

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [pgAgent](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Installing pgAgent](#)
 - [Database setup](#)
 - [Daemon installation on Unix](#)
 - [Service installation on Windows](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
 - [Using pgAgent](#)
 - [Installing pgAgent](#)
 - [Creating a pgAgent Job](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Installing pgAgent ¶

pgAgent runs as a daemon on Unix systems, and a service on Windows systems. In most cases it will run on the database server itself - for this reason, pgAgent is not automatically configured when pgAdmin is installed. In some cases however, it may be preferable to run pgAgent on multiple systems, against the same database; individual jobs may be targeted at a particular host, or left for execution by any host. Locking prevents execution of the same instance of a job by multiple hosts.

Database setup ¶

Before using pgAdmin to manage pgAgent, you must create the pgAgent extension in the maintenance database registered with pgAdmin. To install pgAgent on a PostgreSQL host, connect to the *postgres* database, and navigate through the *Tools* menu to open the Query tool. For server versions 9.1 or later, and pgAgent 3.4.0 or later, enter the following command in the query window, and click the *Execute* icon:

```
CREATE EXTENSION pgagent;
```

This command will create a number of tables and other objects in a schema called 'pgagent'.

The database must also have the pl/pgsql procedural language installed - use the PostgreSQL CREATE LANGUAGE command to install pl/pgsql if necessary. To install pl/pgsql, enter the following command in the query window, and click the *Execute* icon:

```
CREATE LANGUAGE plpgsql;
```

If you are using an earlier version of PostgreSQL or pgAgent, use the *Open file* icon on the Query Tool toolbar to open a browser window and locate the *pgagent.sql* script. The installation script is installed by pgAdmin, and the installation location varies from operating system to operating system:

- On Windows, it is usually located under *C:\Program files\pgAdmin III* (or *C:\Program files\PostgreSQL\8.x\pgAdmin III* if installed with the PostgreSQL server installer).
- On Linux, it is usually located under */usr/local/pgadmin3/share/pgadmin3* or */usr/share/pgadmin3*.

After loading the file into the Query Tool, click the *Execute* icon to execute the script. The script will create a number of tables and other objects in a schema named *pgagent*.

Daemon installation on Unix ¶

Note

pgAgent is available in Debian/Ubuntu (DEB) and Redhat/Fedora (RPM) packages for Linux users, as well as source code. See the [pgAdmin Website](#). for more information.

To install the pgAgent daemon on a Unix system, you will normally need to have root privileges to modify the system startup scripts. Modifying system startup scripts is quite system-specific so you should consult your system documentation for further information.

The program itself takes few command line options, most of which are only needed for debugging or specialised configurations:

```
Usage:
/path/to/pgagent [options] <connect-string>

options:
  -f run in the foreground (do not detach from the
terminal)
  -t <poll time interval in seconds (default 10)>
  -r <retry period after connection abort in seconds
(>=10, default 30)>
  -s <log file (messages are logged to STDOUT if not
specified)>
  -l <logging verbosity (ERROR=0, WARNING=1, DEBUG=2,
default 0)>
```

The connection string is a standard PostgreSQL libpq connection string (see the [PostgreSQL documentation on the connection string](#) for further details). For example, the following command line will run pgAgent against a server listening on the localhost, using a database called 'pgadmin', connecting as the user 'postgres':

```
/path/to/pgagent hostaddr=127.0.0.1 dbname=postgres
user=postgres
```

Service installation on Windows ¶

Note

pgAgent is available in a pre-built installer if you use [EnterpriseDB's PostgreSQL Installers](#). Use the StackBuilder application to download and install it. If installed in this way, the service will automatically be created and the instructions below can be ignored.

pgAgent can install itself as a service on Windows systems. The command line options available are similar to those on Unix systems, but include an additional parameter to tell the service what to do:

```
Usage:
  pgAgent REMOVE <serviceName>
  pgAgent INSTALL <serviceName> [options] <connect-
string>
  pgAgent DEBUG [options] <connect-string>

options:
  -u <user or DOMAIN\user>
  -p <password>
  -d <displayname>
  -t <poll time interval in seconds (default 10)>
  -r <retry period after connection abort in seconds
(>=10, default 30)>
  -l <logging verbosity (ERROR=0, WARNING=1, DEBUG=2,
default 0)>
```

The service may be quite simply installed from the command line as follows (adjust the path as required):

```
"C:\Program Files\pgAgent\bin\pgAgent" INSTALL pgAgent -u
postgres -p secret hostaddr=127.0.0.1 dbname=postgres
user=postgres
```

You can then start the service at the command line using *net start pgAgent*, or from the Services control panel applet. Any logging output or errors will be reported in the Application event log. The DEBUG mode may be used to run pgAgent from a command prompt. When run this way, log messages will output to the command window.