PostgreSQL connection strings

dotConnect for PostgreSQL (former Core Labs PostgreSQLDirect)

Standard

```
User ID=root; Password=myPassword; Host=localhost; Port=5432; Database=myDataBase; Pooling
```

PostgreSQL OLE DB Provider

Standard

PgOleDb requires a PQLib of version 7.4 or up and it also requires a backend of version 7.4 or up. Timestamps are only guarenteed to work with backends of version 8.0 and up.

```
Provider=PostgreSQL OLE DB
Provider; Data Source=myServerAddress; location=myDataBase; User ID=myUsername; password=
```

Other valid Provider values is "PostgreSQL" and "PostgreSQL.1"

Standard alternative

Some reported problems with the above one. Try removing the timeout parameter to make it work.

```
Provider=PostgreSQL OLE DB
Provider; Data Source=myServerAddress; location=myDataBase; User ID=myUsername; password=
```

.NET Framework Data Provider for OLE DB

Use an OLE DB provider from .NET

```
Provider=any oledb provider's name; OledbKey1=someValue; OledbKey2=someValue;
```

See the respective OLEDB provider's connection strings options. The .net OleDbConnection will just pass on the connection string to the specified OLEDB provider. Read more here.

PostgreSQL ODBC Driver (psqlODBC)

Standard

```
Driver={PostgreSQL}; Server=IP
address; Port=5432; Database=myDataBase; Uid=myUsername; Pwd=myPassword;
```

ANSI

```
Driver={PostgreSQL ANSI}; Server=IP
address; Port=5432; Database=myDataBase; Uid=myUsername; Pwd=myPassword;
```

Unicode

```
Driver={PostgreSQL UNICODE}; Server=IP
address; Port=5432; Database=myDataBase; Uid=myUsername; Pwd=myPassword;
```

SSL

Secure sockets layer for this driver only works from version 8.0 and above.

```
Driver={PostgreSQL ANSI}; Server=IP
address; Port=5432; Database=myDataBase; Uid=myUsername; Pwd=myPassword; sslmode=require
```

Please note that sslmode=require is case sensitive, it should be written in lower case letters.

Npgsql

Standard

```
Server=127.0.0.1; Port=5432; Database=myDataBase; User Id=myUsername; Password=myPasswor
```

Using windows security

```
Server=127.0.0.1; Port=5432; Database=myDataBase; Integrated Security=true;
```

Setting command timeout

```
Server=127.0.0.1; Port=5432; Database=myDataBase; User Id=myUsername; Password=myPasswor
```

The CommandTimeout parameter is measured in seconds and controls for how long to wait for a command to finish before throwing an error.

Setting connection timeout

```
Server=127.0.0.1; Port=5432; Database=myDataBase; User Id=myUsername; Password=myPasswor
```

The Timeout parameter is measured in seconds and controls for how long to wait for a connection to open before throwing an error.

Specifying protocol version

```
Server=127.0.0.1;Port=5432;Database=myDataBase;User Id=myUsername;Password=myPasswor
```

Valid values for the key Protocol is 2 or 3.

SSL activated

```
Server=127.0.0.1;Port=5432;Database=myDataBase;Userid=myUsername;Password=myPassword
```

Without SSL

```
Server=127.0.0.1;Port=5432;Database=myDataBase;Userid=myUsername;Password=myPassword
```

Controlling pooling mechanisms

```
Server=127.0.0.1;Port=5432;Database=myDataBase;Userid=myUsername;Password=myPassword
```

.NET Framework Data Provider for ODBC

Use an ODBC driver from .NET

```
Driver={any odbc driver's name};OdbcKey1=someValue;OdbcKey2=someValue;
```

See the respective ODBC driver's connection strings options. The .net OdbcConnection will just pass on the connection string to the specified ODBC driver. Read more here.