Appendix C

The following are the core instructions for enabling logging in PostgreSQL.

Note: The following instructions use the PGDATA and PGVER environment variables. See supplementary content APPENDIX-F and APPENDIX-H for instructions on configuring PGDATA and PGVER.

Where to Log

stderr

PostgreSQL can be configured to use `stderr` for logging. This allows the server to log events from the database to a directory specified in `postgresql.conf`.

Configuring stderr Logging

As the database administrator (usually `postgres`), edit the `postgresql.conf` file:

```
$ sudo su - postgres
$ vi ${PGDATA?}/postgresql.conf
```

The following parameters must be configured:

```
log_destination = 'stderr'
logging_collector = on
log_directory = 'pg_log'
log_filename = 'postgresql-%a.log'
log_file_mode = 0600
log_truncate_on_rotation = on
log_rotation_age = 1d
log_rotation_size = 0
```

As a sudo user, reload the PostgreSQL server:

```
# SERVER USING SYSTEMCTL ONLY
$ sudo systemctl reload postgresql-${PGVER?}

# SERVER USING INITD ONLY
$ sudo service postgresql-${PGVER?} reload
```

syslog

PostgreSQL can be configured to use `syslog` for logging. This allows the server to log events from the database to a centralized location and give log ownership to root instead of the database administrator (usually `postgres`). It is advised to use `syslog` whenever possible.

Configuring syslog Logging

Note: Syslog must be configured in your organization. The following instructions are only to configure PostgreSQL to use syslog.

As the database administrator (usually `postgres`), edit the `postgresql.conf` file:

```
$ sudo su - postgres
$ vi ${PGDATA?}/postgresql.conf
```

The following parameters must be configured:

```
log_destination = 'syslog'
syslog_facility = 'LOCAL0' # choose the facility that makes sense
syslog_ident = 'postgres'
```

As a sudo user, reload the PostgreSQL server:

```
# SERVER USING SYSTEMCTL ONLY
$ sudo systemctl reload postgresql-${PGVER?}

# SERVER USING INITD ONLY
$ sudo service postgresql-${PGVER?} reload
```

What to Log

PostgreSQL can log a variety of events out of the box. The following parameters can be set to log additional information.

As the database administrator (usually `postgres`), edit the `postgresql.conf` file:

```
$ sudo su - postgres
$ vi ${PGDATA?}/postgresql.conf
```

The following parameters can be configured:

```
log_checkpoints = on
log_connections = on
log_disconnections = on
log_duration = off
log_error_verbosity = default
log_hostname = off
log_lock_waits = on
log_statement = 'none' # pgaudit will be configured to log specific
events
log_timezone = 'UTC'
```

Additionally, `log_line_prefix` can be configured to include extra information:

```
log_line_prefix = '< %m %a %u %d %c %s %r >'
#   %a = application name
#   %u = user name
#   %d = database name
#   %r = remote host and port
#   %h = remote host
#   %p = process ID
#   %t = timestamp without milliseconds
#   %m = timestamp with milliseconds
#   %i = command tag
#   %e = SQL state
#   %c = session ID
#   %l = session line number
#   %s = session start timestamp
#   %v = virtual transaction ID
```

```
#   %x = transaction ID (0 if none)
#   %q = stop here in non-session
#        processes
```

When to Log

PostgreSQL allows administrators to control what levels of detail are exposed to logs, administrators, and clients.

As the database administrator (usually `postgres`), edit the `postgresql.conf` file:

```
$ sudo su - postgres
$ vi ${PGDATA?}/postgresql.conf
```

The following parameters can be configured:

```
client_min_messages = error
log_min_messages = warning
log_min_error_statement = error
log_min_duration_statement = -1
```

For more information on logging, see the official documentation: http://www.postgresql.org/docs/current/static/runtime-config-logging.html.