

PostgreSQL

The db.config file contains configuration parameters that control the startup and shutdown behavior of the backend PostgreSQL database, as well as the client connection parameters.

DBStartup

DBStartup must be set to True for the engine to start a postgresql server if one is not already running when the engine is started.

```
"DBStartup": True,
```

DBUseExisting

DBUseExisting must be set to True for the engine to use an already-running database server process when the engine is started. Some sites may wish to leave the job database running even when the engine is stopped; others may set this value to False as an assertion that a new engine invocation should not write to a database server that it did not start itself.

```
"DBUseExisting": True,
```

DBAutoUpgrade

NOTE: This feature is not available in Beta 0.

DBAutoUpgrade must be set to False to suppress the automatic upgrading of the database schema when a new version of Tractor (with a new schema version) is first starting. If set to False and a schema upgrade is required, the engine will not run. The site admin will need to run tractor-dbctl --upgrade manually to perform the upgrade; otherwise, the older (schema-compatible) version of Tractor must be run. If the admin does not wish to save existing job data, the entire database can be removed with tractor-dbctl --destroy, and the an empty database under the new schema will automatically be created when the database is started by the engine or manually. This variable is also observed by tractor-dbctl --start.

```
"DBAutoUpgrade": True,
```

DBInit

DBInit must be set to true for a new database to be created if they are not found at engine startup time. A site may wish to set this to False in order to avoid creating an entirely new database should there happen to be a problem mounting the filesystem holding the existing database.

```
"DBInit": True,
```

DBShutdown

DBShutdown must be set to true for the postgresql server to be shutdown when the engine stops. A site may wish to set this to False in order to allow postgresql clients to run queries against the database should the engine has been shutdown or has crashed. Note that URL-based queries are proxied through the engine and cannot run when the engine is down.

```
"DBShutdown": True,
```

DBArchiving

DBArchiving, when True, will cause deleted jobs and their associated tasks, commands, and invocations to be moved to special partitions in the postgresql database. To see them, users will need to use the --archives flag with tq, the archives keyword parameter in the Python query API, or select the archives button in the Dashboard query pane.

```
"DBArchiving": True,
```

DBDebug

DBDebug, when True, will cause debugging output to be emitted when creating, starting, syncing, and stopping the postgresql server.

```
"DBDebug": False,
```

DBDataDir

DBDataDir indicates the directory that will store the database tables. A full path to a different location may be specified.

```
"DBDataDir": "${TractorDataDirectory}/psql",
```

DBConnection

DBConnection are the client side postgres connection parameters. Values can be edited or added to this dictionary as required. # 'host' specifies the hostname where tractor-engine should find its PostgreSQL server. Use "localhost" for the usual case where the job database is on the same host as tractor-engine and is managed by it. # 'port' indicates the port that the postgresql server will listen on. By default, it is set to the standard PostgreSQL "postmaster" port, but could be changed in case there is some other service already using that port, such as another PostgreSQL server, or the unusual case of testing two tractor-engines on the same host (not recommended). Note that a corresponding change should be made in postgres.conf. # 'dbname' specifies the database name where the tables are stored, it should always be "tractor" except in unusual circumstances. #

```
{
  "DBConnection": {
    "host": "localhost",
    "port": 9876,
    "dbname": "tractor"
  }
}
```

DBRolePasswords

Sites may restrict external access to the Tractor job database in several ways. One is to specify "localhost" above to prevent network access. Another is to add passwords for the various database login roles. Tractor-engine will need to know these passwords, so they must be enumerated here. Obviously read-access to this configuration file must then be restricted to administrators (and the Tractor "EngineOwner") when sensitive passwords are specified here. #

```
"DBRolePasswords": {
  "root": "",
  "bootstrap": "",
  "dispatcher": "",
  "spooler": "",
  "dashboard": "",
  "reader": "",
  "dev_read": "",
  "dev_write": ""
}
```

Simple Configuration Example

The following settings are ideal for an out-of-box experience where the site wishes to leave management of postgresql to the engine with little or not intervention from the admin. Here the postgres server runs on the same host as the engine, and only when the engine is running.

```
{
  "DBStartup": True,
  "DBUseExisting": True,
  "DBAutoUpgrade": True,
  "DBInit": True,
  "DBShutdown": True,
  "DBDebug": False,

  "DBDataDir": "${TractorDataDirectory}/psql",
  "DBConnection": {
    "host": "localhost",
    "port": 9876,
    "dbname": "tractor"
  }
}
```

Complex Configuration Example

The following settings are ideal for site that which to manage a custom postgresql server and wants to leave the postgresql server running while the engine is down for reasons such as: (a) to service queries that are not proxied through the engine, (b) the postgresql server runs on a different host and requires a different manner of startup/shutdown, (c) other database services are being managed on a central postgresql server (d) other reasons to simply manually manage the postgresql server. In this example, schema upgrades are handled manually so that should a new tractor version be installed and there is a schema change, there will not be unexpected downtime from waiting for schema upgrade operations to complete; note that the admin will need to observe that the newer version of the engine failed to start and will need to start the older version manually.

```
{
  "DBStartup": False,
  "DBUseExisting": True,
  "DBAutoUpgrade": False,
  "DBInit": False, # switch to True later
  "DBShutdown": False,
  "DBDebug": False,

  "DBDataDir": "/path/to/fast/filesystem/psql",
  "DBConnection": {
    "host": "otherhost",
    "port": 9876,
    "dbname": "tractor"
  }
}
```