of the text editor. By default, on a Linux system (installed with the graphical installer) the log files are written to:

```
/opt/PostgresPlus/9.5AS/data/edb_audit
```

On a Windows system, the audit logs are stored in:

```
C:\ProgramFiles\PostgresPlus\9.5AS\data\edb_audit
```

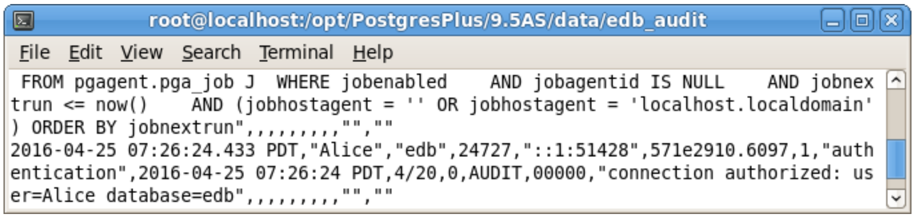
The time and date that the log file was created is part of the log file name; the most recent entries will be located in the file with the most recent date/time in their file name (see Figure 3.1).


root@localhost:/opt/PostgresPlus/9.5AS/data/edb_audit
File Edit View Search Terminal Help

```
[root@localhost edb_audit]# cd /opt/PostgresPlus/9.5AS/data/edb_audit
[root@localhost edb_audit]# ls -ltr
total 568
-rw-------. 1 enterprisedb enterprisedb 505365 Apr 28 03:20 audit-2016-04-27_031835.csv
-rw-------. 1 enterprisedb enterprisedb  42462 Apr 28 04:14 audit-2016-04-28_032042.csv
-rw-------. 1 enterprisedb enterprisedb    157 Apr 28 04:46 audit-2016-04-28_041429.csv
-rw-------. 1 enterprisedb enterprisedb  14139 Apr 28 04:48 audit-2016-04-28_044636.csv
[root@localhost edb_audit]# 
```

Figure 3.1 – Viewing a list of audit files ordered by date and time.

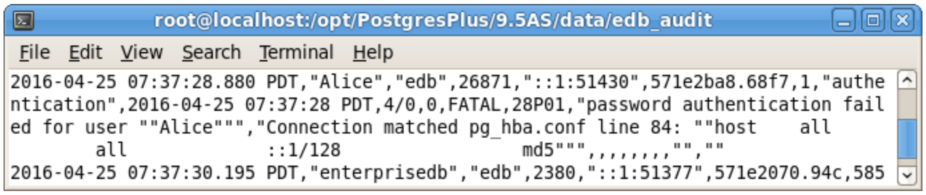
Entries in the audit logs provide an overview of the activity on the database server. If, for example, a `Login Role` named `Alice` logs in and provides the correct password, the connection information is added to the audit log (see Figure 3.2) because `all` connection attempts are logged as specified in the `postgresql.conf` file.


root@localhost:/opt/PostgresPlus/9.5AS/data/edb_audit
File Edit View Search Terminal Help

```
 FROM pgagent.pga_job J  WHERE jobenabled    AND jobagentid IS NULL    AND jobnex
trun <= now()    AND (jobhostagent = '' OR jobhostagent = 'localhost.localdomain'
) ORDER BY jobnextrun",,,,,,,,,"",""
2016-04-25 07:26:24.433 PDT,"Alice","edb",24727,"::1:51428",571e2910.6097,1,"auth
entication",2016-04-25 07:26:24 PDT,4/20,0,AUDIT,00000,"connection authorized: us
er=Alice database=edb",,,,,,,,,"",""
```

Figure 3.2 – Viewing connection information for a Login Role.

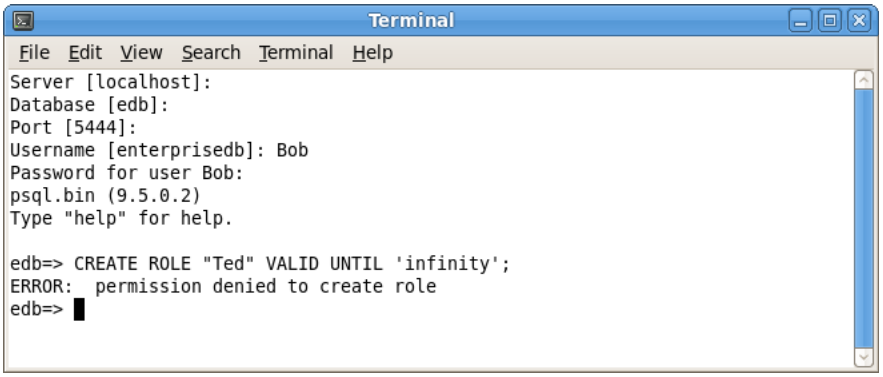
If `Alice` provides the wrong password, authentication fails and the failed connection attempt is recorded in the audit log (see Figure 3.3).


root@localhost:/opt/PostgresPlus/9.5AS/data/edb_audit
File Edit View Search Terminal Help

```
2016-04-25 07:37:28.880 PDT,"Alice","edb",26871,"::1:51430",571e2ba8.68f7,1,"authe
ntication",2016-04-25 07:37:28 PDT,4/0,0,FATAL,28P01,"password authentication fail
ed for user ""Alice""","Connection matched pg_hba.conf line 84: ""host    all
        all             ::1/128                 md5""",,,,,,,,"",""
2016-04-25 07:37:30.195 PDT,"enterprisedb","edb",2380,"::1:51377",571e2070.94c,585
```

Figure 3.3 – Audit log records unauthorized connection information.

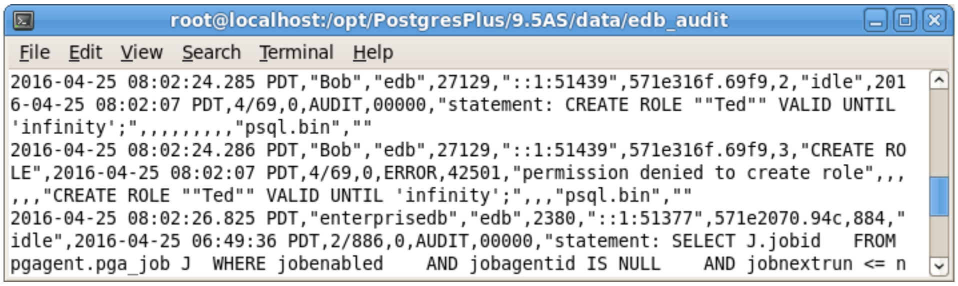
If a `Login Role` named `Bob` (who has `CREATE DATABASE` privileges but does not have `CREATE ROLE` privileges) attempts to create a new `Login Role` named `Ted` (see Figure 3.4):



```
Server [localhost]:
Database [edb]:
Port [5444]:
Username [enterprisedb]: Bob
Password for user Bob:
psql.bin (9.5.0.2)
Type "help" for help.

edb=> CREATE ROLE "Ted" VALID UNTIL 'infinity';
ERROR:  permission denied to create role
edb=> █
```

Figure 3.4 – Example of an unauthorized command.

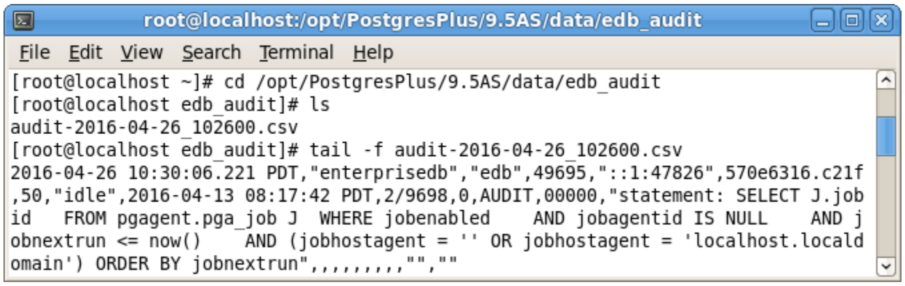
The attempt to create the new role fails because `Bob` did not have sufficient privileges. The audit log records the failed attempt (see Figure 3.5):


root@localhost:/opt/PostgresPlus/9.5AS/data/edb_audit
File Edit View Search Terminal Help

```
2016-04-25 08:02:24.285 PDT,"Bob","edb",27129,"::1:51439",571e316f.69f9,2,"idle",201
6-04-25 08:02:07 PDT,4/69,0,AUDIT,00000,"statement: CREATE ROLE ""Ted"" VALID UNTIL
'infinity';",,,,,,,,,"psql.bin",""
2016-04-25 08:02:24.286 PDT,"Bob","edb",27129,"::1:51439",571e316f.69f9,3,"CREATE RO
LE",2016-04-25 08:02:07 PDT,4/69,0,ERROR,42501,"permission denied to create role",,,
,,,"CREATE ROLE ""Ted"" VALID UNTIL 'infinity';",,,"psql.bin",""
2016-04-25 08:02:26.825 PDT,"enterprisedb","edb",2380,"::1:51377",571e2070.94c,884,"
idle",2016-04-25 06:49:36 PDT,2/886,0,AUDIT,00000,"statement: SELECT J.jobid   FROM
pgagent.pga_job J  WHERE jobenabled    AND jobagentid IS NULL    AND jobnextrun <= n
```

Figure 3.5 – Audit log records for the example above.

For debugging or monitoring purposes, the `tail -f` command can be useful; invoke the command on the most recent audit log to see current activity performed against the server (see Figure 3.6).


root@localhost:/opt/PostgresPlus/9.5AS/data/edb_audit
File Edit View Search Terminal Help

```
[root@localhost ~]# cd /opt/PostgresPlus/9.5AS/data/edb_audit
[root@localhost edb_audit]# ls
audit-2016-04-26_102600.csv
[root@localhost edb_audit]# tail -f audit-2016-04-26_102600.csv
2016-04-26 10:30:06.221 PDT,"enterprisedb","edb",49695,"::1:47826",570e6316.c21f
,50,"idle",2016-04-13 08:17:42 PDT,2/9698,0,AUDIT,00000,"statement: SELECT J.job
id   FROM pgagent.pga_job J  WHERE jobenabled    AND jobagentid IS NULL    AND j
obnextrun <= now()    AND (jobhostagent = '' OR jobhostagent = 'localhost.locald
omain') ORDER BY jobnextrun",,,,,,,,,"",""
```

Figure 3.6 – Viewing an audit log.

4 Audit Tagging

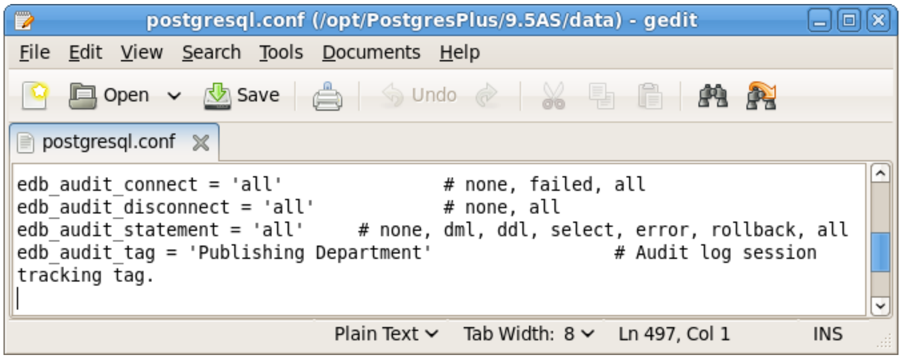
Audit tagging allows an application developer to associate an arbitrary string (for example an application user name or a task identifier meaningful to the application developer) with activities in the audit log. This provides additional detailed information about database activity and richer auditing capabilities.

The `edb_audit_tag` parameter provides the mechanism to store this additional information in the audit log. You can modify the audit tag:

- with a `SET` statement in an application. Typically an application may change an audit tag as it moves from task to task.
- in the `postgresql.conf` file. The tag specified in the `postgresql.conf` file will act as a default value, applied to the entire cluster.
- with the PEM or psql client. You can associate an audit tag with a function, procedure, or role with the `SET` clause of an `ALTER` or `CREATE` statement.

To add an audit tag in the server configuration file, open the `postgresql.conf` file with an editor of your choice. If prompted, provide your password.

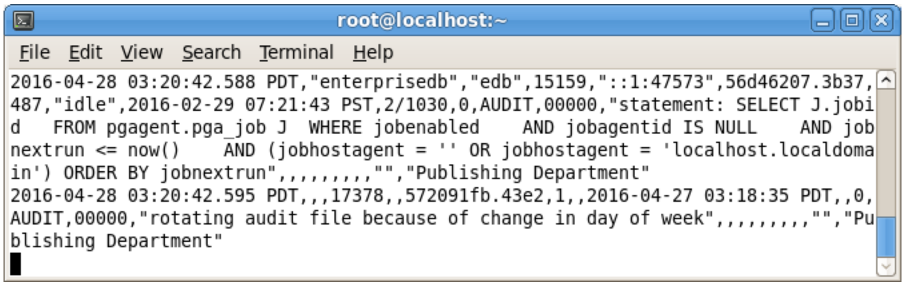
To enable audit tagging, remove the # sign in front of `edb_audit_tag` in the `EDB Audit` section of the `postgresql.conf` file. Enter a string value to the right of the = sign and inside the straight quotes (see Figure 4.1). In the example below, the data tag `Publishing Department` is added.



```
edb_audit_connect = 'all'              # none, failed, all
edb_audit_disconnect = 'all'           # none, all
edb_audit_statement = 'all'     # none, dml, ddl, select, error, rollback, all
edb_audit_tag = 'Publishing Department'                # Audit log session
tracking tag.
```

Figure 4.1 – Modifying the edb_audit_tag parameter.

When configuration parameters are reloaded, the specified audit tag will be added to audit log entries (see Figure 4.2).



```
2016-04-28 03:20:42.588 PDT,"enterprisedb","edb",15159,"::1:47573",56d46207.3b37,
487,"idle",2016-02-29 07:21:43 PST,2/1030,0,AUDIT,00000,"statement: SELECT J.jobi
d   FROM pgagent.pga_job J  WHERE jobenabled    AND jobagentid IS NULL    AND job
nextrun <= now()    AND (jobhostagent = '' OR jobhostagent = 'localhost.localdoma
in') ORDER BY jobnextrun",,,,,,,,,"","Publishing Department"
2016-04-28 03:20:42.595 PDT,,,17378,,572091fb.43e2,1,,2016-04-27 03:18:35 PDT,,0,
AUDIT,00000,"rotating audit file because of change in day of week",,,,,,,,,"","Pu
blishing Department"
```

Figure 4.2 – Audit log entries with audit tags.

You can also use the PEM client to associate an audit tag value with a role or database object. To start the PEM client, click `Postgres Enterprise Manager` from the `Postgres Plus Advanced Server` selection in the `Applications` menu (see Figure 4.3).

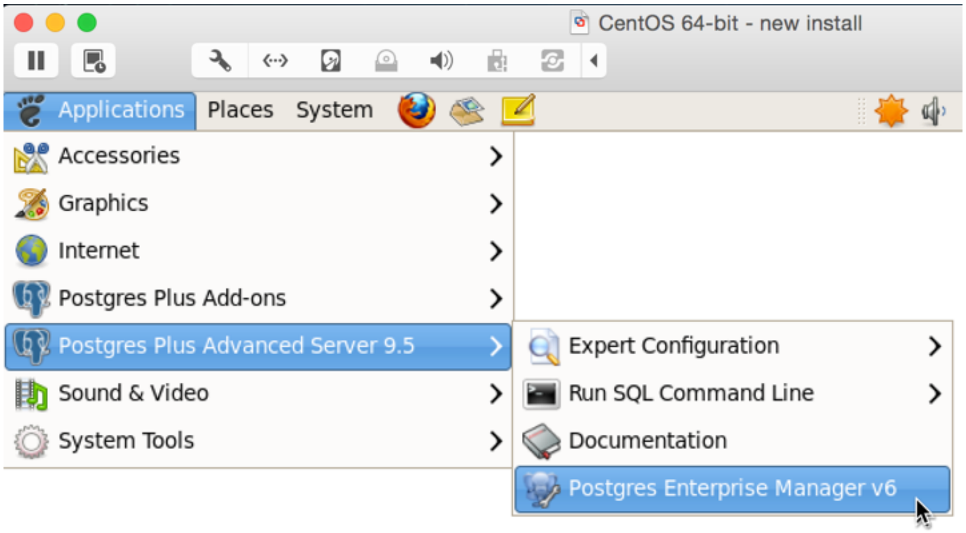
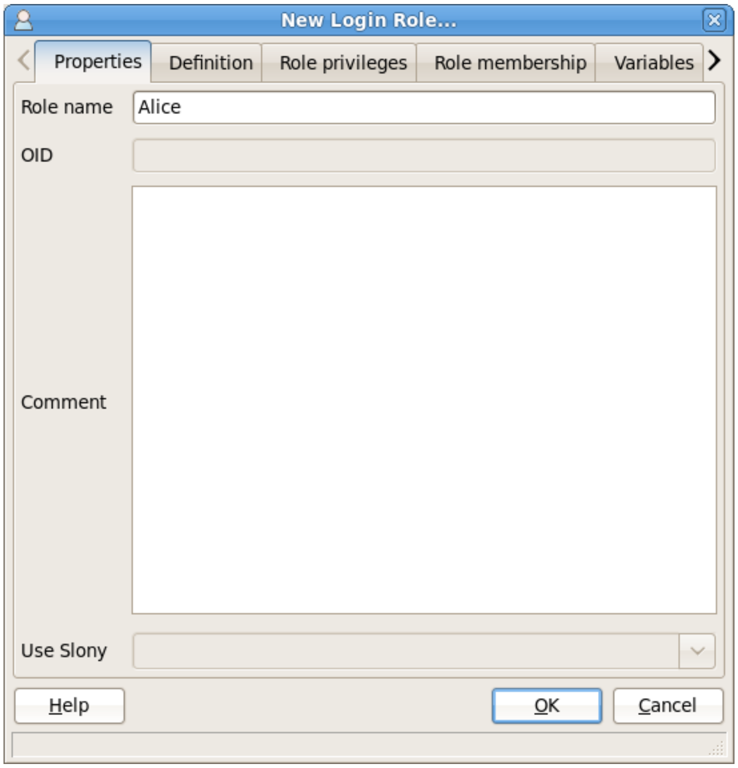


Figure 4.3 – Opening the PEM client.

When the PEM client opens, right-click `Postgres Plus Advanced Server` in the `Object browser` tree control and select `Connect` from the context menu to connect to the server. If prompted, provide your password for authentication.

After authenticating with the server, the server's node of the tree control will be populated with the objects that reside on that server.

To create a new role with an associated audit tag, right click the Login Role node of the PEM client's tree control, and select New Login Role... to launch the New Login Role dialog (see Figure 4.4).



Role name: Alice

Figure 4.4 – The New Login Role... dialog.

For example, you can use the following selections to create a role named Alice with an associated audit tag that displays Clerk, Human Resources:

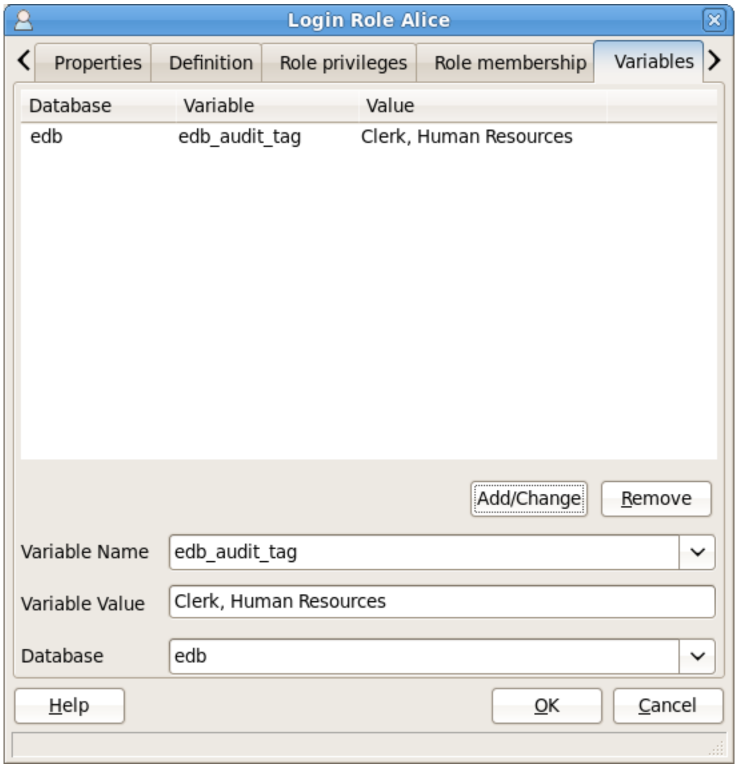
On the Properties tab, use the Role name field to add the name Alice.

On the Definition tab:

- Provide a password in the Password field.
- Reenter the password in the Password (again) field.

On the Variables tab,

- Use the drop-down listbox next to Variable Name to select edb_audit_tag.
- Use the Variable Value field to add Clerk, Human Resources.
- Use the drop-down listbox next to the Database field to select edb.
- Click Add to enable audit tagging for Alice (see Figure 4.5).



Database	Variable	Value
edb	edb_audit_tag	Clerk, Human Resources

Variable Name: edb_audit_tag

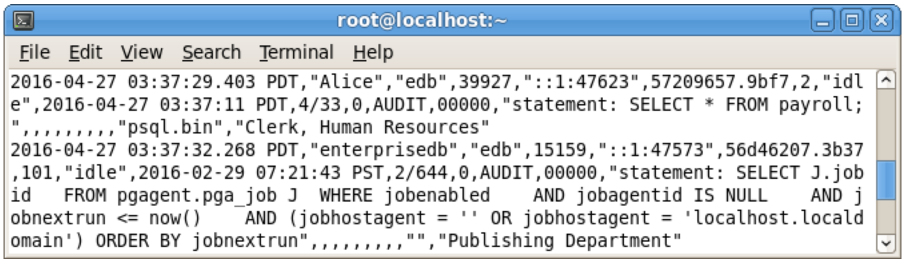
Variable Value: Clerk, Human Resources

Database: edb

Figure 4.5 – Adding an audit tag.

After making your selections on the `New Login Role…` dialog, click the `OK` button to create `Alice`.

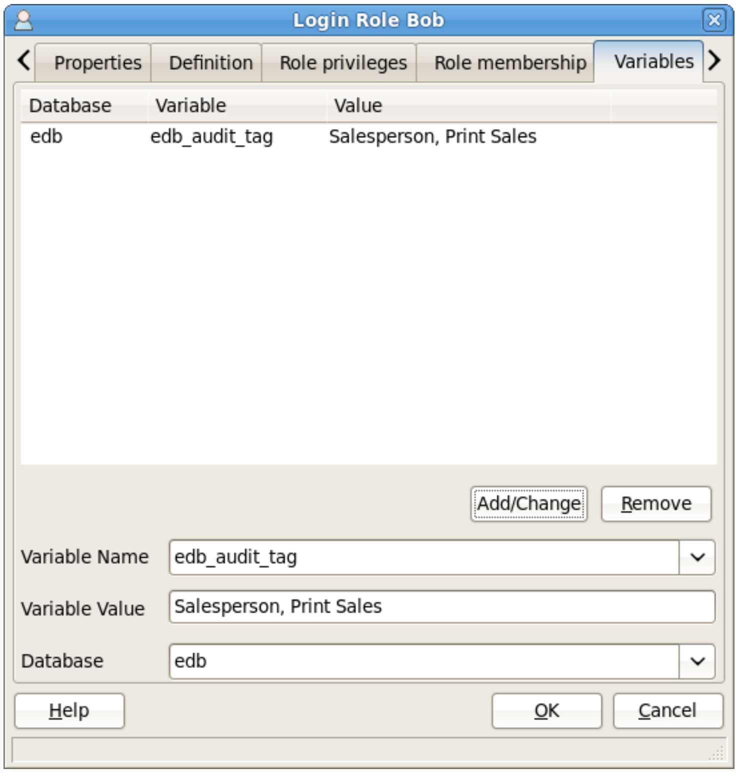
If `Alice` logs in and makes a statement, the audit log will record the activity and include the audit tag `Clerk, Human Resources` (see Figure 4.6). Notice the very next entry in the audit log includes the audit tag `Publishing Department`, which was added to the `postgresql.conf` file.



```
2016-04-27 03:37:29.403 PDT,"Alice","edb",39927,"::1:47623",57209657.9bf7,2,"idl
e",2016-04-27 03:37:11 PDT,4/33,0,AUDIT,00000,"statement: SELECT * FROM payroll;
",,,,,,,,,"psql.bin","Clerk, Human Resources"
2016-04-27 03:37:32.268 PDT,"enterprisedb","edb",15159,"::1:47573",56d46207.3b37
,101,"idle",2016-02-29 07:21:43 PST,2/644,0,AUDIT,00000,"statement: SELECT J.job
id   FROM pgagent.pga_job J  WHERE jobenabled    AND jobagentid IS NULL    AND j
obnextrun <= now()    AND (jobhostagent = '' OR jobhostagent = 'localhost.locald
omain') ORDER BY jobnextrun",,,,,,,,,"","Publishing Department"
```

Figure 4.6 – The audit log entry.

You can attach a different audit tag to a role named `Bob` (see Figure 4.7)



Database	Variable	Value
edb	edb_audit_tag	Salesperson, Print Sales

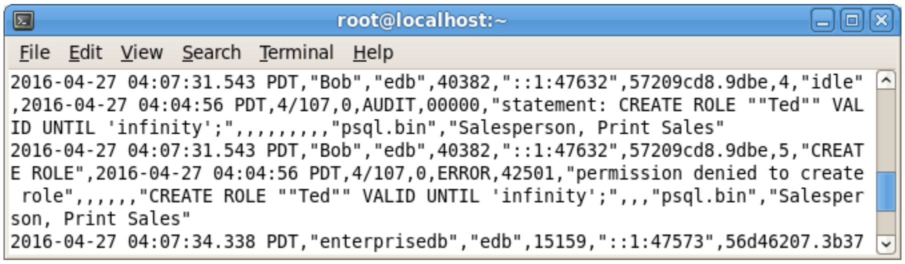
Variable Name: edb_audit_tag

Variable Value: Salesperson, Print Sales

Database: edb

Figure 4.7 – Adding a different audit tag.

This time, when `Bob` logs in and attempts to create a new role, the audit log will record the unauthorized activity and it will also include the audit tag `Salesperson, Print Sales` (see Figure 4.8).



```
2016-04-27 04:07:31.543 PDT,"Bob","edb",40382,"::1:47632",57209cd8.9dbe,4,"idle"
,2016-04-27 04:04:56 PDT,4/107,0,AUDIT,00000,"statement: CREATE ROLE ""Ted"" VAL
ID UNTIL 'infinity';",,,,,,,,,"psql.bin","Salesperson, Print Sales"
2016-04-27 04:07:31.543 PDT,"Bob","edb",40382,"::1:47632",57209cd8.9dbe,5,"CREAT
E ROLE",2016-04-27 04:04:56 PDT,4/107,0,ERROR,42501,"permission denied to create
 role",,,,,,"CREATE ROLE ""Ted"" VALID UNTIL 'infinity';",,,"psql.bin","Salesper
son, Print Sales"
2016-04-27 04:07:34.338 PDT,"enterprisedb","edb",15159,"::1:47573",56d46207.3b37
```

Figure 4.8 – The audit log entry.

5 Appendix

5.1 Auditing Settings

This section describes configuration parameters used by the Advanced Server database auditing feature.

5.1.1 edb_audit

Parameter Type: Enum

Default Value: `none`

Range: {`none` | `csv` | `xml`}

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

Enables or disables database auditing. The values `xml` or `csv` will enable database auditing. These values represent the file format in which auditing information will be captured. `none` will disable database auditing and is also the default.

5.1.2 edb_audit_directory

Parameter Type: String

Default Value: `edb_audit`

Range: n/a

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

Specifies the directory where the audit log files will be created. The path of the directory can be absolute or relative to the *`POSTGRES_PLUS_HOME`*`/data` directory.

5.1.3 edb_audit_filename

Parameter Type: String

Default Value: `audit-%Y%m%d_%H%M%S`

Range: n/a

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

Specifies the file name of the audit file where the auditing information will be stored. The default file name will be `audit-%Y%m%d_%H%M%S`. The escape sequences, `%Y`, `%m` etc., will be replaced by the appropriate current values according to the system date and time.

5.1.4 edb_audit_rotation_day

Parameter Type: String

Default Value: `every`

Range: {`none` | `every` | `sun` | `mon` | `tue` | `wed` | `thu` | `fri` | `sat`} ...

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

Specifies the day of the week on which to rotate the audit files. Valid values are `sun`, `mon`, `tue`, `wed`, `thu`, `fri`, `sat`, `every`, and `none`. To disable rotation, set the value to `none`. To rotate the file every day, set the `edb_audit_rotation_day` value to `every`. To rotate the file on a specific day of the week, set the value to the desired day of the week.

5.1.5 edb_audit_rotation_size

Parameter Type: Integer

Default Value: 0MB

Range: 0MB to 5000MB

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

Specifies a file size threshold in megabytes when file rotation will be forced to occur. The default value is 0MB. If the parameter is commented out or set to 0, rotation of the file on a size basis will not occur.

5.1.6 edb_audit_rotation_seconds

Parameter Type: Integer

Default Value: 0s

Range: 0s to 2147483647s

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

Specifies the rotation time in seconds when a new log file should be created. To disable this feature, set this parameter to 0.

5.1.7 edb_audit_connect

Parameter Type: Enum

Default Value: `failed`

Range: {`none` | `failed` | `all`}

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

Enables auditing of database connection attempts by users. To disable auditing of all connection attempts, set `edb_audit_connect` to `none`. To audit all failed connection attempts, set the value to `failed`. To audit all connection attempts, set the value to `all`.

5.1.8 edb_audit_disconnect

Parameter Type: Enum

Default Value: `none`

Range: {`none` | `all`}

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

Enables auditing of database disconnections by connected users. To enable auditing of disconnections, set the value to `all`. To disable, set the value to `none`.

5.1.9 edb_audit_statement

Parameter Type: String

Default Value: `ddl, error`

Range: {`none` | `ddl` | `dml` | `select` | `error` | `rollback` | `all`}

Minimum Scope of Effect: Cluster

When Value Changes Take Effect: Reload

Required Authorization to Activate: PPAS service account

This configuration parameter is used to specify auditing of different categories of SQL statements. To audit statements resulting in error, set the parameter value to `error`. To audit DDL statements such as `CREATE TABLE`, `ALTER TABLE`, etc., set the parameter value to `ddl`. Modification statements such as `INSERT`, `UPDATE`, `DELETE` or `TRUNCATE` can be audited by setting `edb_audit_statement` to `dml`. Setting the value to `all` will audit every statement while `none` disables this feature.

5.1.10 edb_audit_tag

Parameter Type: String

Default Value: none

Minimum Scope of Effect: Session

When Value Changes Take Effect: Immediate

Required Authorization to Activate: User

Use `edb_audit_tag` to specify a string value that will be included in the audit log when the `edb_audit` parameter is set to `csv` or `xml`.