

※ How to install AgensGraph in Linux environment

- ① Set the environment variables (optional).

Assuming you have downloaded and unzipped the package (AgensGraph_v2.1.0_linux_CE.tar.gz) to `/usr/local/AgensGraph/`, add the following three lines to your shell startup file (e.g. `.bash_profile`).

```
export LD_LIBRARY_PATH=/usr/local/AgensGraph/lib:$LD_LIBRARY_PATH
export PATH=/usr/local/AgensGraph/bin:$PATH
export AGDATA=/path/to/make/db_cluster
```

```
[agens@:/usr/local/AgensGraph]#cat .bash_profile
export LD_LIBRARY_PATH=/usr/local/AgensGraph/lib:$LD_LIBRARY_PATH
export PATH=/usr/local/AgensGraph/bin:$PATH
export AGDATA=/usr/local/AgensGraph/db_cluster
```

- ② Create a database cluster.

Create a database cluster by entering the following command:

```
initdb [-D /path/to/make/db_cluster]
```

```
[agens@:/usr/local/AgensGraph]#initdb -D /usr/local/AgensGraph/db_cluster
The files belonging to this database system will be owned by user "agens".
This user must also own the server process.

The database cluster will be initialized with locale "ko_KR.UTF-8".
The default database encoding has accordingly been set to "UTF8".
initdb: could not find suitable text search configuration for locale "ko_KR.UTF-8"
The default text search configuration will be set to "simple".

Data page checksums are disabled.

creating directory /usr/local/AgensGraph/db_cluster ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    ag_ctl -D /usr/local/AgensGraph/db_cluster -l logfile start
```

- ③ Start the server.

Start AgensGraph by entering the following command:

```
ag_ctl start [-D /path/created/by/initdb]
```

```
[agens@:/usr/local/AgensGraph]#ag_ctl -D /usr/local/AgensGraph/db_cluster -l logfile start
server starting
```

- ④ Create a database and execute the terminal.

Create a database on the AgensGraph and run an AgensGraph terminal to access the database using the following commands:

```
createdb [dbname]
agens [dbname]
```

```
username=#
```

```
[agens@:bitnine01]$ createdb
[agens@:bitnine01]$ agens
agens (AgensGraph 2.1.0, based on PostgreSQL 10.4)
Type "help" for help.
agens=#
```

If it is the superuser, "= #" will be displayed in the prompt; other users will be displayed as "=>".

```
[agens@:bitnine01]$ agens
agens (AgensGraph 2.1.0, based on PostgreSQL 10.4)
Type "help" for help.
agens=# CREATE GRAPH test_graph1;
CREATE GRAPH
agens=# SET graph_path = test_graph1;
SET
agens=# CREATE (testNode:Test {name: 'TestNode'});
GRAPH WRITE (INSERT VERTEX 1, INSERT EDGE 0)
agens=# MATCH (testNode:Test {name: 'TestNode'})
agens=# RETURN testNode;
          testnode
-----
 test[3.1]{"name": "TestNode"}
(1 row)
```

✂ How to install AgensGraph in Windows environment

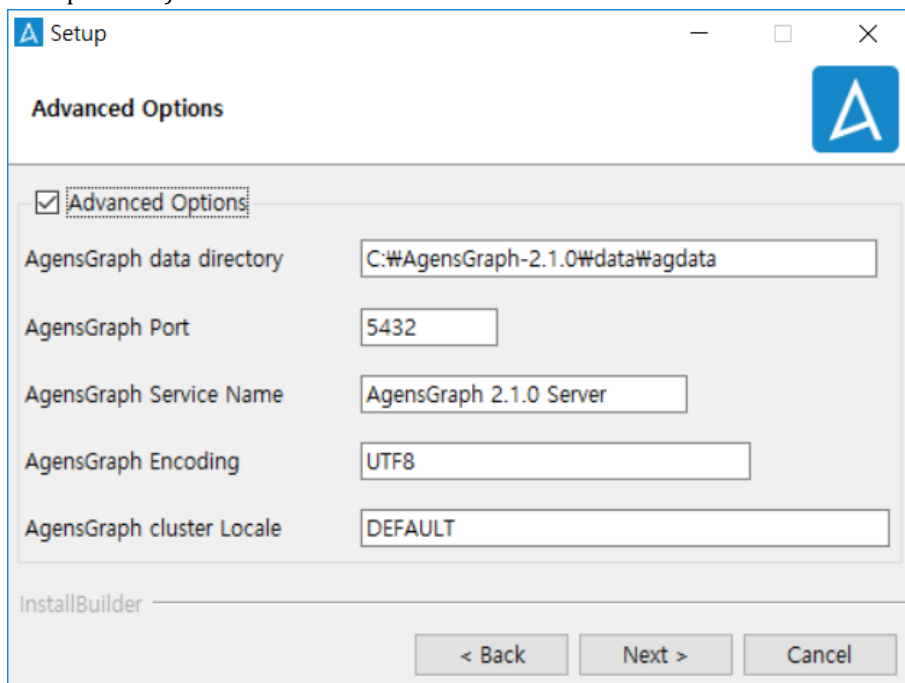
- ① Run the installation file as an administrator.

Run the installation file you have downloaded (e.g. AgensGraph-2.1.0-windows-installer) as an administrator.

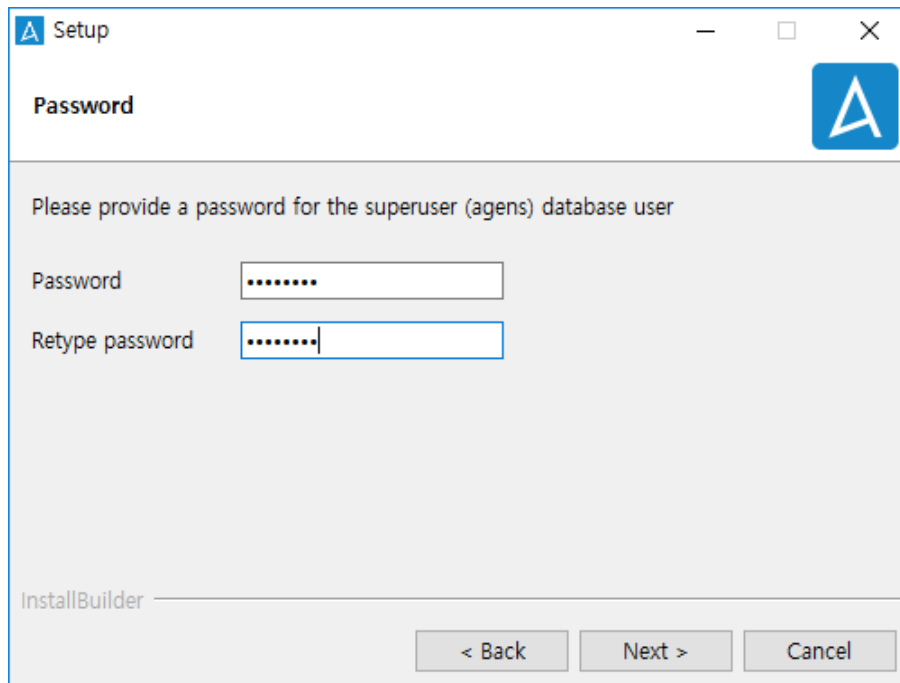
- ② Accept the license terms and specify the installation path.



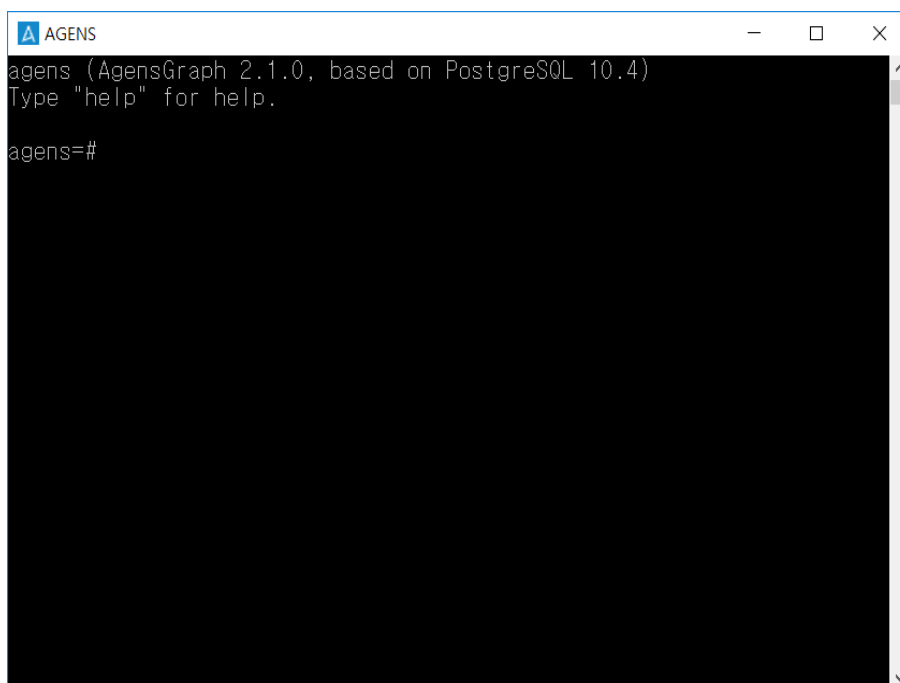
- ③ Configure the installation settings including the installation path, encoding to use, and port to be assigned to AgensGraph (If you do not want to make changes, clear the Advanced Options checkbox and proceed.)



- ④ Set the password of the superuser account (default superuser: agens) and click Next. A screen to confirm the installation options will appear; click Next to proceed with the installation.



⑤ After your installation is complete, click on the AGENS icon (A) registered in the Windows Start Menu to start AgensGraph as shown below:



⑥ In order to check if AgensGraph runs normally, we created a sample graph, set the graph path, and then performed a query that creates and returns a single vertex. As you can see below, all the processes are executed as intended:

```
선택 AGENS
agens (AgensGraph 2.1.0, based on PostgreSQL 10.4)
Type "help" for help.

agens=# CREATE GRAPH test_graph1;
CREATE GRAPH
agens=# SET graph_path = test_graph1;
SET
agens=# CREATE (testNode:Test {name: 'TestNode'});
GRAPH WRITE (INSERT VERTEX 1, INSERT EDGE 0)
agens=# MATCH (testNode:Test {name: 'TestNode'})
agens-# RETURN testNode;
          testnode
-----
 test[3.1>{"name": "TestNode"}
(1 row)

agens=#
```

For more details on each installation (Linux/Windows), click the link(s) above or see [AgensGraph Developer Manual](#).