TKa 09/2015

EasyConnect - Database Connection

Important note: This application note is only valid beginning with Easy on-PC software version 2.0.1.04 and EasyOne Pro / LAB software version 2.0.1.05.

This application note describes how to use and manage SQL server databases in general.

The tested versions include Microsoft SQL Server 2008 and Microsoft Server 2014.

There is not a high amount of storage required. If you use 30 devices on a daily basis over 5 years, they would create around 16GB of data.

This information applies to EasyOne Pro as well as Easy on-PC; i.e. to all products that use the software EasyWare Pro.

Content

1 Database possibilities

With the software update of Easy on-PC and EasyOne Pro / LAB from V1.9 to V2.X, the database type changed from Access to SQL / SQLite. Now you can choose between a file based database (SQLite) which is stored locally and a Microsoft SQL server based database which is a network database. The option to store the file based database on your network is still possible as with previous versions.

The new Microsoft SQL database is encrypted and password protected. It is possible to create an encrypted SSL connection, but this is dependent upon your server. In order to set up an SSL connection go to point 3.

The Microsoft SQL server solution is recommended when there will be a large number of users (›5) and / or large databases (›1 GB).

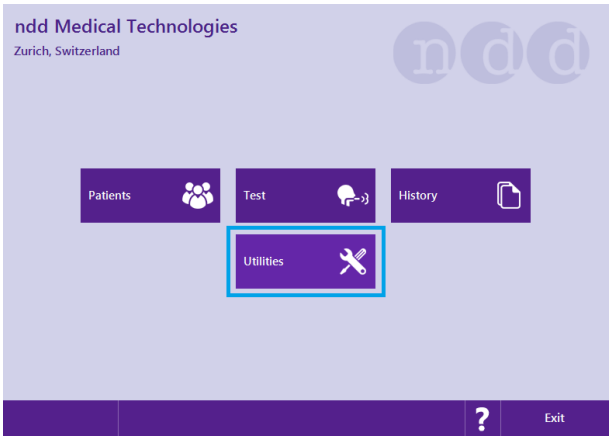
1.1 Set up a file based database

In order to create, select and update a database follow the instructions below.

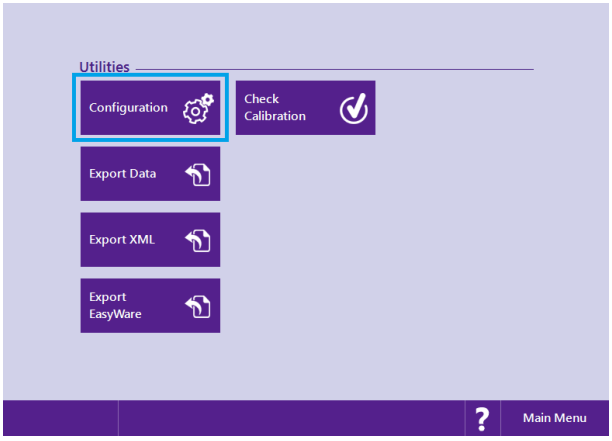
File paths may differ from OS to OS or from an Easy on-PC system to an EasyOne Pro system.

1.1.1 Create a new file based database

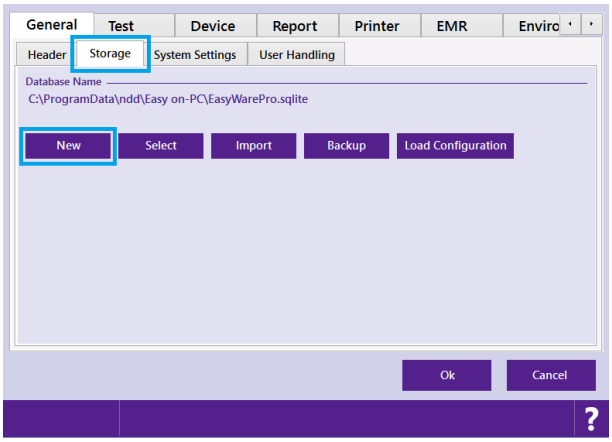
Go to Utilities



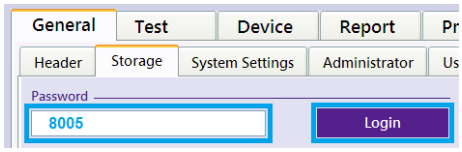
Configuration



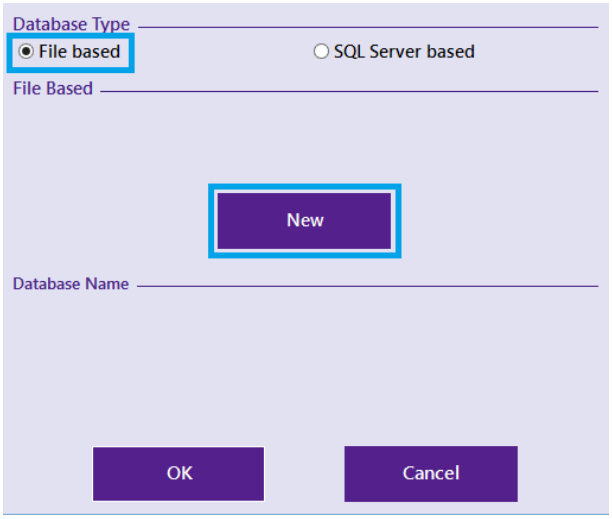
Tab: Storage and click on New



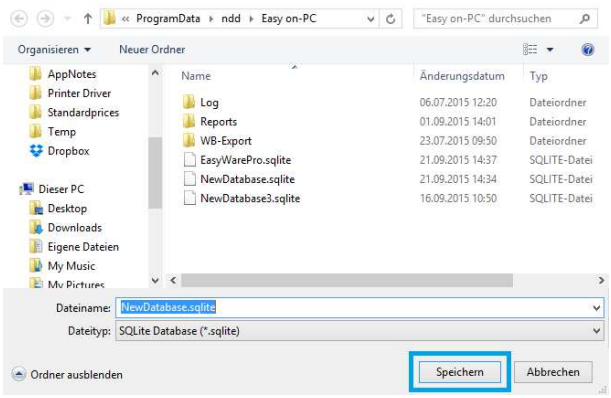
For EasyOne Pro / LAB users type in 8005 to log in.



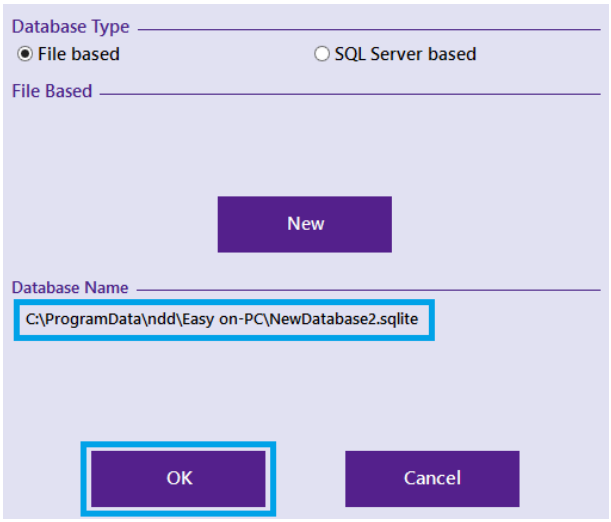
Make sure you check the file based option and click on New.



Select the folder where the database is stored, rename the database if you want and click Save.

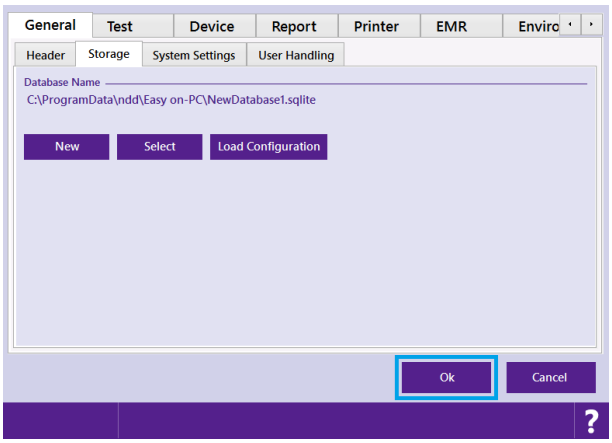


Double check the path and click OK.



Click OK to apply the changes.

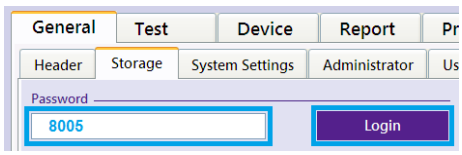
The software will restart automatically and load the new database. You can now start to use the new database.

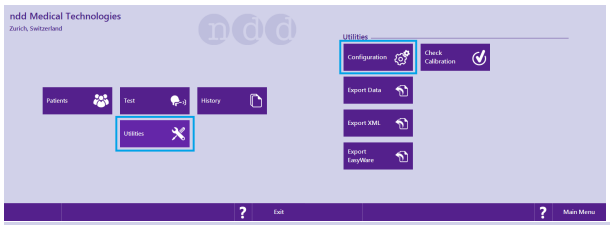


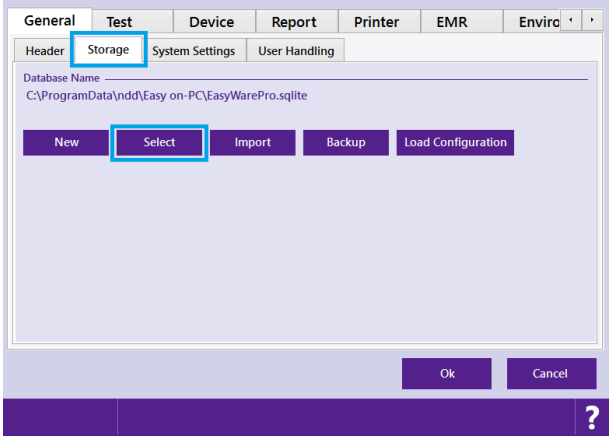
1.1.2 Select a file based database

Go to Utilities / Configuration / tab: Storage and click Select

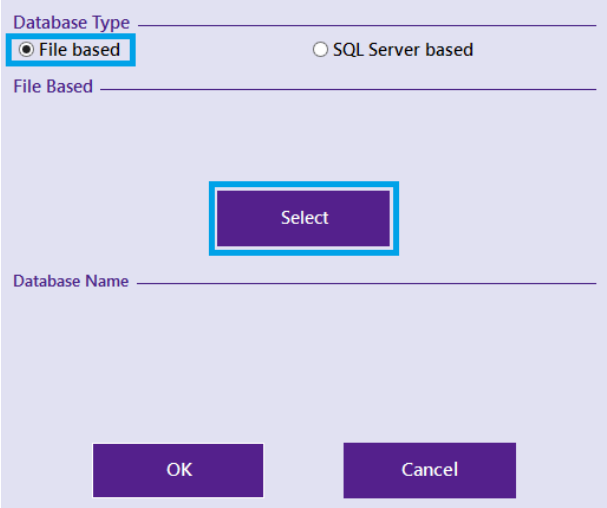
For EasyOne Pro / LAB users type in 8005 to log in.



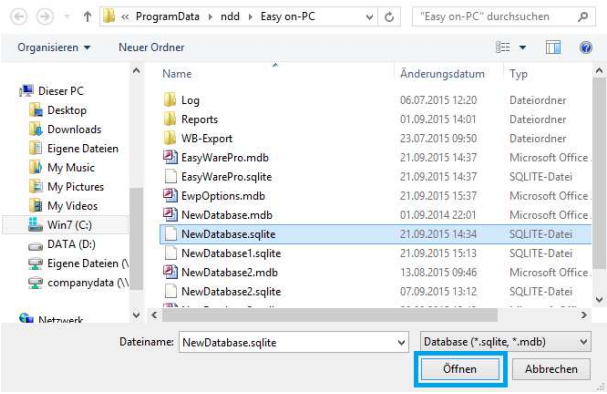




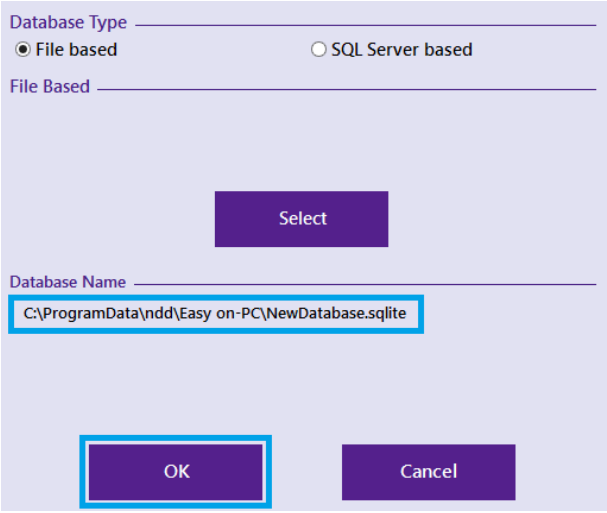
Make sure you check the file based option and click Select.



Select the database and click Open.

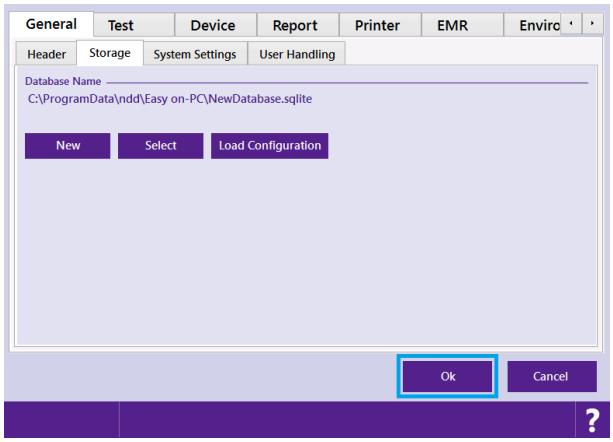


Double check the path and click OK.



Click OK to apply the changes.

The software will restart automatically and load the new database. You can now start to use your selected database.



1.1.3 Update an old file based database

If you already have been using an Easy on-PC or an EasyOne Pro / LAB with versions older than V2.X and you created more than one database, simply use the select option and the software will automatically ask you if you want to update your database while the software performs the automatic restart.

1.2 Set up a Microsoft SQL Server based database

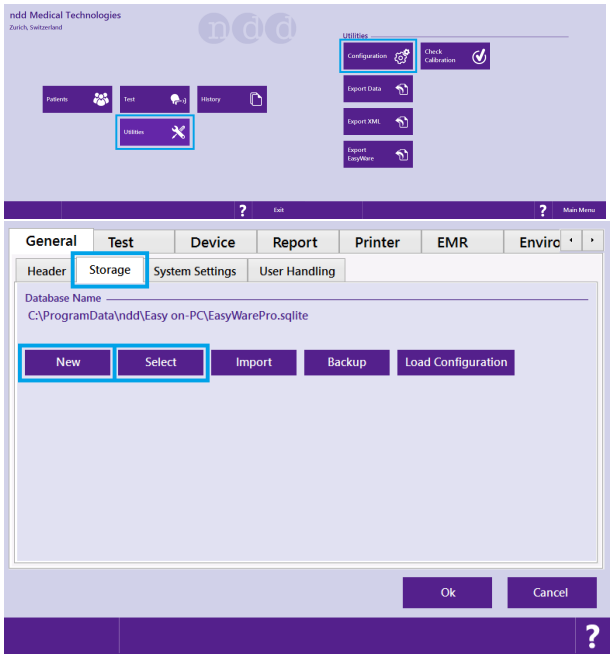
In order to connect to a Microsoft SQL Server based database with two different authentications follow the instructions below.

- **Microsoft Windows authentication** is your Windows account login. Make sure that the user is created and that windows authentication is used.
- **Microsoft SQL Server authentication** is a predefined user with a password.

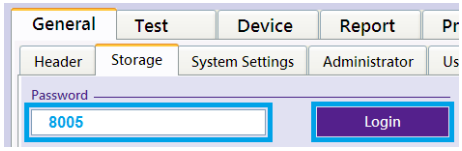
You can configure, create users and adjust database permissions with Microsoft SQL Server Management Studio Express.

1.2.1 Microsoft SQL server based database connection with Microsoft Windows authentication

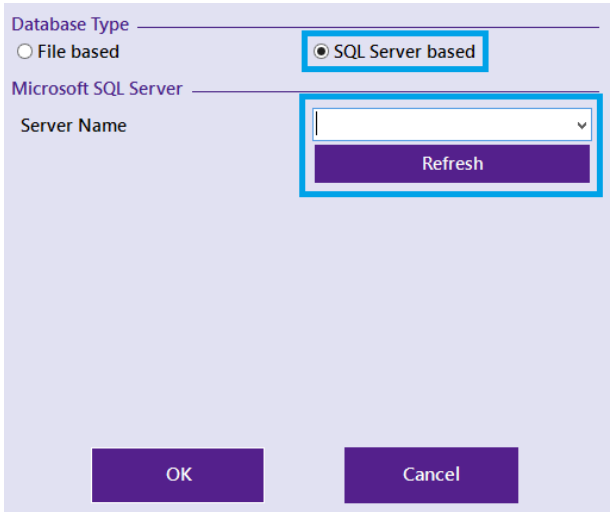
Go to Utilities / Configuration / tab: Storage and click New or Select



For EasyOne Pro / LAB users type in 8005 to log in.



Check the SQL Server based option and type the server name where your Microsoft SQL server is installed or click into the Server Name Combobox and let the system fill in the server names available in your network.

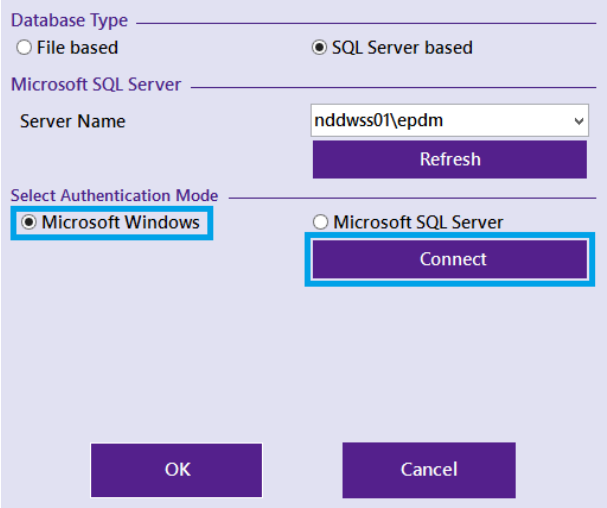


Select the Microsoft Windows authentication mode and click Connect.

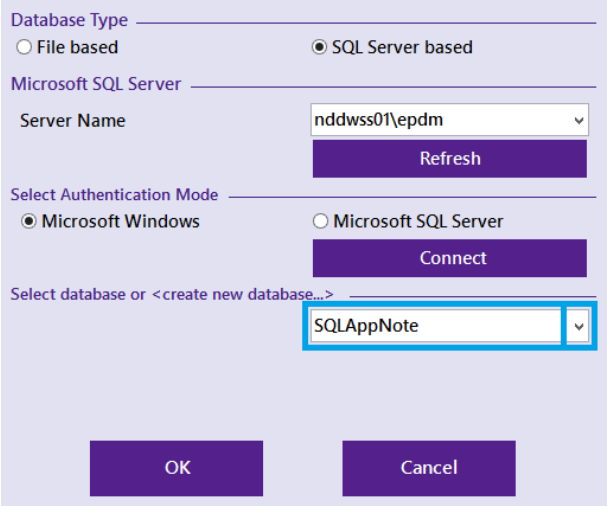
For EasyOne Pro / LAB, configure an nddUser login.

For example:

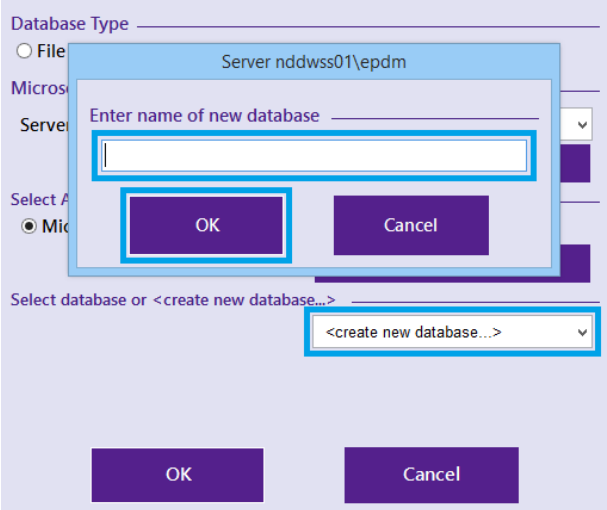
Login Name: nddwss01\nddUser



After successfully connecting, you can select your database in the dropdown list.



In order to create a new database, select ‹create new database...›, enter the name of the new database and click OK.

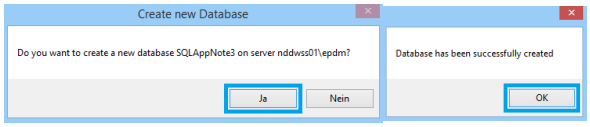


Confirm the database creation with Yes and OK.

You can now find the newly created database in the dropdown list.

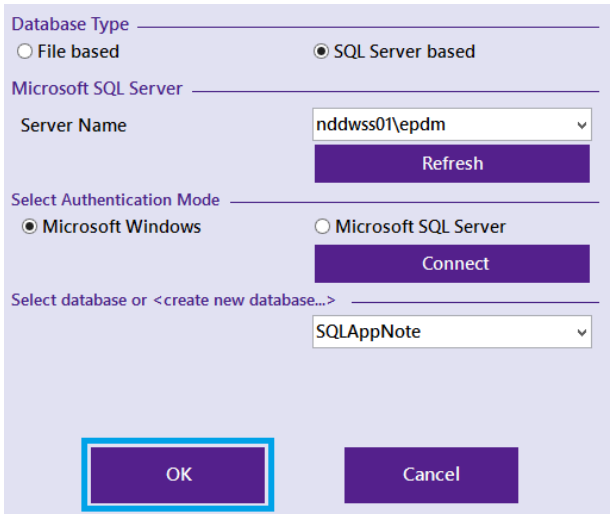
The logged in user needs to have the permissions on the server to create a new database.

The other option would be to use an empty Microsoft SQL Server database (a backup that can be provided from ndd) and load it onto the server manually.



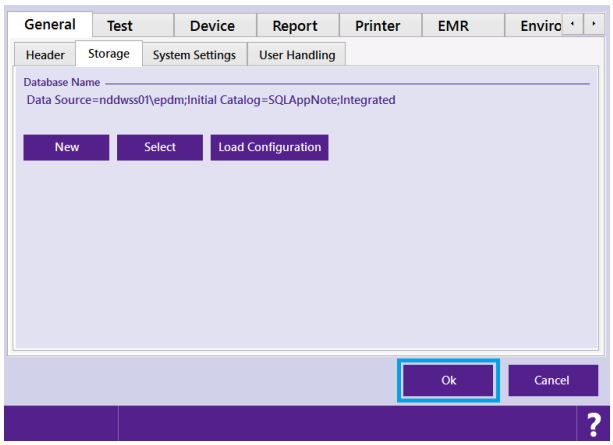
Contact for backup: support@ndd.ch

Click OK



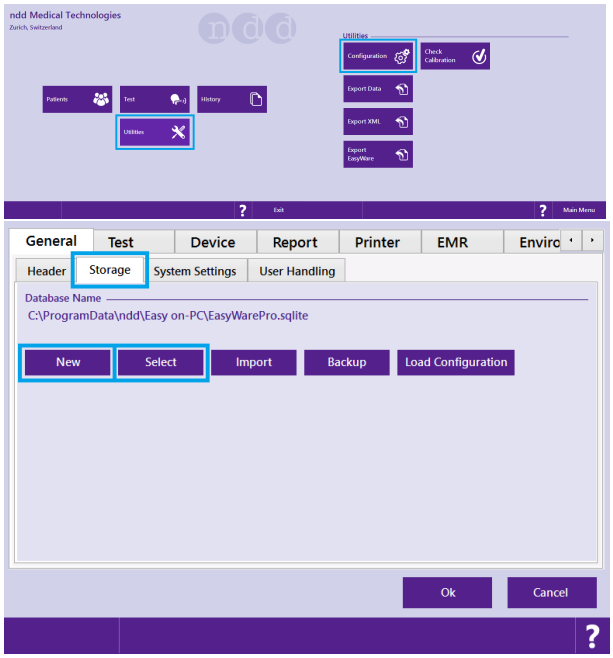
Click OK and the software will restart automatically.

You can now start to use your database.

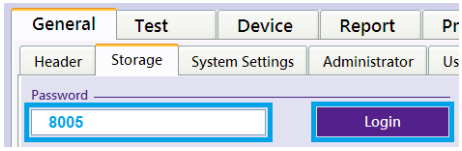


1.2.2 Microsoft SQL server based database connection with Microsoft SQL Server authentication

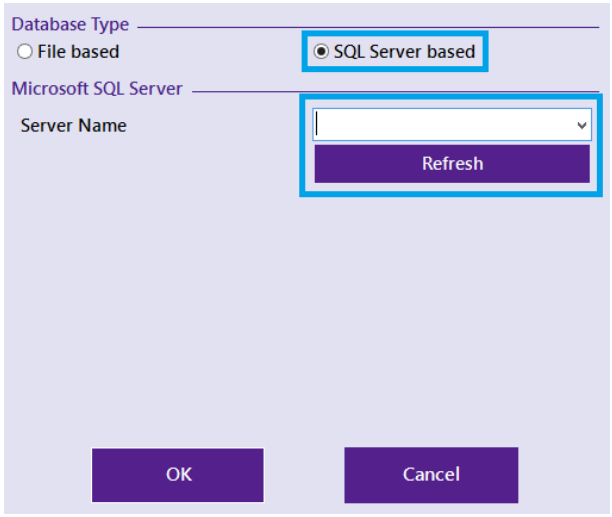
Go into Utilities / Configuration / tab: Storage and click New or Select



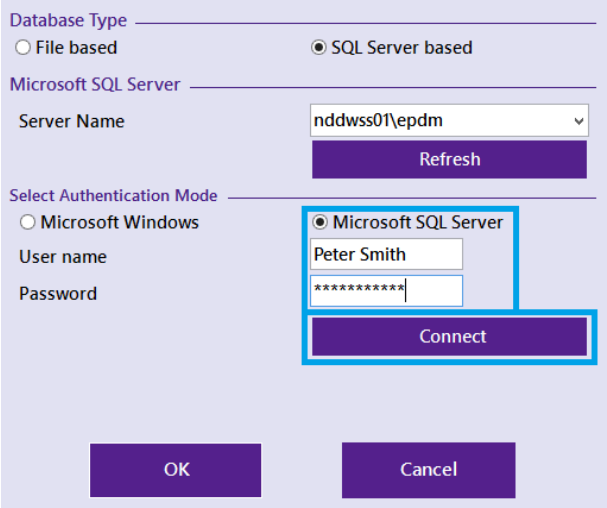
For EasyOne Pro / LAB users type in 8005 to log in.



Check the SQL Server based option and type the server name where your Microsoft SQL server is installed or click into the Server Name Combobox and let the system fill in the server names available in network

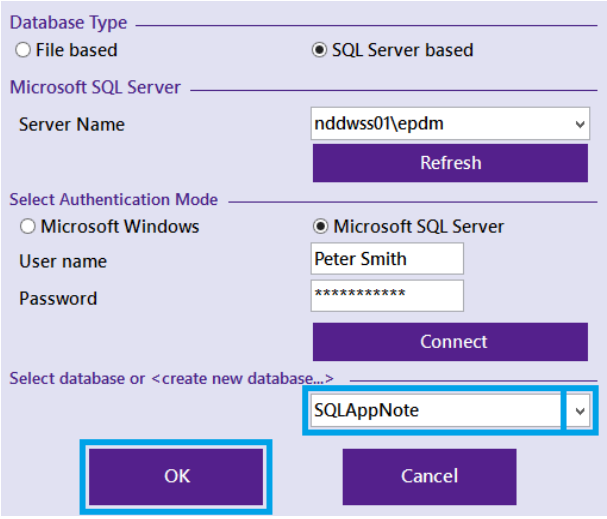


Select the Microsoft SQL Server authentication mode, enter the User name and password and click Connect.



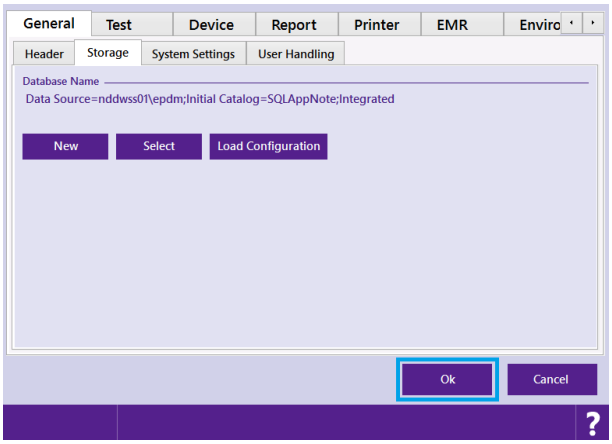
After successfully connecting, you can select your database in the dropdown list and click OK.

See *1.2.1 Microsoft Windows authentication* mode for how to create a new database.

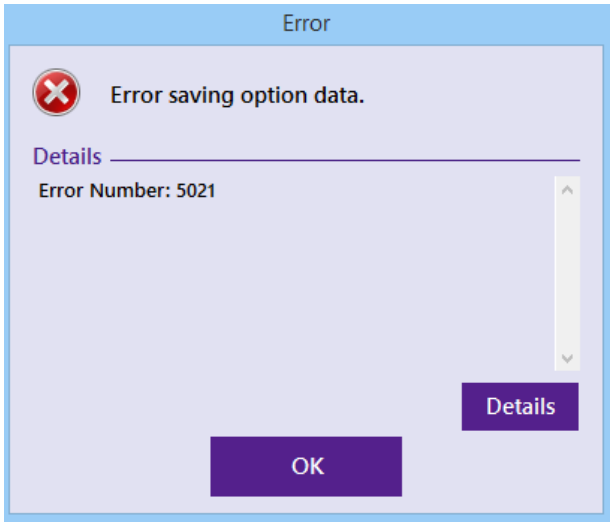


Click OK and the software will restart automatically.

You can now start to use your database.



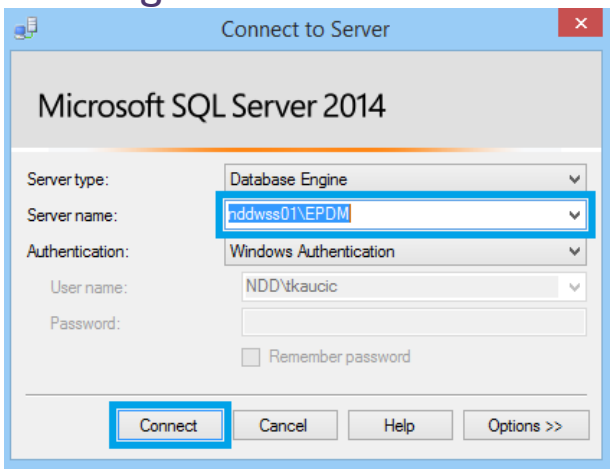
If this error appears, talk to your server administrator. Your Login likely does not have permission to view the database.



2 Microsoft SQL Database management

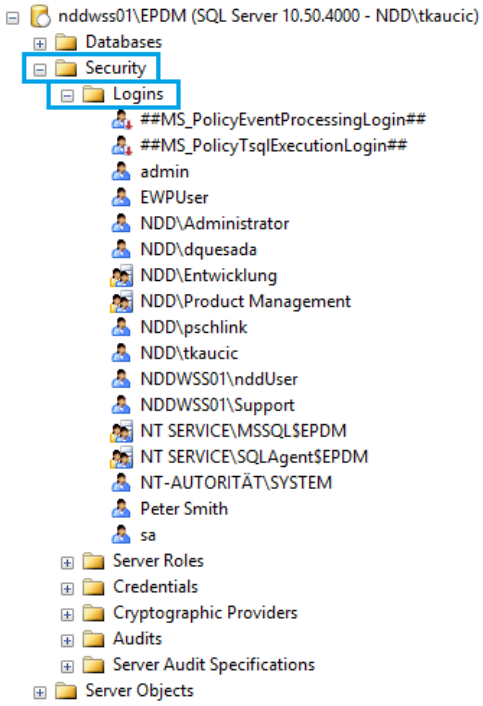
Start your Microsoft SQL Server Management Studio and connect to your server.

Example: nddwss01\EPDM

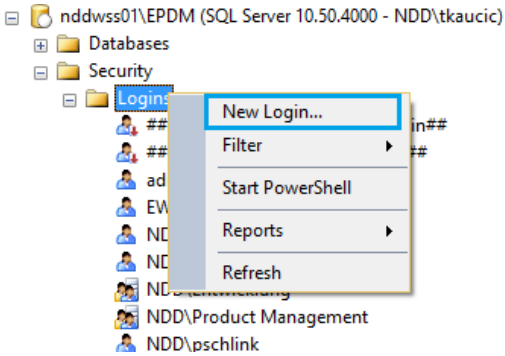


Browse to Security / Logins

Here you can manage the login accounts that have access to your databases.

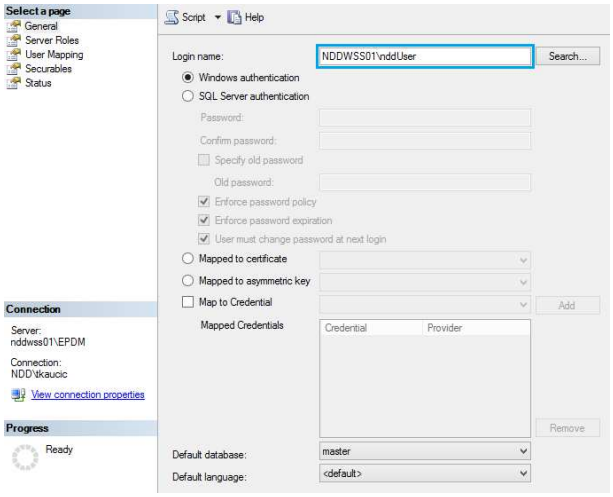


Right clicking on Logins allows you to create new logins.



In order to have access on an SQL server database with an EasyOne Pro / LAB, you need to create a login for the default Windows login of your EasyOne Pro / LAB.

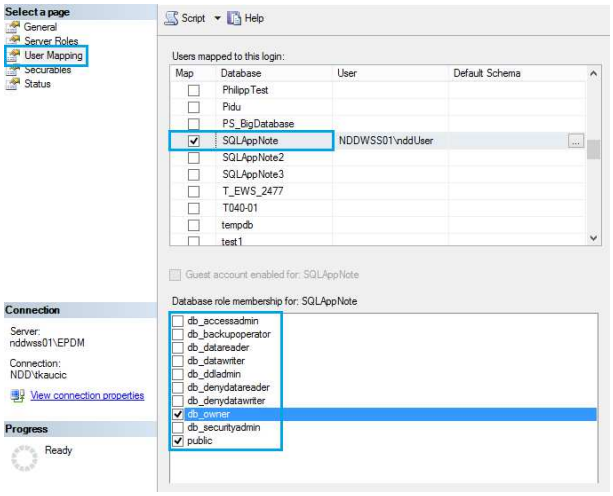
Enter your SQL server name and nddUser into Login Name as shown in the example.



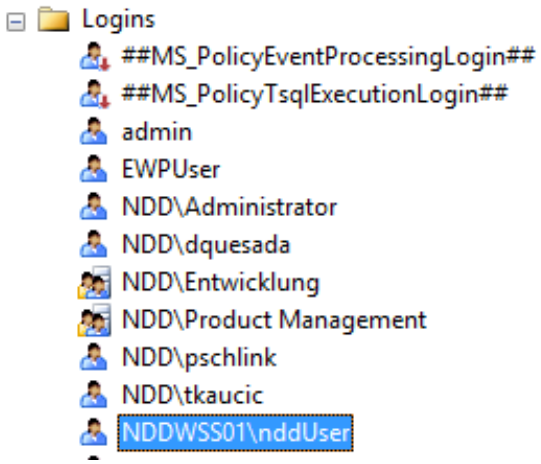
Go to the User Mapping page

Here you can check the databases that you want to have access to on your EasyOne Pro / LAB. Below check what the user can do with this database. For example db_owner has complete access and read and write permissions.

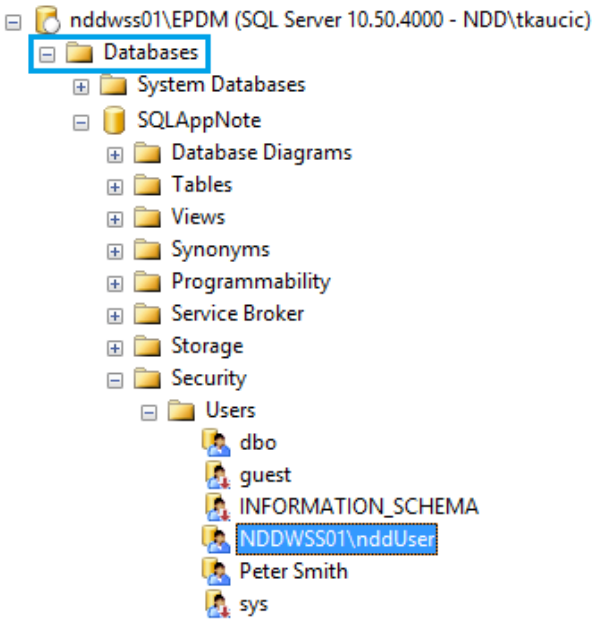
Click OK



You should see the nddUser login account in your Logins now.



You can also check the login entry when you browse into Databases / Name of your database / Security / Users.



3 Encrypting Connections to Microsoft SQL Server

Enabling SSL encryption increases the security of data transmitted across networks between instances of Microsoft SQL Server and applications. Easy on-PC and EasyOne Pro / LAB can make use of Microsoft SQL Server's ability to encrypt data transmission between the server and the application by using Secure Sockets Layer (SSL).

(Excerpts from https://technet.microsoft.com/en-us/library/ms189067(v=sql.105).aspx)

Microsoft SQL Server can use Secure Sockets Layer (SSL) to encrypt data that is transmitted across a network between an instance of SQL Server and a client application. The SSL encryption is performed within the protocol layer and is available to all SQL Server clients except DB Library and MDAC 2.53 clients.

The level of encryption used by SSL, 40-bit or 128-bit, depends on the version of the Microsoft Windows operating system that is running on the application and database computers.

3.1 Create a server certificate

3.1.1 Certificate Requirements

Under "normal" circumstances you don't have to configure any advanced options when creating the certificate / certificate signing request (CSR). If you still want to know the requirements for the SQL Server certificate – refer to https://technet.microsoft.com/en-us/library/ms189067(v=sql.105).aspx / "Certificate Requirements".

3.1.2 Option 1: Certificate signed by a public certificate authority

If the instance of SQL Server is running on a computer that has been assigned a certificate from a public certification authority, identity of the computer and the instance of SQL Server is vouched for by the chain of certificates that lead to the trusted root authority. Such server validation requires that the computer on which the client application is running be configured to trust the root authority of the certificate that is used by the server.

Many public root authorities are pre-configured on your Windows system. View them by executing → "*How to: View Certificates with the MMC Snap-in*" and navigate to Root › Trusted Root Certificate Authorities › Certificates. If you install a certificate issued by one of those authorities, no additional certificate configuration is needed on the client.

3.1.3 Option 2: Self-Signed Certificate

Encryption with a self-signed certificate is possible, but a self-signed certificate offers only limited protection (SSL connections that are encrypted by using a self-signed certificate are susceptible to man-in-the-middle attacks. You should not rely on SSL using self-signed certificates in a production environment or on servers that are connected to the Internet).

Your security policy may determine whether you can use self-signed certificates. Ask your admin or IT security officer.

3.1.4 Option 3: Wildcard certificate (signed by a public certificate authority)

A wildcard certificate secures an unlimited number of subdomains. *.yourcompany.com secures all your hosts / subdomains below your-company.com, e.g. sqlserver.yourcompany.com, www.yourcompany.com, mail.your-company.com etc…

Wildcard certificates are support from SQL Server 2008 R2 and SQL Server 2008 R2 Native Client onwards.

3.2 Install the certificate on the server that runs Microsoft SQL Server

Parts of installation process are described here: https://support.microsoft.com/en-us/kb/316898

- Execute → "*How to: View Certificates with the MMC Snap-in*" using admin privileges on the server that runs MS SQL Server and navigate to the certificates container of the personal certificates.
- Click to select the Personal folder in the left-hand pane.
- Right-click in the right-hand pane, point to All Tasks, and then click Request New Certificate....
- The Certificate Request Wizard dialog box opens. Click Next. Select Certificate type "computer".
- In the Friendly Name text box you can type a friendly name for the certificate or leave the text box blank, and then complete the wizard. After the wizard finishes, you will see the certificate in the folder with the fully qualified computer domain name.

3.3 Activate SSL on Microsoft SQL Server

You can also try to use https://technet.microsoft.com/en-us/library/ms189067(v=sql.105).aspx / "Configuring SSL for SQL Server" – but this method is not always successful. This is why we prefer the registry key method as follows:

The certificate used by SQL Server to encrypt connections is specified in the following registry key:

```
HKLM = HKEY_LOCAL_MACHINE

HKLM\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.x\MSSQLServer\SuperSocketNetLib\Certificate
```

This key contains a property of the certificate known as thumbprint that identifies each certificate in the server.

- Navigate to the certificate store where the FQDN certificate is stored. On the properties page for the certificate, go to the Details tab and copy the thumbprint value of the certificate to a Notepad window.
- Remove the spaces between the hex characters in the thumbprint value in Notepad.
- Start regedit, navigate to the following registry key, and copy the value from step 2:
  `HKLM\SOFTWARE\Microsoft\Microsoft SQL Server\<instance>\MSSQLServer\SuperSocketNetLib\Certificate`

3.4 Install the certificate on the client (only if using a self-signed certificate)

In order for the encrypted SQL Server connection to work, your client must trust your server certificate. As you certificate is not signed by a public certification authority, you must tell your client to trust your server certificate. You will need to install the certificate into the “Trusted Root Certification Authorities” store.

See http://community.spiceworks.com/how_to/1839-installing-self-signed-ca-certificate-in-windows

3.5 Test your client connection

See https://support.microsoft.com/en-us/kb/316898

To test with SQL Server Management Studio, follow these steps:

- Navigate to the SQL Server Client ‹version› Configuration page in SQL Server Configuration Manager.
- In the properties windows, set the Force protocol encryption option to "Yes."
- Connect to the server that is running SQL Server by using SQL Server Management Studio.
- Monitor the communication by using Microsoft Network Monitor or a network sniffer.

Connect to you SQL Server using the same FQDN (fully qualified domain name) as stated in the certificate. Otherwise SQL Server Management Studio will return an exception.

Execute this query in SQL Server Management Studio:

```
SELECT encrypt_option
FROM sys.dm_exec_connections
WHERE session_id = @@SPID
```

Remove the WHERE clause to see all connections.

3.6 How to: View Certificates with the MMC Snap-in

See https://msdn.microsoft.com/en-us/library/ms788967(v=vs.110).aspx

Open the certificates stores for the local computer (step 11 – skip steps 12 + 13)