

# D2000 System databases

In D2000 System, since version V10.0.38 PostgreSQL enables to operate these databases:

- configuration
- log
- archive
- depository
- application

The log depository databases are not supported for PostgreSQL. The new databases, except the application ones (see the section [Application databases in D2000 System](#)), are created in D2000 Management Console when creating the application or archive. This program also enables to create and manage the database services.

For the common users, the creation and management of databases on PostgreSQL is hide and the only requirement is to have this system installed. In certain situations (creating the application or archive), user must enter the password of main user.

When creating the application with name *Application* over PostgreSQL, the user must select the existing database service, which ensures running of database, and password to its main user with name *postgres*. Then, D2MC will create the databases *aplikacia\_syscfg* and *aplikacia\_logfile* in this database service and save them into separate tablespaces (*ts\_aplikacia\_syscfg*, *ts\_aplikacia\_logfile*), placed in subdirectories *syscfg* and *logfile* in application directory of D2000 System. It also creates DSN records *Aplikacia.SysCfg* and *Aplikacia.LogFile* for these databases. The created user *dba* is the owner of these databases and has the access to them.

When creating the archive for *Application* over PostgreSQL, the user must select the existing database service, which ensures running of database, and password to its main user with name *postgres*. Then, D2MC will create the database *aplikacia\_archiv* in this database service and save them into separate tablespaces (*ts\_aplikacia\_archiv*), placed in subdirectory *archiv* in application directory of D2000 System. It also creates DSN record *Aplikacia.Archiv* for this database. The created user *dba* is the owner of this database and has the access to it.

Each [depository](#), or depository segment is implemented as the separate database. The names are derived from the application name (*<application>\_TS\_#ID#*, wher #ID# is depository number, or *<application>\_TS\_#ID#\_#SEG#*, where #SEG# is the number of depository segment).

## Backup of D2000 System databases

As backup of PostgreSQL databases can be done in various ways, D2000 System does not contain any tool or functionality, which would ensure a backup of these databases. Backup should, therefore, be done manually according to one of the methods described in this manual: <http://www.postgresql.org/docs/9.4/static/backup.html>.

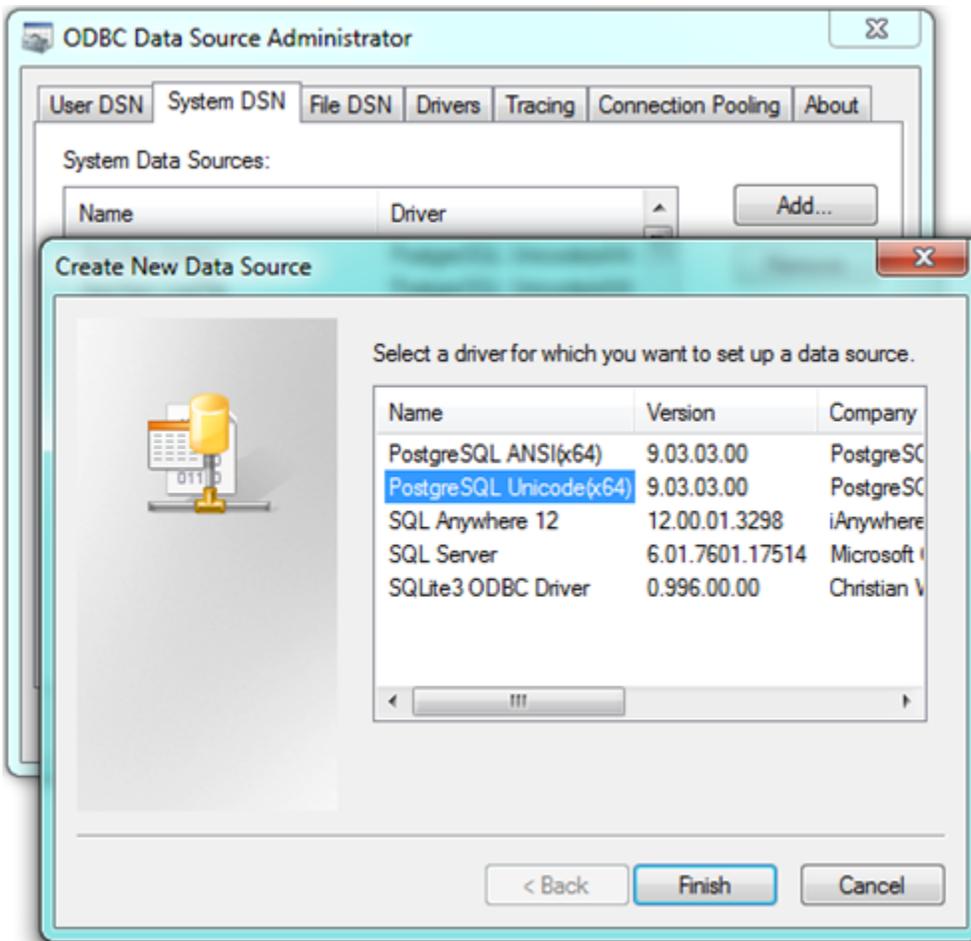
List of databases, which is necessary to backup this way, can be derived from the name of application as follows:

- *<application name>\_syscfg*
- *<application name>\_logfile*
- *<application name>\_archiv*
- *<application name>\_TS\_<ID>*

## Application databases in D2000 System

---

In the event that D2000 System should work with the external application database, for its integration user must create a system DSN record in ODBC Administrator on the computer with running D2000 DBManager.

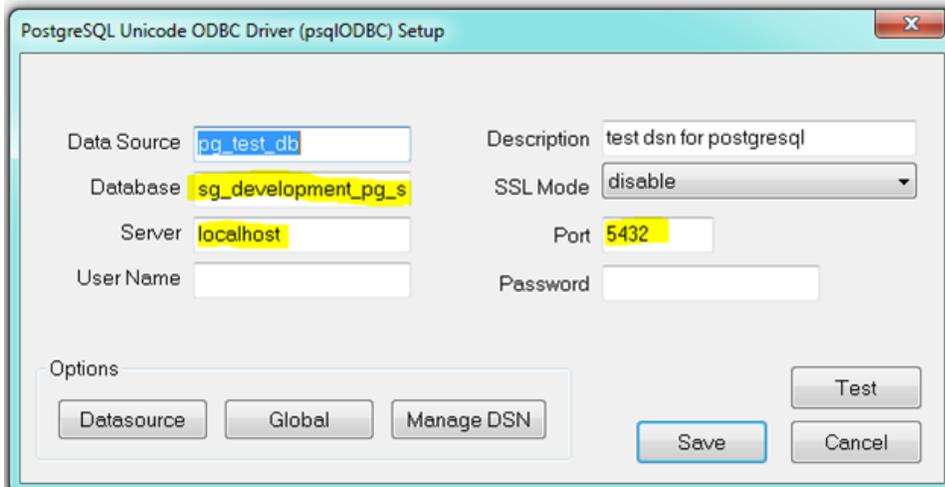


**Warning:** Both 32 and 64 bit ODBC Administrators exist in 64-bit OS Windows. DSN record must be created in the 32/64 ODBC Administrator depending on version of D2000 DBManager (as 32-bit applications cannot use 64-bit DSN and vice versa).

In case of selecting more types of DSN records, user should select the one which contain *Unicode* in the name.

The configuration of DSN record for PostgreSQL should be done according to these steps:

1. Write, correctly, these parameters in the main window:



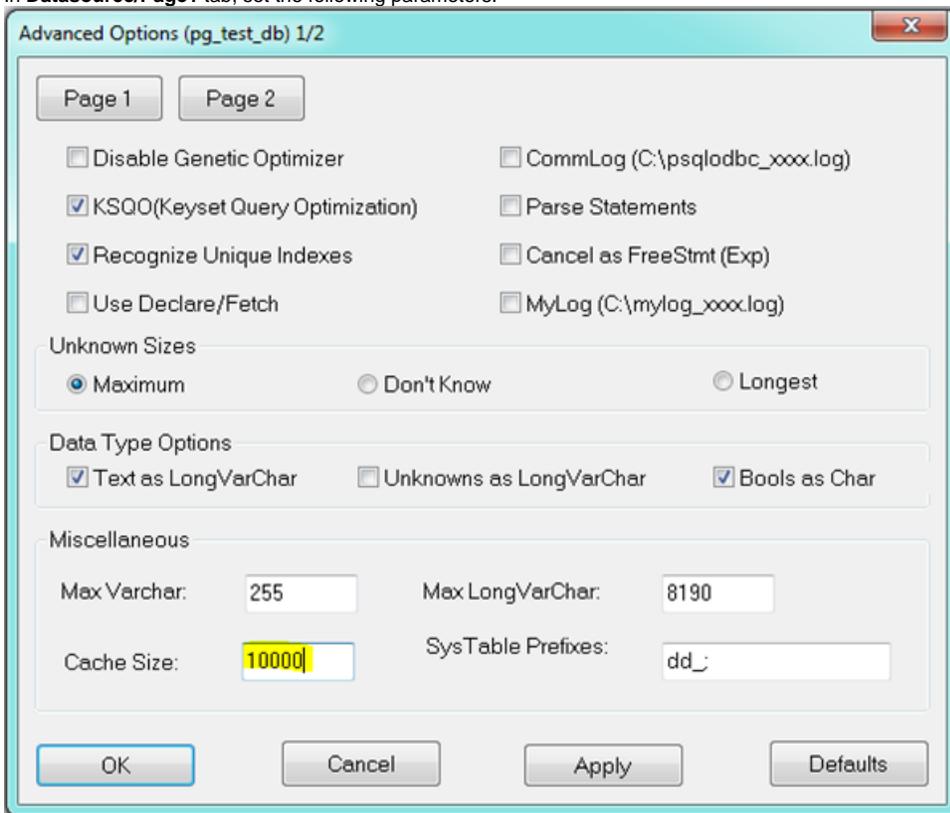
**Database** – name of existing database.

**Server** – address of server, which PostgreSQL service runs on (or localhost in case of the same computer),  
**Port** – port number, which PostgreSQL service listens on.

After entering the name and password and clicking the **Test**, D2000 should connect to database successfully. If the connection failed, you must check the entered data and settings of PostgreSQL service, which database runs on (see the section [Connection to PostgreSQL database running on dedicated machine](#)). Warning! For security reasons, do not keep the name and password filled in the final state!

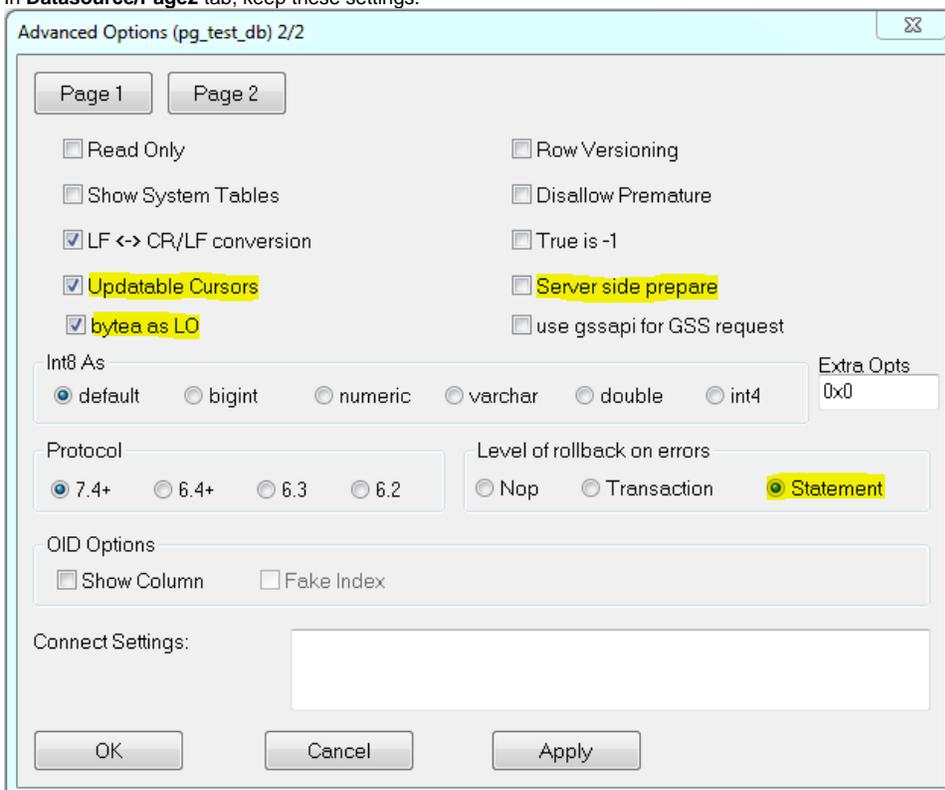
For proper functioning with D2000 System, it is necessary to change other default parameters.

2. In **Datasource/Page1** tab, set the following parameters:



**CacheSize** can be set at your discretion.

3. In **Datasource/Page2** tab, keep these settings:



The highlighted parameters must be set as follows:

**Updatable Cursors** – the ability to use cursors to writes (enables updateable cursor emulation in the driver)

**bytea as LO** - enables support for writing Large Objects (blobs) into bytea columns

**Server side prepare** - compiled cursors (e.g. for parameterized queries) on the server side can improve performance. If possible, we recommend that you enable this option.

**Level of rollback on errors** - if transaction operations are used and there may be an error during command execution (e.g. INSERT fails because of a unique index violation), then if the parameter is set to **Statement**, only the last command is rolled back. If the parameter is set to **Transaction**, the entire transaction is rolled back. On the other hand, the **Statement** setting makes "microcommits" (savepoints) after each operation, which both reduces performance and consumes transactional IDs, which in turn leads to increased vacuuming activity of PostgreSQL database.

This DSN record can be used for [connection](#) to external PostgreSQL database in D2000 System.

## Connection to PostgreSQL database running on dedicated machine

In the event that D2000 System should connect to the remote server, it is necessary to configure DSN record in ODBC administrator on client according to the [previous instructions](#) with the only difference - into *Server* parameter enter the name or IP address of server, which PostgreSQL database runs on.

On dedicated machine with the running database, which you want to connect to, you must properly define the rights for connection. Default settings of PostgreSQL enables connection to the running database service only from the computer on which it is started. When necessary to connect from the network, you must change this setting by the configuration file *pg\_hba.conf*, which is placed in the selected directory of database service.

It contains this default settings:

```

# TYPE  DATABASE      USER                ADDRESS              METHOD
# IPv4 local connections:
host    all            all                 127.0.0.1/32        md5
# IPv6 local connections:
host    all            all                 ::1/128             md5

```

This setting enables connection to database server only for the user from the same computer (127.0.0.1 represents the localhost).

If the user wants to connect from any computer, the configuration file must contain this line:

```

# TYPE  DATABASE      USER                ADDRESS              METHOD
# IPv4 local connections:
host    all            all                 127.0.0.1/32        md5
host    all            all                 0.0.0.0/0           md5
# IPv6 local connections:
host    all            all                 ::1/128             md5

```

This line enables to connect all users to all databases of running service from all addresses. However, the password is required (md5 method).

The setting, mentioned above, has the illustrative character and it is not recommended for use on the production systems because it can represent a security risk when using the weak passwords. The configuration file `pg_hba.conf` enables to limit the assigning the rights for connection to database (based on the user or database), which can be found for example on this link <http://www.postgresql.org/docs/9.4/static/auth-pg-hba-conf.html>.

Each change of this configuration file requires reloading of configuration ([Reloading of configuration database service](#)), or restarting the database service.

## Specialities of PostgreSQL

---

### Using the quotation marks for name of tables and columns

The identifiers of tables and columns in PostgreSQL, which are not in quotation marks, are not case sensitive. It means, the column can be created in this way:

```
create table t (id integer)
```

Then, these queries can be created:

```
select id from t
```

```
select Id from t
```

```
select ID from t
```

```
select "id" from t
```

All forms of querying of the given column are functional.

The identifiers of tables and columns can be written in quotation marks, too. However, within queries the identifier must have the same size of characters as when it was created, and always it must be in quotation marks.

It means, when creating this column:

```
create table t ("Id" integer)
```

it is possible to refer to it only in the same format:

```
select "Id" from t
```

Any other format of this record will not work.

In regard of complications, which can occur in case of using the identifiers with quotation marks, we do not recommend to use this format of entry. In some situations, this format is necessary, e.g. if name of column is the same as some key word in PostgreSQL.

When using the application database in D2000 System, we do not recommend to use the identifiers in quotation marks. D2000 System supports both formats of entries, but when using quotation marks, it must be used even in selection and sorting conditions. This can relate to both the using of database operations in ESL scripts (with WHERE conditions), and displaying database table in D2000 HI (if some selection or sorting condition is entered).



**Related pages:**

[Support of PostgreSQL in D2000 System](#)