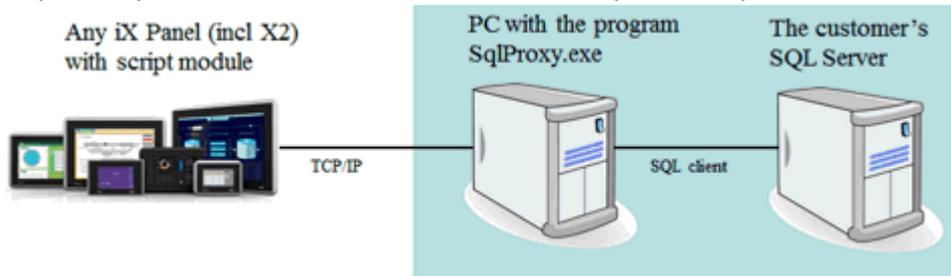


SqlProxy – Connects iX Panels transparently to SQL Server



SqlProxy - Connects iX Panels seamlessly to SQL Server

This goes also for X2 panels running CE8 / Windows Embedded Compact 2013.

There's sometimes a need to connect to SQL Server from an iX Panel. In CE6 based panel, this has been possible using sqlclient and script.

Nothing similar is however supported in CE8 (most X2 panels). CE8 is also called Windows Embedded Compact 2013 ("WEC2013").

During the years, we have helped customers creating script solutions based on SQL Server connectivity.

There is a need to be able to move such an application to X2 aswell with minimal change of script code.

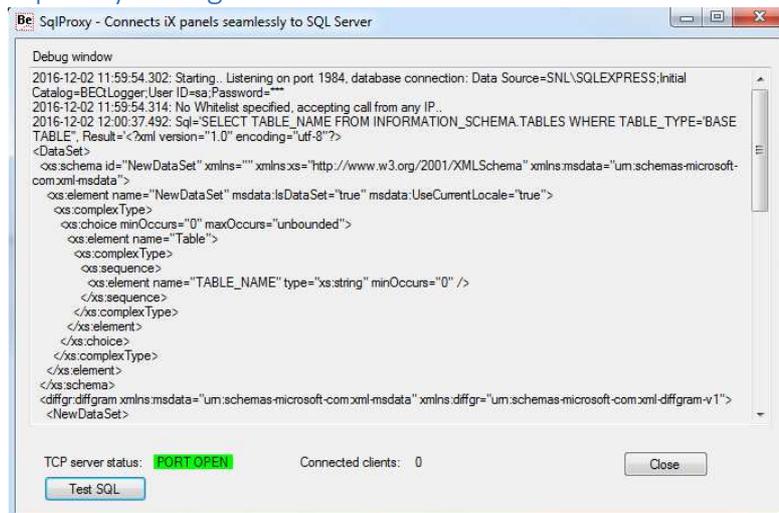
One solution is this "SqlProxy".

SqlProxy.ini settings

```
[Settings]
DBConnection=Data Source=SERVER\SQLSERVER;Initial Catalog=Production;User ID=sa;Password=sqlPr0xy
Port=1984
```

```
Whitelist=
;example:
;Whitelist=192.168.1.1;192.168.1.25
```

SqlProxy debug window



Download

https://www.beijerelectronics.com/en/support/Help_online?code=4D4C0877-4291-43C8-BA7B-54A456B0B5FE

Miscellaneous notes

- Install (copy the files) to e.g. C:\Users\Public\SqlProxy to avoid access right issues!
- See separate additional document about running SqlProxy as a service.
- Time consuming or repeated SQL queries should run from an own thread in iX !
- Using whitelist increases cyber security !
- Support primarily for ordinary sql statements like `select, insert, delete, update etc`
- Limited support for stored procedures, e.g. "spProc1 'Recipe1', returning `RowsAffected`, for StoredProcedure returning Dataset, use prefix " (SPD)", e.g. " (SPD) spProc2 'Recipe1'".

This document is updated for SqlProxy version 1.1.0.0, January 15, 2019.

This version has been tested with unicode data, e.g. Chinese data and SQL queries like `select * from Customers where Name = N'公司黄页'`

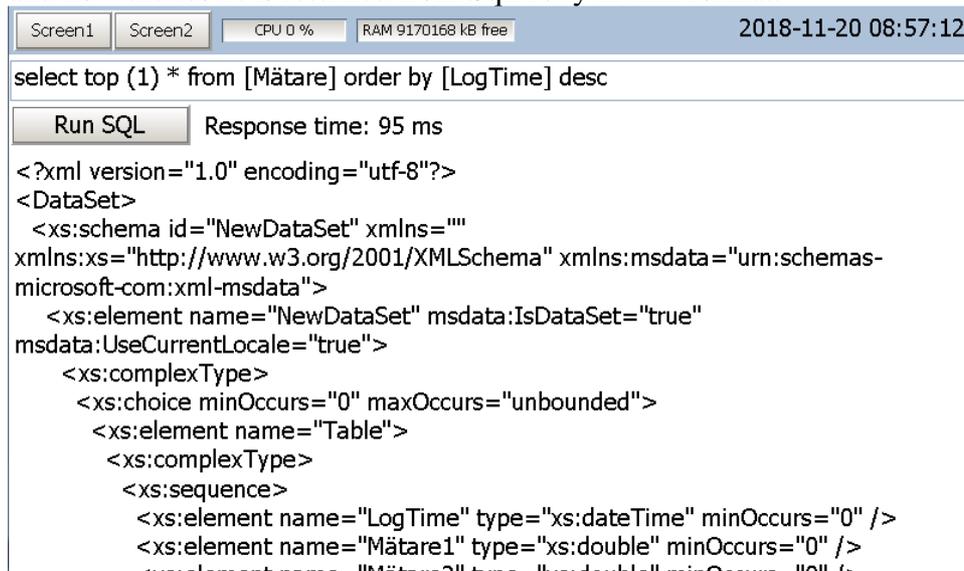
Future improvements

Connectivity: Other database (if strong business case)

iX Demo project

The iX demo project shows how you can use SqlProxy .

Screen1 is demonstrating how you can specify your own SQL query or command and how the result is returned from SqlProxy in xml format.



```
Screen1 | Screen2 | CPU 0 % | RAM 9170168 kB free | 2018-11-20 08:57:12
select top (1) * from [Mätare] order by [LogTime] desc
Run SQL | Response time: 95 ms
<?xml version="1.0" encoding="utf-8"?>
<DataSet>
  <xs:schema id="NewDataSet" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Table">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="LogTime" type="xs:dateTime" minOccurs="0" />
              <xs:element name="Mätare1" type="xs:double" minOccurs="0" />
              <xs:element name="Mätare2" type="xs:double" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
<table border="1">
  <tr>
    <td>2018-11-20 08:57:12</td>
    <td>1.0</td>
  </tr>
</table>
</DataSet>
```

The result could be either RowsAffected or a returned Dataset depending on if you executed a SQL command or query. Or an error if something went wrong. You don't have to parse the xml result yourself. There are built in functions for it all.

```
Globals.scmSqlProxy.isRowsAffected(|
bool scmSqlProxy.isRowsAffected (string xmlData)
int iRowsAffected = Globals.scmSqlProxy.RowsAffected(|
int scmSqlProxy.RowsAffected (string xmlData)

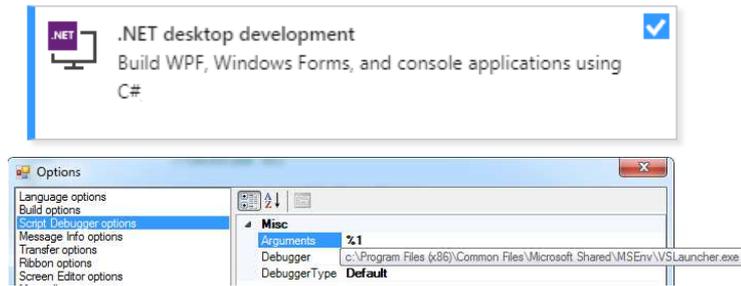
DataTable dt = Globals.scmSqlProxy.FromXml(xml);
//Process DataT DataTable scmSqlProxy.FromXml (string xmlData)
```

Finetune your SQL query so that the amount of returned records in the Dataset are reduced to a minimum! Time consuming or repeated SQL queries should run from an own thread in iX !

```
string sError = Globals.scmSqlProxy.Error(|
string scmSqlProxy.Error (string xmlData)
```

You probably want to debug your script, so you will need Visual Studio (<https://visualstudio.microsoft.com/downloads>) installed and connected as Script Debugger in iX Developer in File | Options

Installing — Visual Studio Community 2017 — 15.8.9



Screen2 showing data from a SQL Server table in a DataGrid control.

| LogTime | Mätare1 | Mätare2 | Mätare3 | Mätare4 |
|------------------|---------|---------|---------|---------|
| 2018-11-20 08:55 | 911,449 | 451,732 | 91,563 | 327,525 |
| 2018-11-20 08:50 | 981,935 | 72,67 | 106,557 | 524,123 |
| 2018-11-20 08:45 | 947,48 | 501,596 | 218,093 | 869,567 |
| 2018-11-20 08:40 | 98,451 | 78,347 | 773,852 | 219,259 |
| 2018-11-20 08:35 | 624,69 | 506,833 | 841,788 | 830,269 |
| 2018-11-20 08:30 | 913,854 | 460,02 | 895,191 | 806,168 |
| 2018-11-20 08:25 | 72,411 | 516,94 | 501,65 | 530,126 |
| 2018-11-20 08:20 | 513,043 | 900,488 | 609,674 | 305,053 |
| 2018-11-20 08:15 | 702,204 | 734,739 | 276,72 | 669,515 |
| 2018-11-20 08:10 | 411,583 | 790,561 | 932,377 | 108,186 |
| 2018-11-20 08:05 | 67,23 | 938,245 | 520,453 | 125,575 |
| 2018-11-20 08:00 | 40,077 | 775,928 | 113,725 | 901,787 |
| 2018-11-20 07:55 | 971,628 | 152,542 | 845,091 | 999,072 |
| 2018-11-20 07:50 | 478,542 | 554,039 | 248,874 | 837,894 |
| 2018-11-20 07:45 | 829,887 | 571,661 | 356,042 | 911,285 |

```

DataTable dt = Globals.scmSqlProxy.FromXml(xml);
//Process DataTable here!
Globals.scmSqlProxy.InitializeGrid(DataGrid, dt)
void scmSqlProxy.InitializeGrid(DataGrid dgView, DataTable table)

```

If you are missing this control, go to Objects | Add Control and add the Windows.Forms DataGrid control!

