https://www.halvorsen.blog

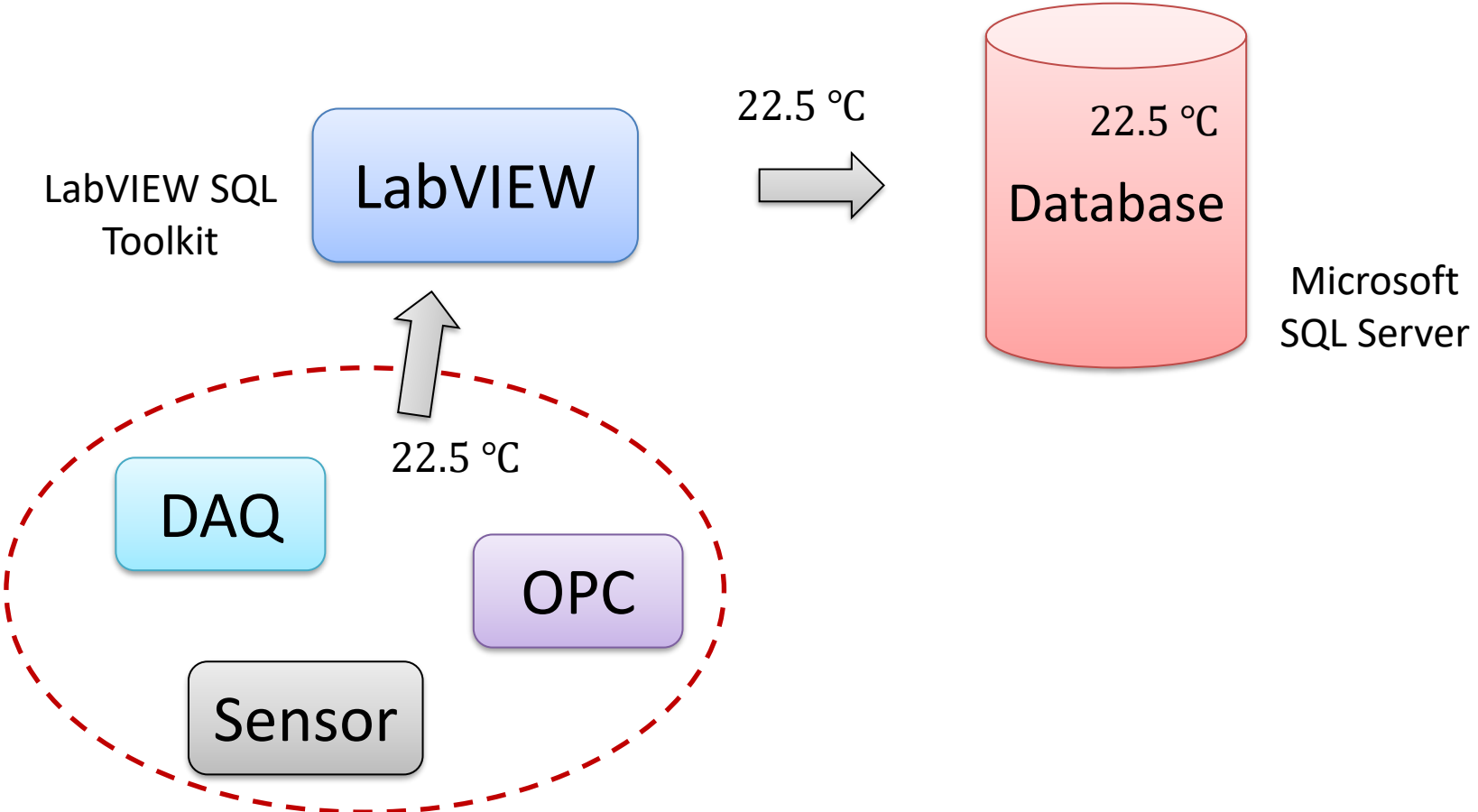
LabVIEW Database Logging

Hans-Petter Halvorsen

Contents

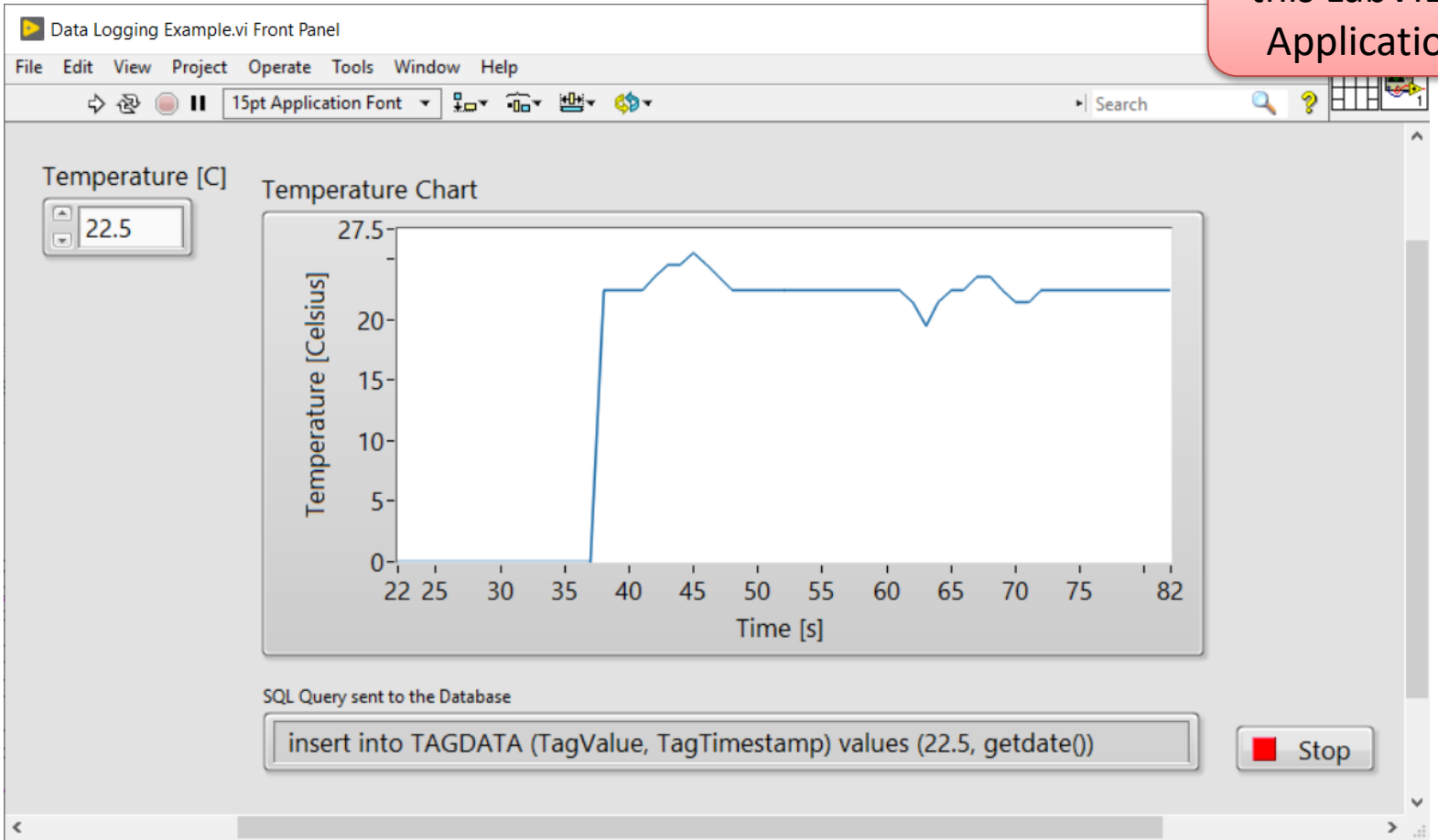
- Create a simple Database used for Logging in SQL Server
- Log Data from LabVIEW into SQL Server

System Sketch



LabVIEW Application

We will create this LabVIEW Application



https://www.halvorsen.blog

SQL Server

Hans-Petter Halvorsen

SQL Server

- SQL Server Express (Free Download)
- Make sure to install both
  - “SQL Server Express” and
  - “SQL Server Management Studio”

https://www.microsoft.com/sql-server/

SQL Server Management Studio

Database.sql - NUCHPH\SQLEXPRESS.MEASUREMENTS (sa (61)) - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

New Query | MEASUREMENTS | Execute

Object Explorer

Connect

- NUCHPH\SQLEXPRESS (SQL Server 15.0.2070 - sa)
  - Databases
    - System Databases
    - Database Snapshots
    - BOOKS
    - CHART
    - LIBRARY
    - MEASUREMENTS
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.TAGDATA
      - Views
      - External Resources
      - Synonyms
      - Programmability
      - Service Broker
      - Storage
      - Security
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Management
  - XEvent Profiler

Database.sql - NU...UREMENTS (sa (61)) | SQLQuery5.sql - N...UREMENTS (sa (59))

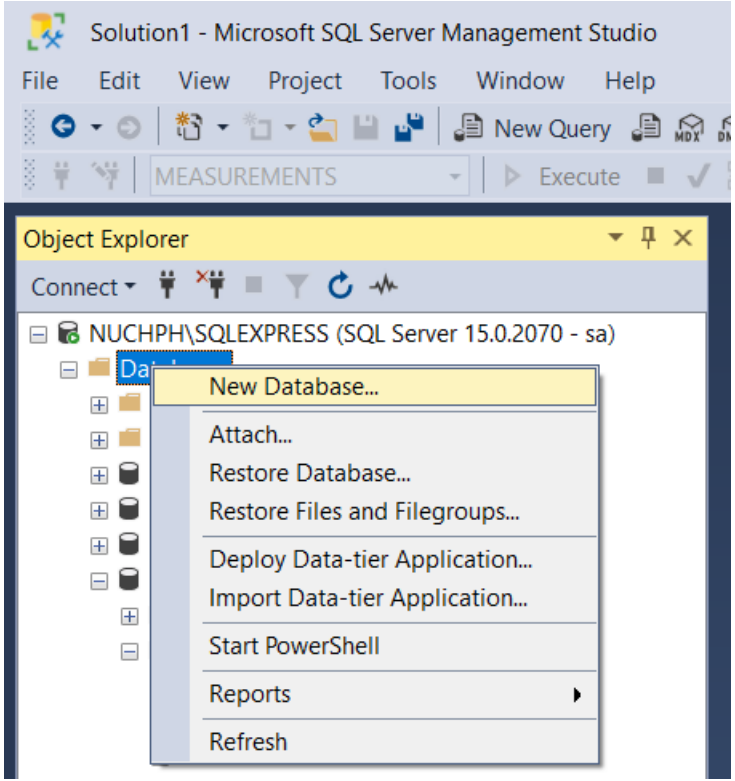
```
CREATE TABLE TAGDATA
(
    TagDataId int Primary Key IDENTITY(1,1),
    TagValue float,
    TagTimestamp datetime
)
```

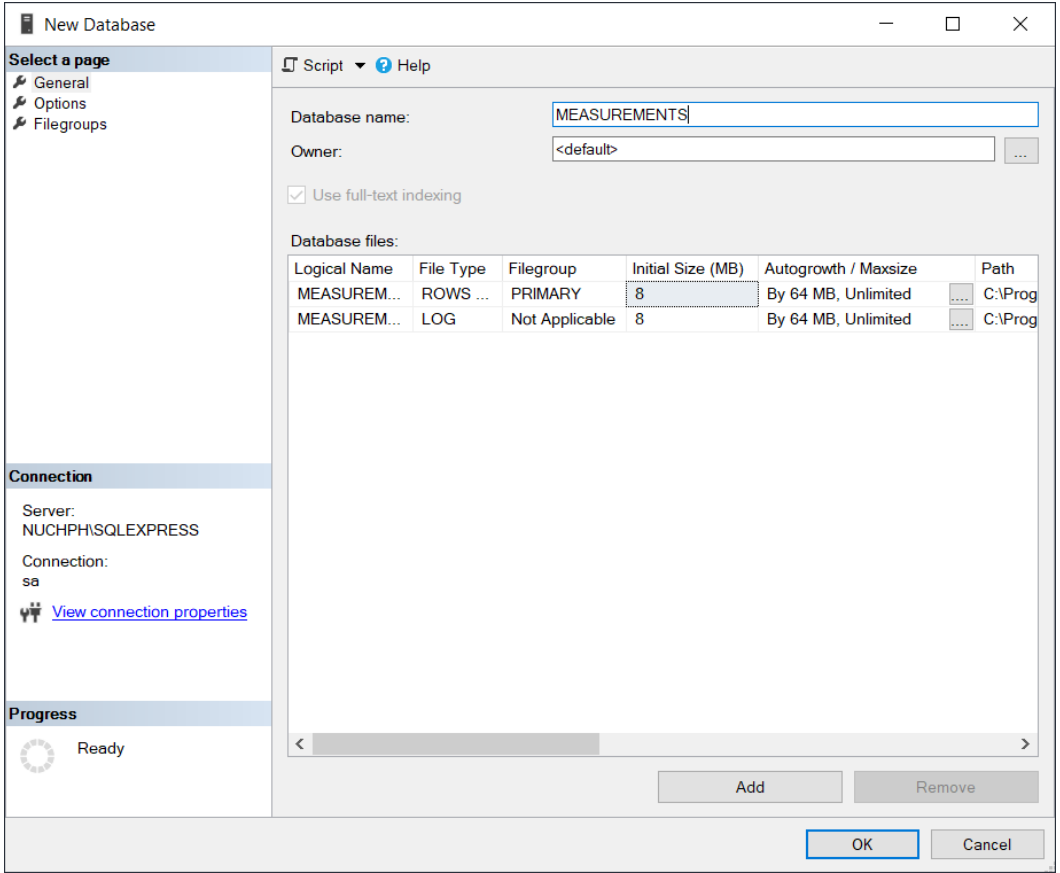
100 %

Connected. (1/1) | NUCHPH\SQLEXPRESS (15.0 RTM) | sa (61) | MEASUREMENTS | 00:00:00 | 0 rows

Ready

Create Database





Database Table

```
CREATE TABLE TAGDATA
(
  TagDataId int Primary Key IDENTITY(1,1),
  TagValue float,
  TagTimestamp datetime
)
```

Create Database Table

Database.sql - NUCHPH\SQLEXPRESS.MEASUREMENTS (sa (61)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query | MEASUREMENTS | Execute

Object Explorer

Connect

- NUCHPH\SQLEXPRESS (SQL Server 15.0.2070 - sa)
  - Databases
    - System Databases
    - Database Snapshots
    - BOOKS
    - CHART
    - LIBRARY
    - MEASUREMENTS
      - Database Diagrams
      - Tables
      - Views
      - External Resources

Execute (F5)

Database.sql - NU...UREMENTS (sa (61)) | SQLQuery5.sql - N...UREM

```
CREATE TABLE TAGDATA
(
    TagDataId int Primary Key IDENTITY(1,1),
    TagValue float,
    TagTimestamp datetime
)
```

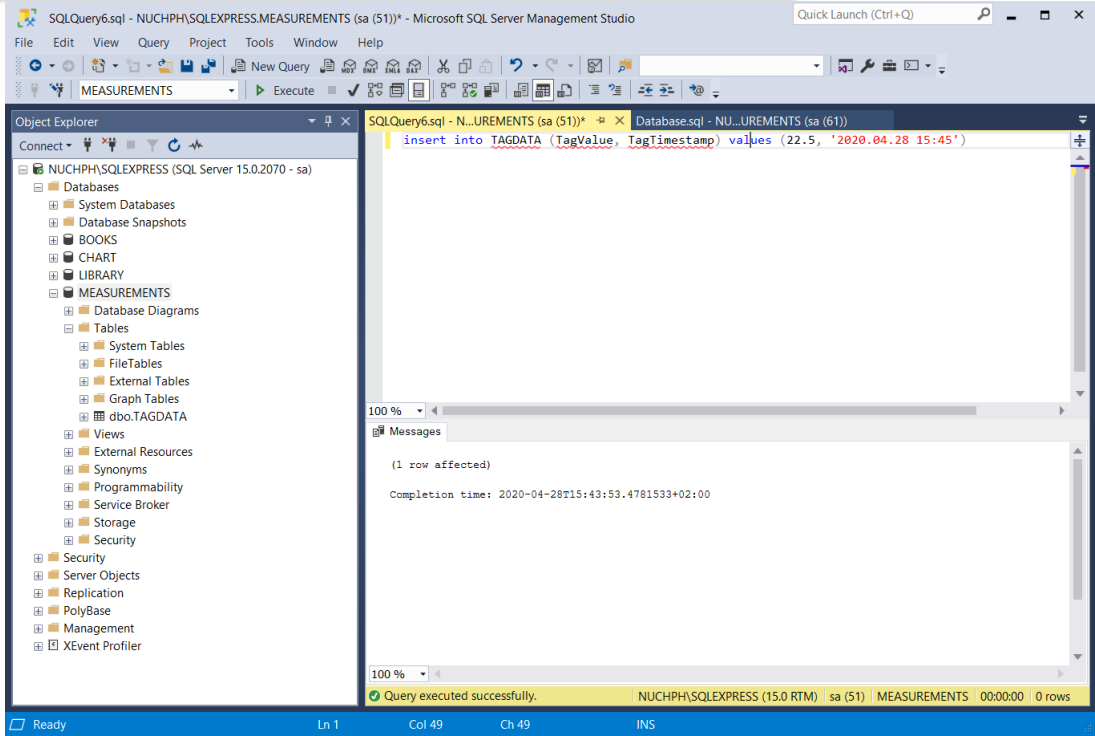
Structured Query Language (SQL)

Value

Date and Time

```
insert into TAGDATA (TagValue, TagTimestamp) values (22.5, '2020.04.28 15:45')
```

Let's insert some data from SQL Server Management Studio before we start creating the LabVIEW Application



Decimal Symbol and SQL

Using Decimal Numbers in SQL can cause problems

Assume we want to insert the Value “22,5”

```
insert into TAGDATA (TagValue, TagTimestamp) values (22,5, '2020.04.28 15:45'))
```

2 columns

SQL assumes you try to insert 3 values into 2 columns since SQL uses “,” as a separation symbol

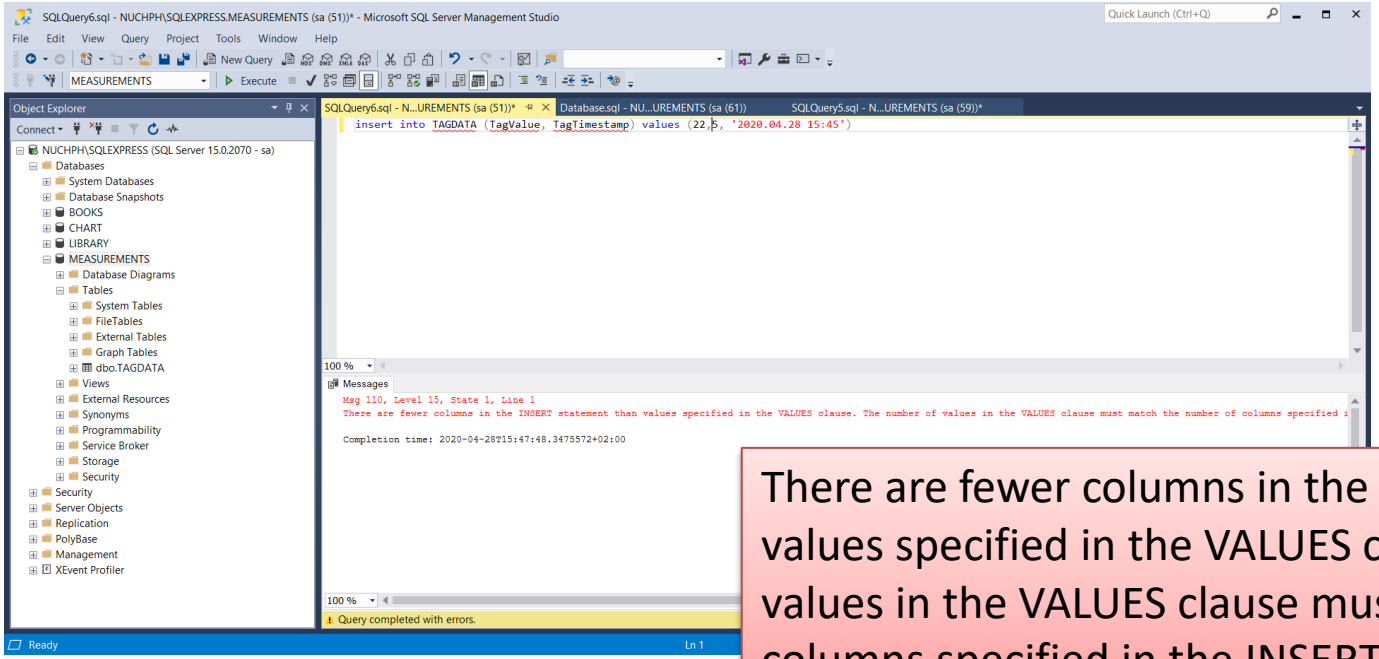
If we use “.” as a Decimal Symbol, Value will then be “22.5”

```
insert into TAGDATA (TagValue, TagTimestamp) values (22.5, '2020.04.28 15:45'))
```

This will work without problems

Error when using wrong Decimal Symbol in SQL

insert into TAGDATA (TagValue, TagTimestamp) values (22,5, '2020.04.28 15:45')



There are fewer columns in the INSERT statement than values specified in the VALUES clause. The number of values in the VALUES clause must match the number of columns specified in the INSERT statement.

Using getdate() function

getdate() is a built-in function in SQL Server

insert into TAGDATA (TagValue, TagTimestamp) values (22.5, getdate())

Let's insert some data from SQL Server Management Studio before we start creating the LabVIEW Application

SQLQuery6.sql - NUCHPH\SQLEXPRESS.MEASUREMENTS (sa (51))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query | MEASUREMENTS | Execute

Quick Launch (Ctrl+Q)

Explorer

CHPH\SQLEXPRESS (SQL Server 15.0.2070 - sa)
Databases
System Databases
Database Snapshots
BOOKS
CHART
LIBRARY
MEASUREMENTS
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.TAGDATA
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security

SQLQuery6.sql - N...UREMENTS (sa (51))* | Database.sql - NU...UREMENTS (sa (61))

```
insert into TAGDATA (TagValue, TagTimestamp) values (22.5, getdate())
```

100 %

Messages

```
(1 row affected)

Completion time: 2020-04-28T15:40:08.0792567+02:00
```

100 %

Query executed successfully. | NUCHPH\SQLEXPRESS (15.0 RTM) | sa (51) | MEASUREMENTS | 00:00:00 | 0 rows

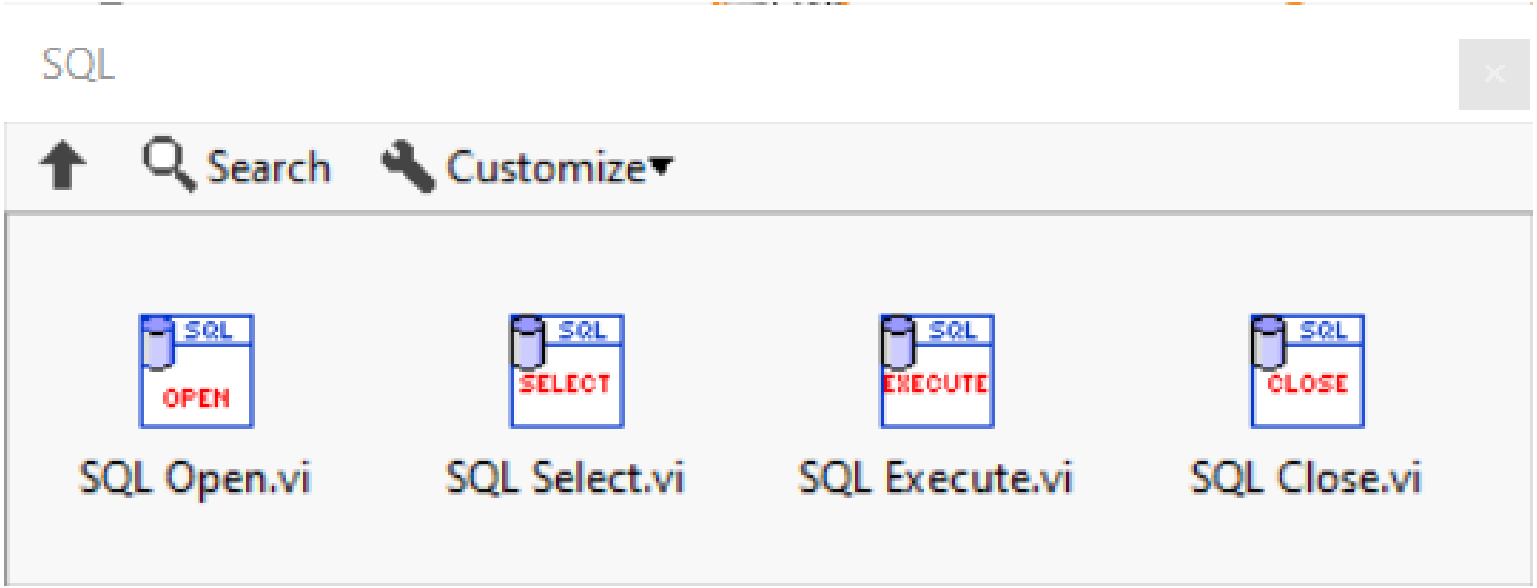
Ready | Ln 5 | Col 1 | Ch 1 | INS

https://www.halvorsen.blog

LabVIEW SQL Toolkit

Hans-Petter Halvorsen

LabVIEW SQL Toolkit (Free Download)



Download for free:
https://www.halvorsen.blog/documents/technology/database/database_labview.php

https://www.halvorsen.blog

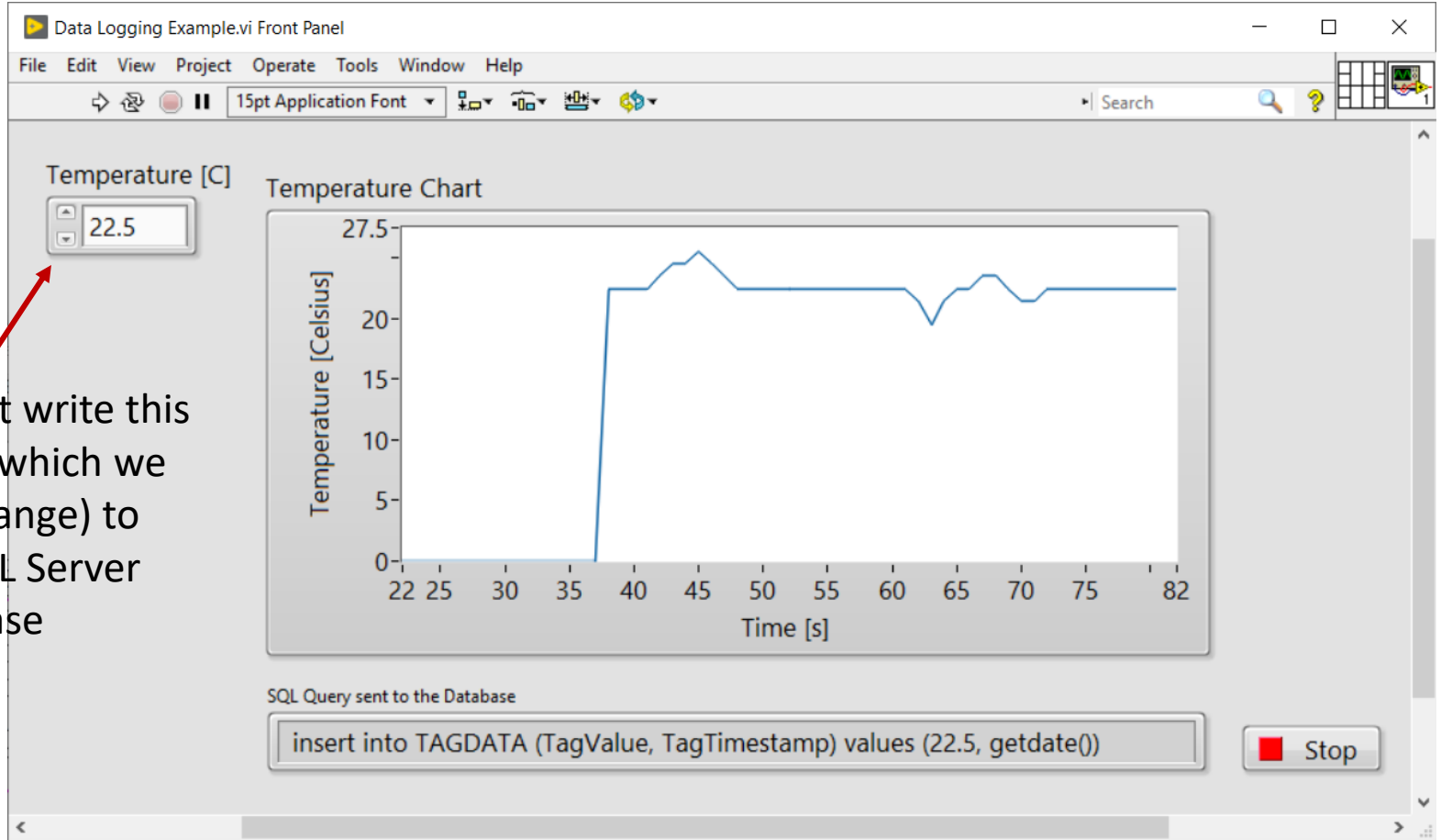
LabVIEW Application

Hans-Petter Halvorsen

LabVIEW Application

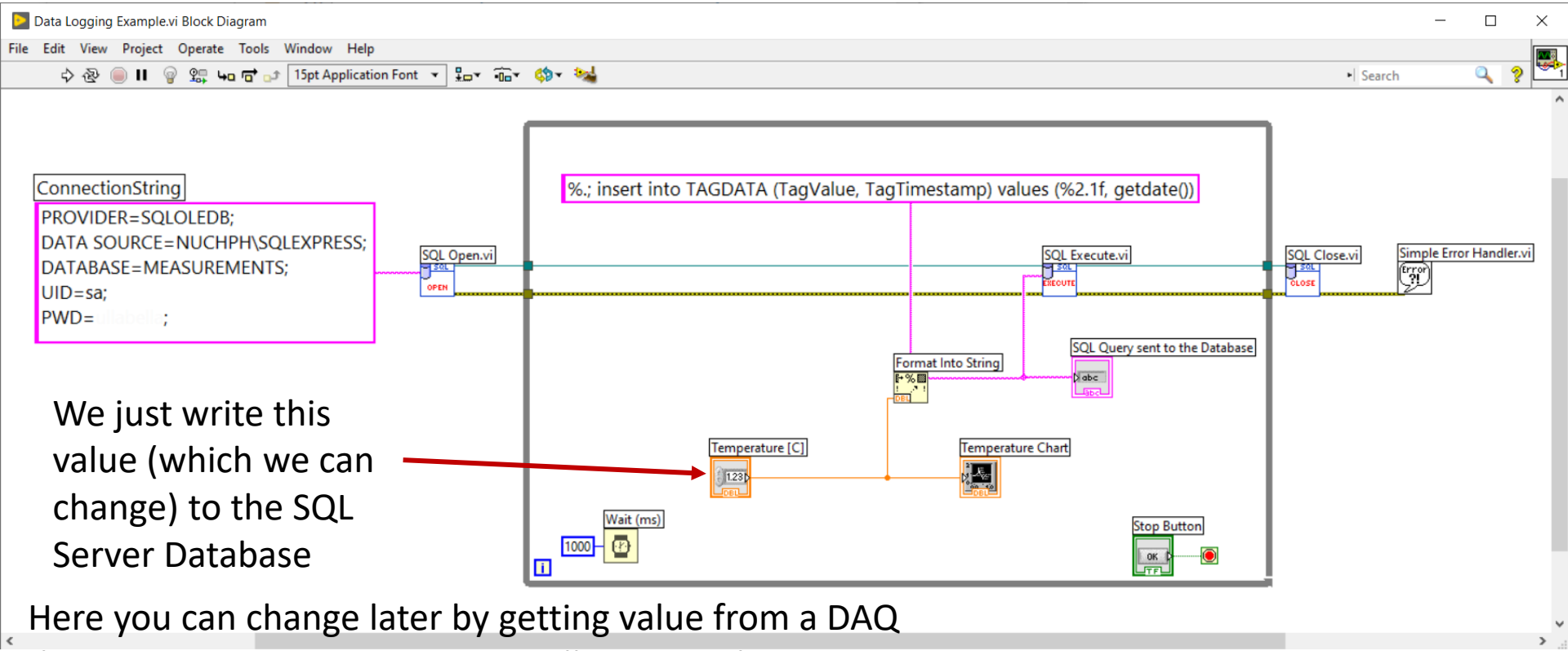
- We will create a LabVIEW Application that inserts the data into the SQL Server Database
- We will show how to connect to the Database with “SQL Open.vi” using either ODBC or Connection String
- We will write to a Database Table using “SQL Execute.vi” in combination with the built-in “Format Into String” function in LabVIEW

LabVIEW Application (Front Panel)



We just write this value (which we can change) to the SQL Server Database

LabVIEW Application (Block Diagram)



We just write this value (which we can change) to the SQL Server Database

Here you can change later by getting value from a DAQ device, an OPC Server, PID Controller, a Simulator, etc.

https://www.halvorsen.blog

How to Connect to Database

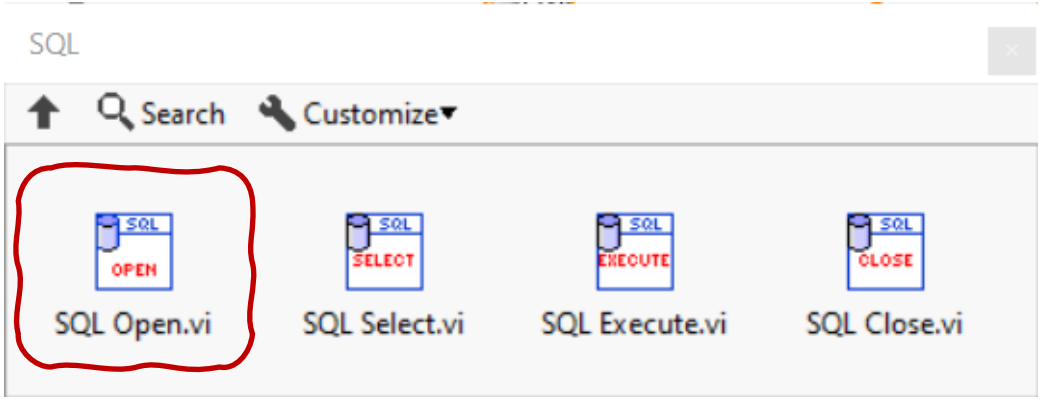
Hans-Petter Halvorsen

Alternatives

We use “SQL Open” to connect to the Dataabase

We can connect to the database in 2 different ways:

- ODBC
- Connection String

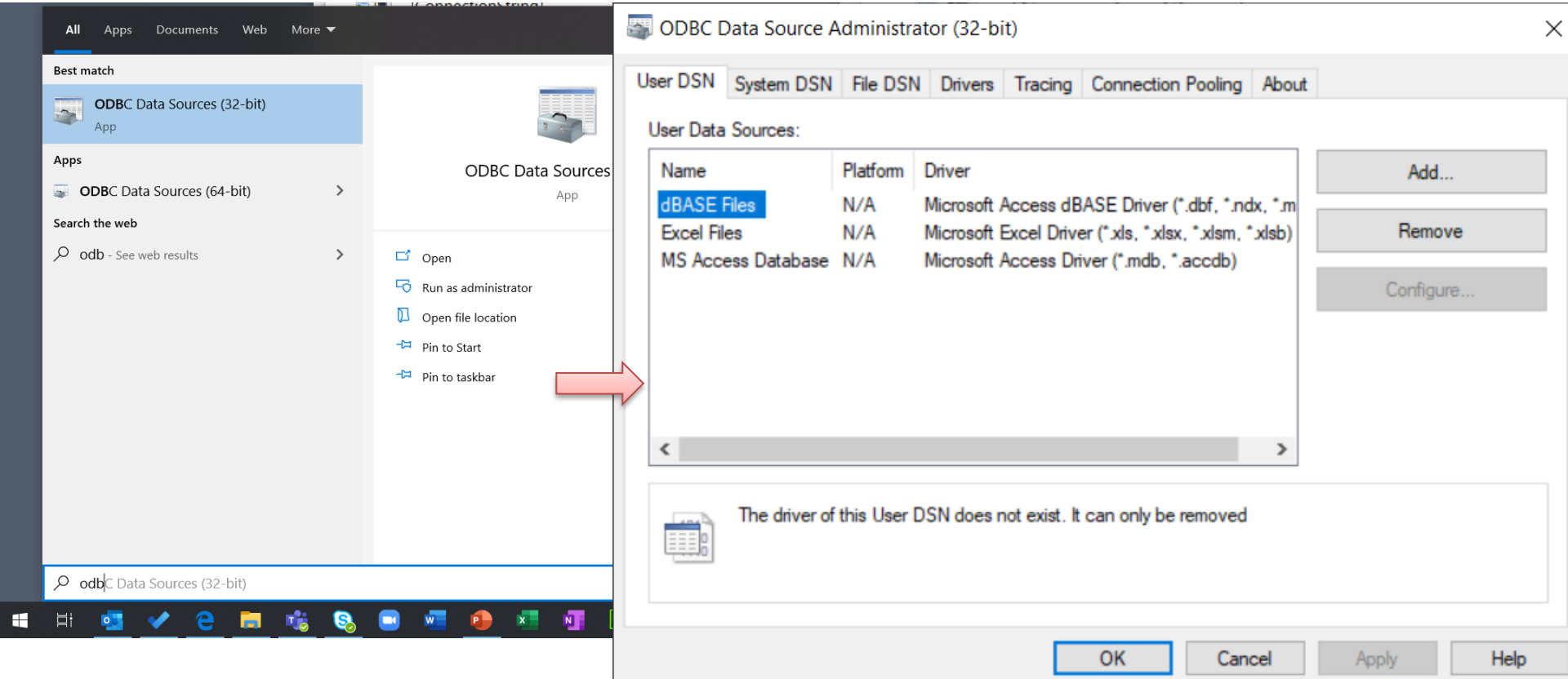


https://www.halvorsen.blog

ODBC

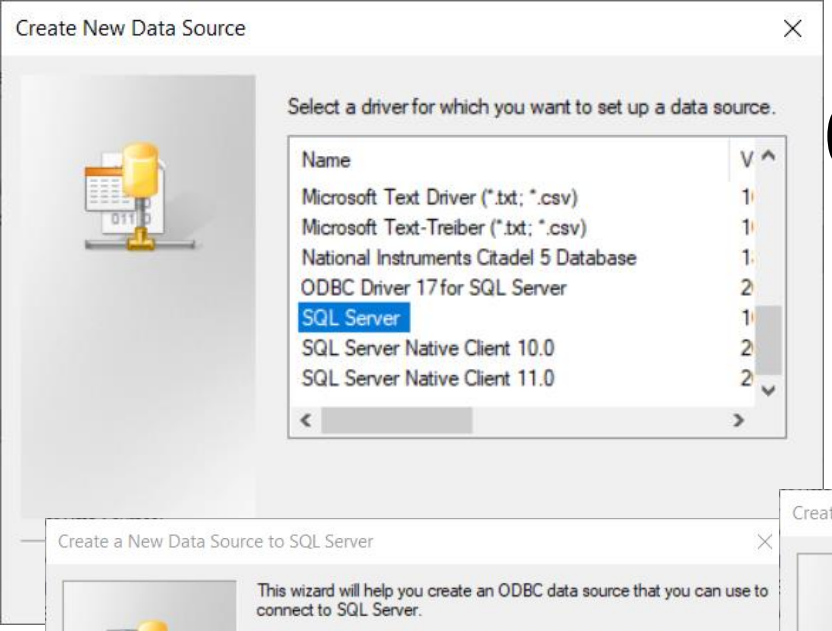
Hans-Petter Halvorsen

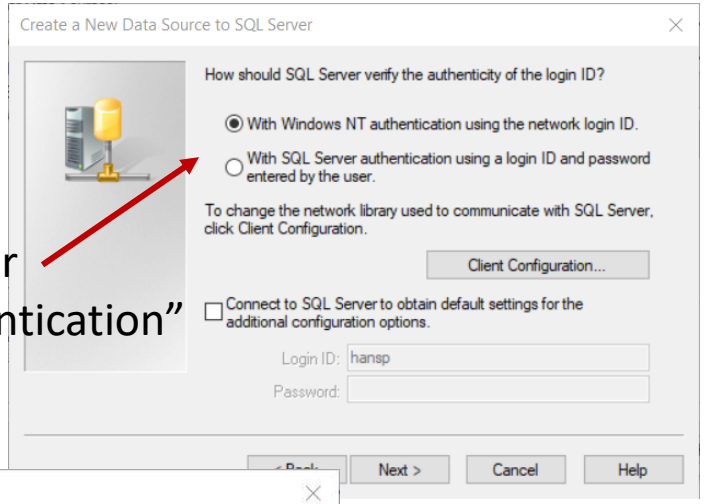
ODBC

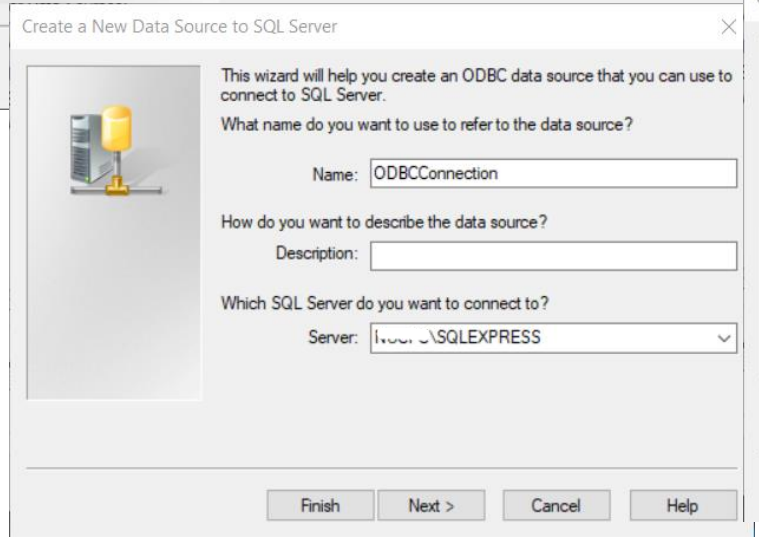

All
Apps
Documents
Web
More
Best match
ODBC Data Sources (32-bit)
App
Apps
ODBC Data Sources (64-bit)
Search the web
odb - See web results
ODBC Data Sources
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
odbC Data Sources (32-bit)
ODBC Data Source Administrator (32-bit)
User DSN
System DSN
File DSN
Drivers
Tracing
Connection Pooling
About
User Data Sources:
Name
Platform
Driver
dBASE Files
N/A
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.m
Excel Files
N/A
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)
MS Access Database
N/A
Microsoft Access Driver (*.mdb, *.accdb)
Add...
Remove
Configure...
The driver of this User DSN does not exist. It can only be removed
OK
Cancel
Apply
Help

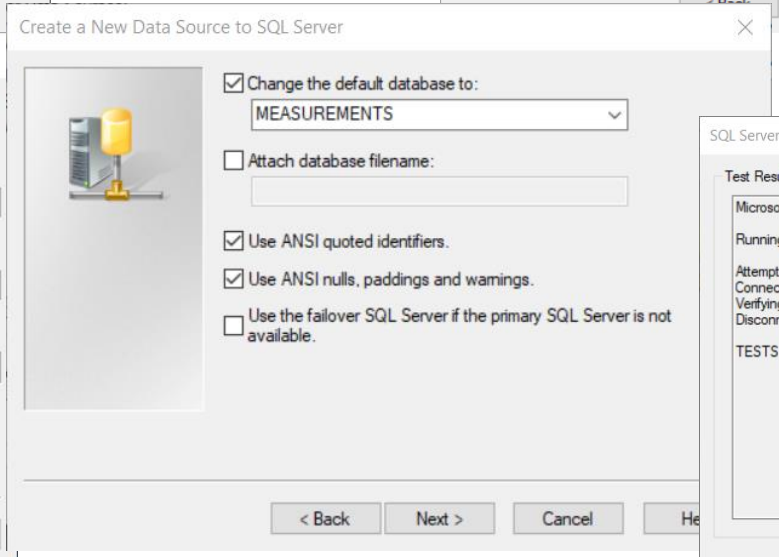
ODBC

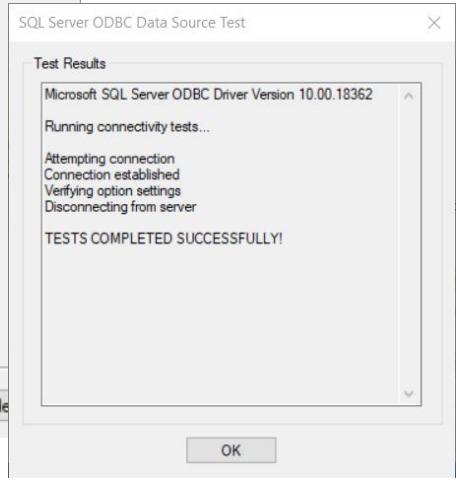
You can use either “Windows Authentication” or “SQL Server Authentication”



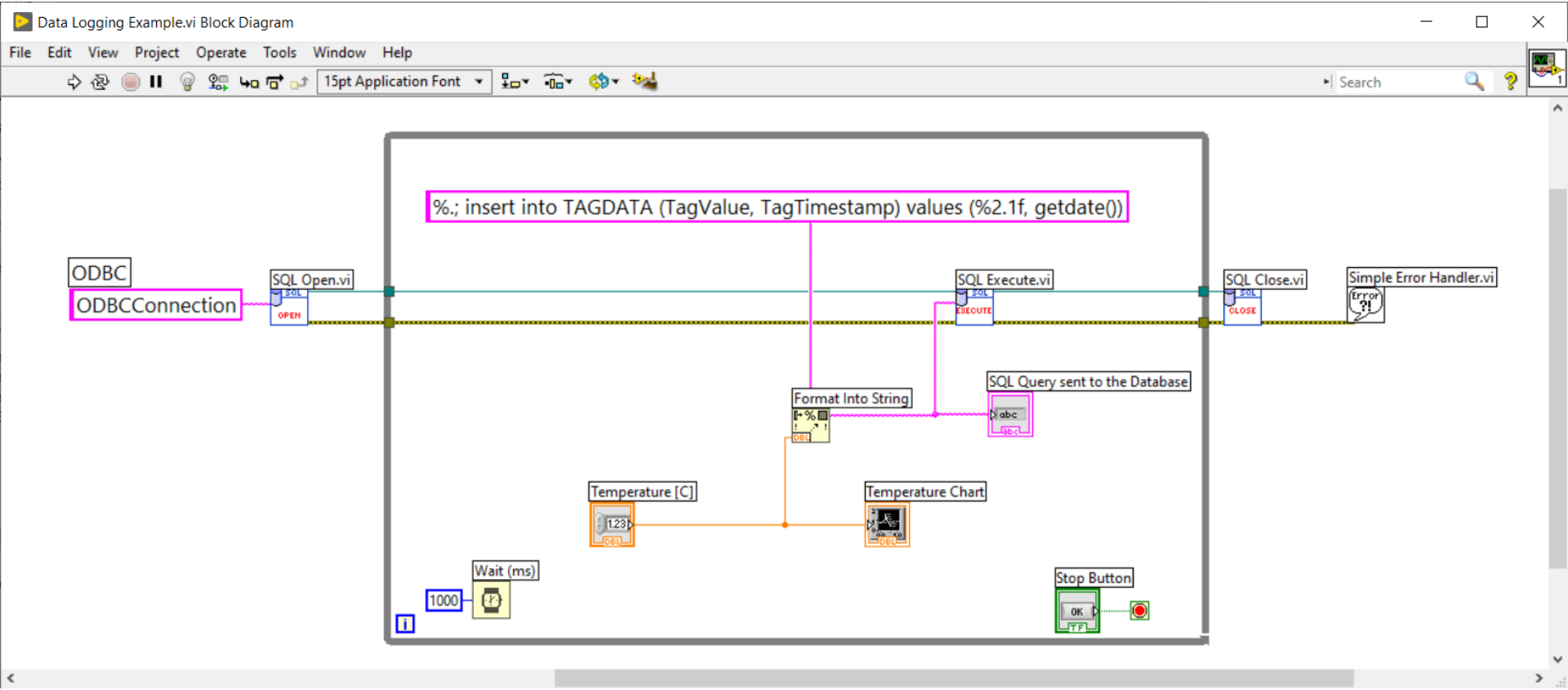








LabVIEW Example



https://www.halvorsen.blog

Connection String

Hans-Petter Halvorsen

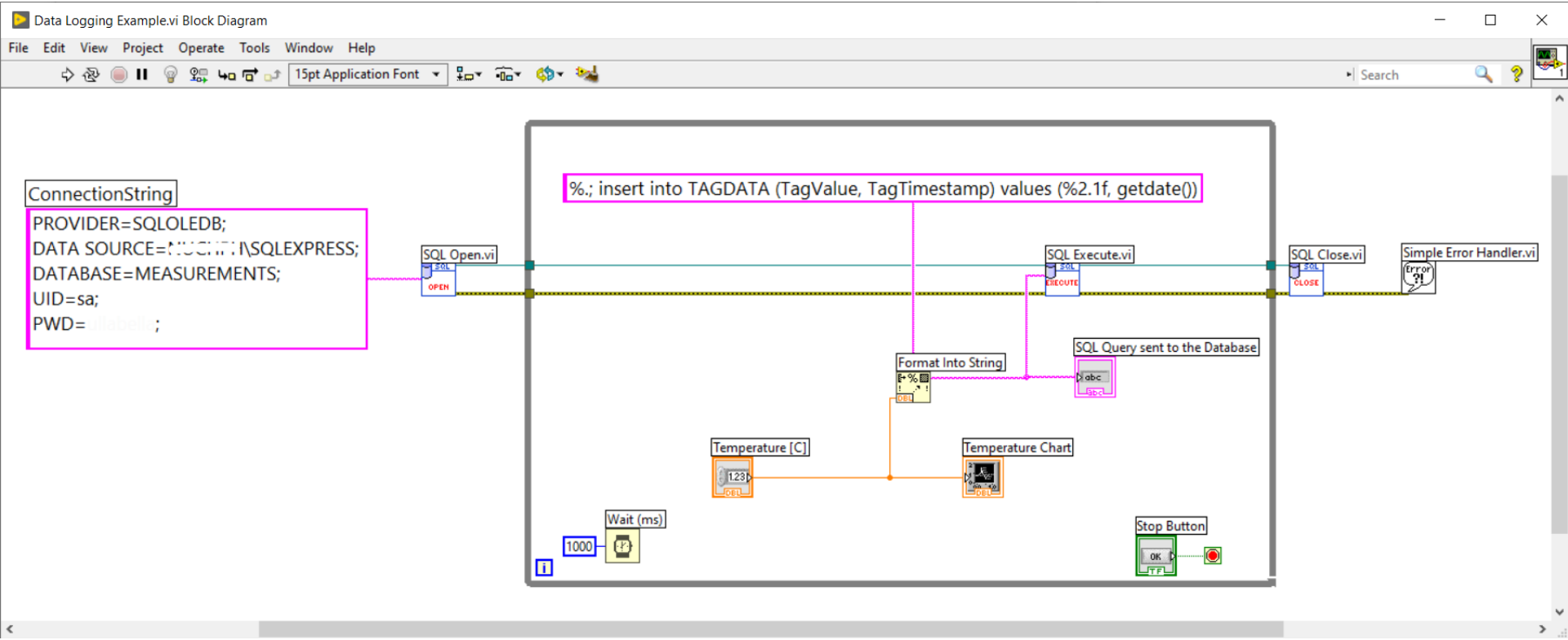
Connection String

PROVIDER=SQLOLEDB;
DATA SOURCE=**COMPUTERNAME**\SQLEXPRESS;
DATABASE=**MEASUREMENTS**;
UID=sa;
PWD=**xxx**;

You can “LOCALHOST” if the Database is on the same computer as your LabVIEW Application

PROVIDER=SQLOLEDB;
DATA SOURCE=LOCALHOST\SQLEXPRESS;
DATABASE=MEASUREMENTS;
UID=sa;
PWD=xxx;

LabVIEW Example



https://www.halvorsen.blog

Decimal Symbol

Hans-Petter Halvorsen

Decimal Symbol and SQL

Using Decimal Numbers in SQL can cause problems

Assume we want to insert the Value “22,5”

```
insert into TAGDATA (TagValue, TagTimestamp) values (22,5, getdate())
```

2 columns

SQL assumes you try to insert 3 values into 2 columns since SQL uses “,” as a separation symbol

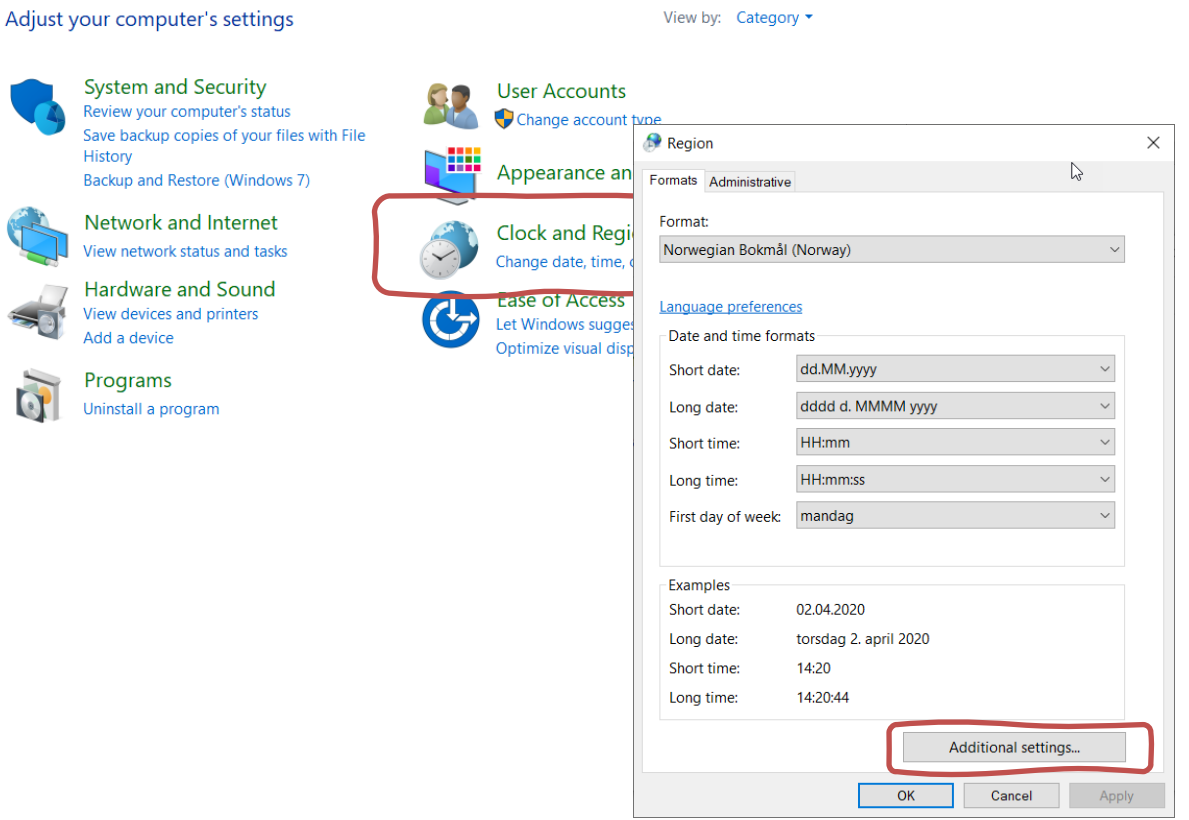
If we use “.” as a Decimal Symbol, Value will then be “22.5”

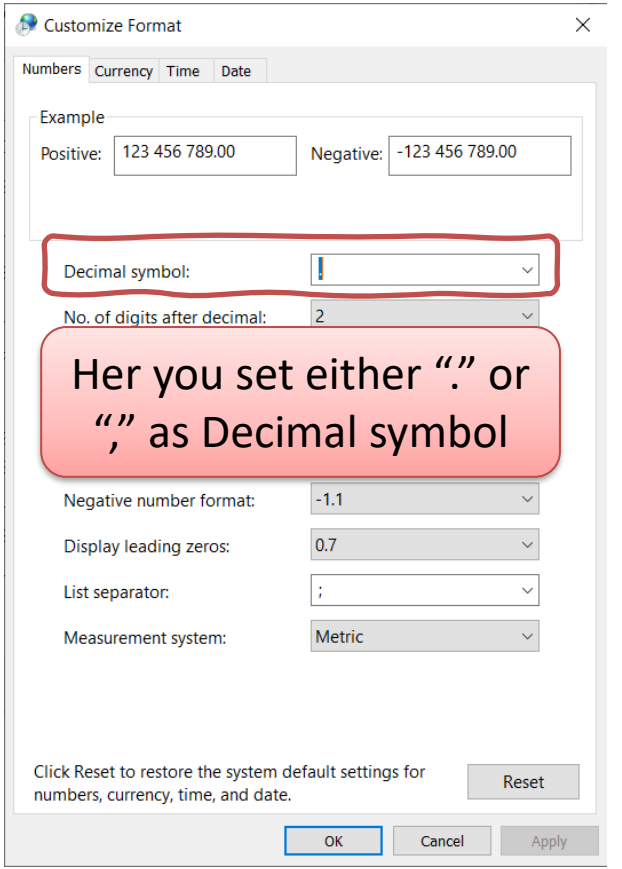
```
insert into TAGDATA (TagValue, TagTimestamp) values (22.5, getdate())
```

This will work without problems

Setting Decimal Symbol in Windows

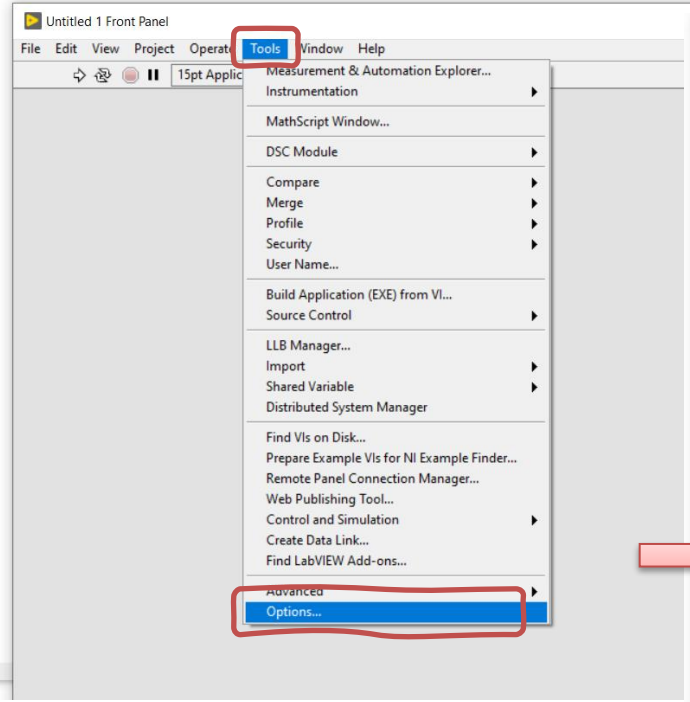
Control Panel

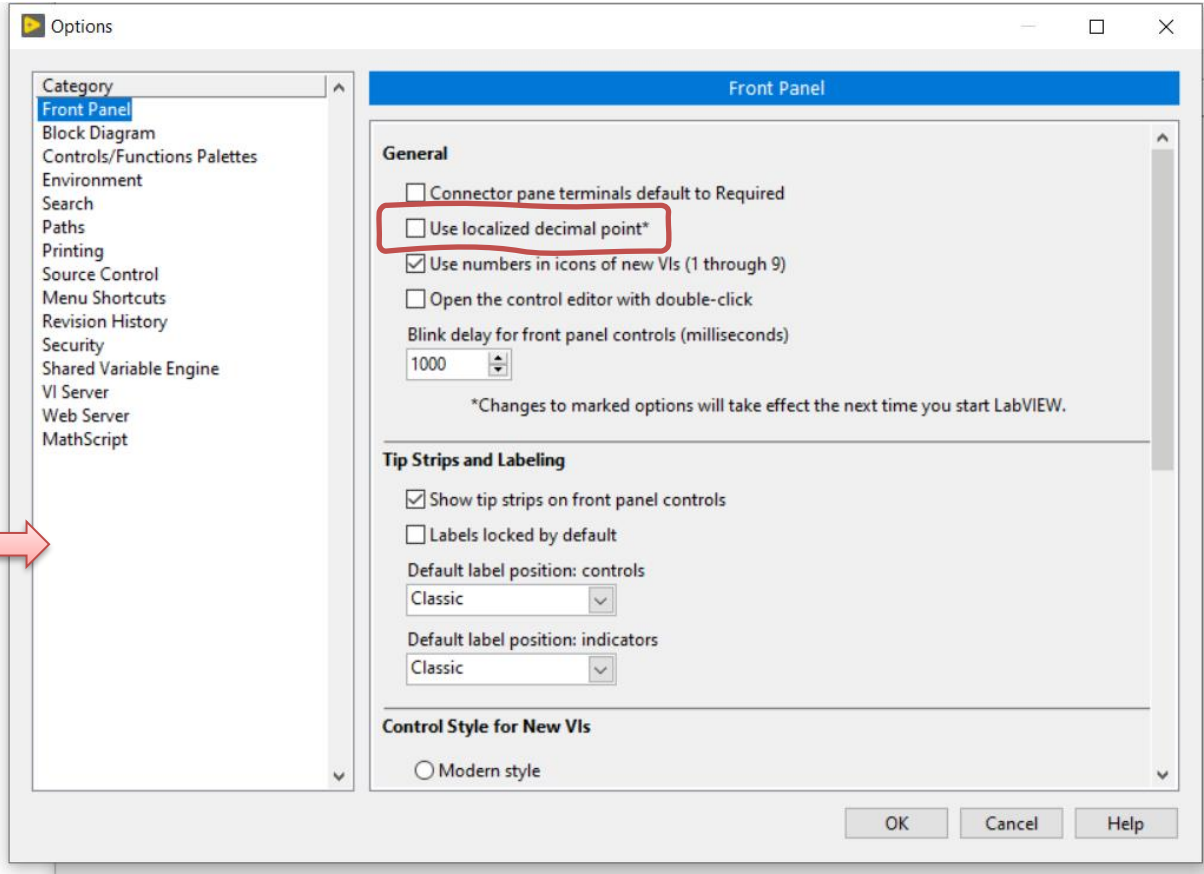




Her you set either "." or "," as Decimal symbol

Setting Decimal Symbol in LabVIEW



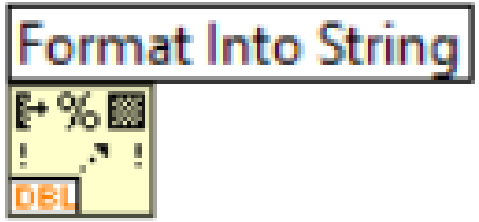


https://www.halvorsen.blog

Format Into String

Hans-Petter Halvorsen

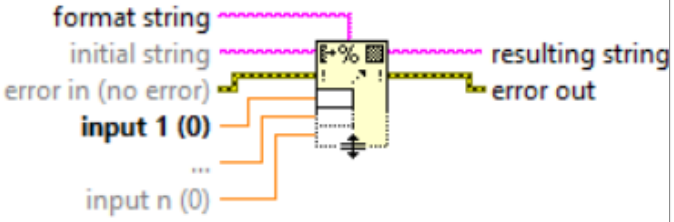
Format Into String



Ctrl + H

Context Help

Format Into String



Formats string, path, enumerated type, time stamp, Boolean, or numeric data as text.

Detailed help

LabVIEW Help

Hide Locate Back Forward Options

Contents Index Search Favorites

- Grouping Data Using Strin
  - Concepts
    - Grouping Data wit
      - Format Specif
    - Grouping Data wit
    - Grouping Data wit
    - Using Matrices
  - How-To
- Local Variables, Global Va
- Graphs and Charts
- Graphics and Sound VIs
- File I/O
- LabVIEW Object-Oriented
- Documenting and Printing
- Programmatically Controllr
- Transferring Data
- Calling Shared Libraries in
- Windows Connectivity
- Formulas and Equations
- Managing Performance ar
- How LabVIEW Stores Dat
- Porting and Localizing VIs
- Application Development
- Signal Processing and An
- Accessing Scanned I/O D
- VI and Function Reference
  - Programming VIs and Fun
    - Application Control VIs
    - Array VIs and Functior
    - Boolean Functions
    - Cluster, Class, & Varia
    - Collection VIs and Fur
    - Comparison Functions
    - Dialog & User Interfac
    - File I/O VIs and Funct
    - Graphics & Sound VIs
    - Report Generation VIs
    - Numeric VIs and Func
    - String Functions
      - Additional String V
      - Array To Spreadsh
      - ASCII Codes
      - Build Text
      - Carriage Return C
      - Concatenate Strin
      - Empty String Cons
      - End of Line Const
      - Flatten/Unflatten
      - Format Codes for t
      - Format Date/Time
      - Format Into String

Format Into String Function

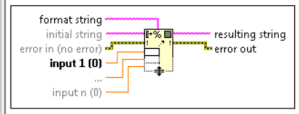
Owning Palette: String Functions

Requires: Base Development System

Formats string, path, enumerated type, time stamp, Boolean, or numeric data as text.

You can use the Format Into String function to convert a number into a string. To format data as text and write the text to a file, use the Format Into File function instead.

Details Example



Add to the block diagram Find on the palette

See next slide

format string specifies how you want the function to convert the input arguments into **resulting string**. Defaults match the data type of the input arguments. Right-click the function and select **Edit Format String** from the shortcut menu to create and edit the format string. Use special escape codes to insert non-displayable characters, the backslash, and the percent characters.

Note This function interprets backslashes as escape characters. To use a literal backslash in **format string**, you must enter `\\`.

initial string specifies the base string to which you can append any arguments to form the **resulting string**.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

input 1..n specifies the input parameters you want the function to convert. This parameter accepts a string, path, enumerated type, time stamp, Boolean, or any numeric data type. For complex numeric data types, this function converts only the real component. You cannot use arrays and clusters with this function. You can specify up to 4,096 characters for each input.

If you specify a Boolean value for this parameter and `%s` as the format code, the Format Into String function outputs the value as TRUE or FALSE. If you specify a Boolean value for this parameter and any numeric format code, the Format Into String function outputs the appropriate version of 1 for TRUE and 0 for FALSE. For example, if you specify `%f` as the format code, the function outputs 1.00000. If you specify `%d`, the function outputs 1.

resulting string contains the concatenation of **initial string** and the formatted output.

error out contains error information. This output provides standard error out functionality.

Format Into String Details

To increase the number of parameters, right-click **input 1** and select **Add Parameter** from the shortcut menu or resize the function.

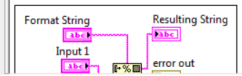
Note If an error occurs, the **error out** cluster element **source** contains a string of the form `Format Into String (arg n)`, where *n* is the first argument for which the error occurred.

If you wire a block diagram constant string to **format string**, LabVIEW uses **format string** to determine the number of outputs and the data type of each output at compile time. If the data types you wire to the outputs do not match the data types determined by **format string**, you must change the output data types before the VI can run.

Specifying Which Input to Use within the Format String

By default, this function uses the order of the inputs to populate the format specifiers, or percent codes in the **Format String**. However, you can use a number followed by a dollar sign (`$`) within a percent code to specify exactly which input to use for that percent code. For example, the percent code `%3$d` uses the third input regardless of how many percent codes appear before `%3$d` in the format string.

Refer to the following block diagram and table for an example of how to use format specifiers:



Format Specifier Syntax

In This Topic

- Format Specifiers Syntax Elements
- Format Specifier Examples

You use format specifiers to format strings, convert a number into a string, and insert non-displayable characters in strings.

For functions that produce a string as an output, such as Format Into String and Array To Spreadsheet String, a format specifier uses the following syntax elements. Double brackets ([]) enclose optional elements.

```
%[$][-][+][#][^][0][Width][.Precision || _SignificantDigits][{Unit}][<Embedded information>]Conversion Code
```

where `Width` must be a number greater than zero and `.Precision` and `_SignificantDigits` must be a number greater than or equal to zero.

For functions that scan a string, such as Scan From String and Spreadsheet String To Array, a format specifier uses the following simplified syntax elements.

```
%[Width]Conversion Code
```

The Format Into String, Format Into File, Scan From String, and Scan From File functions can use multiple format specifiers in the **format string** input, one for each input or output of the expandable function.

Format Specifiers Syntax Elements

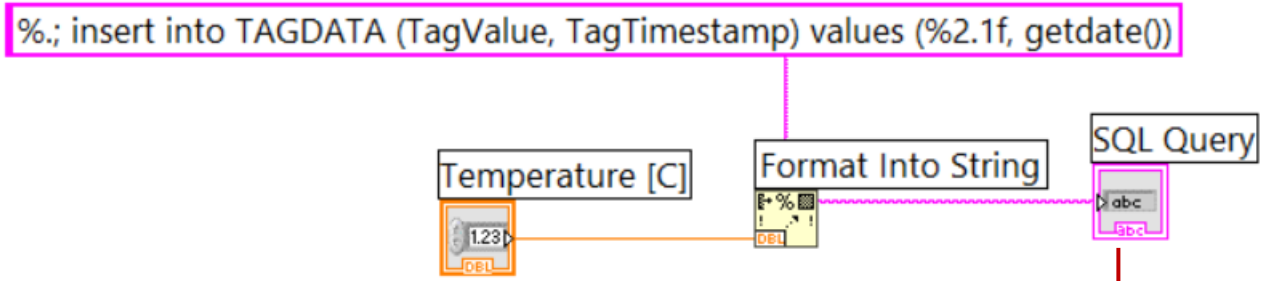
The following table displays the syntax elements for format specifiers. Refer to the examples of format specifiers for more information.

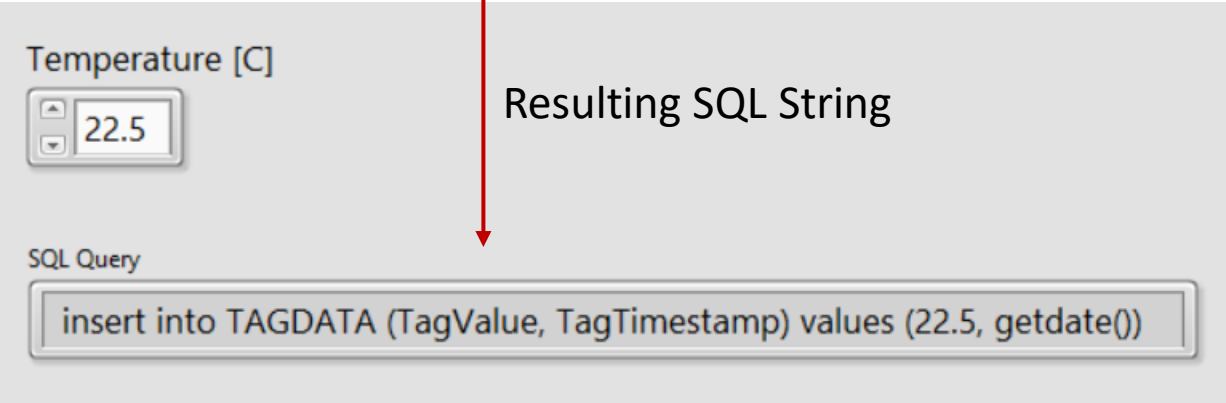
Syntax Element	Description
`%`	Begins the format specifier.
`$` (optional)	When you use a formatting function, this modifier specifies the order in which to display variables. Include the digit that represents the order of the variable immediately before this modifier.
`-` (optional)	When you use a formatting function, this modifier left justifies the parameter rather than right justifies it within its width.
`+` (optional)	When you use a formatting function, this modifier includes sign even when the number is positive.
`^` (optional)	When you use a formatting function and the `e` or `g` conversion codes, this element formats the number in engineering notation, where the exponent is always a multiple of three.
`#` (optional)	When you use a formatting function, this modifier removes trailing zeros. If the number has no fractional part, this modifier also removes the description part.
`0` (optional)	When you use a formatting function, use this modifier without the `-` modifier to pad any excess space to the left of a numeric parameter with zeros rather than with spaces to reach minimum width.
`Width` (optional)	When you use a scanning function, such as Scan From String, the *`Width`* element specifies the maximum character field width to use. LabVIEW scans the maximum number of characters when processing the parameter. When you use a formatting function, the *`Width`* element specifies the minimum character field width of the output. This width is not a maximum width. LabVIEW uses as many characters as necessary to format the parameter without truncating it. LabVIEW pads the field to the left or right of the parameter with spaces, depending on justification. If *`Width`* is missing or `0`, the output is only as long as necessary to contain the converted input parameter.
`.Precision` or *`_Significant Digits`* (optional)	When you use a formatting function, `.` or `_` controls the number of digits displayed. If you use `.`, LabVIEW uses the number that follows as a precision specifier for digits to the right of the decimal point. If you use `_`, LabVIEW uses the number that follows as the specified number of significant digits to use in the display format. *`.Precision`*—When you use it with floating-point notation, this element specifies the number of digits to the right of the decimal point. If `.` is not present, LabVIEW uses a precision of six digits. If `.` is `0`, LabVIEW does not insert a precision. When you use it with string parameters, *`.Precision`* specifies the maximum width of the scanned field. LabVIEW truncates strings longer than this length. *`_Significant Digits`*—Displays the data by rounding to the number of digits you specify. LabVIEW rounds the data only for display purposes, which does not affect the original data. *`.Precision`* affects only the digits to the right of the decimal point, and *`_Significant Digits`* includes all non-spacing digits. For example, • 3.457 has 4 significant digits • 0.0012 has 2 significant digits • 123000 has 3 significant digits **Note** You cannot use precision and significant digits together in a single format specifier.

Localization Codes	Characters that determine if LabVIEW uses a decimal or comma to separate the whole number from the decimal part of the number. These codes control the decimal separator for numeric output. These codes do not cause any input or output to occur. They change the decimal separator for all further inputs and outputs until they find the next `%;`.	
	`%,;`	Comma decimal separator.
	`%.;`	Period decimal separator.
	`%;`	System default separator. If you do not specify a separator, LabVIEW uses the system default separator.

Format Into String Example

"%.;" in front of the string means that "." will be used as Decimal Point

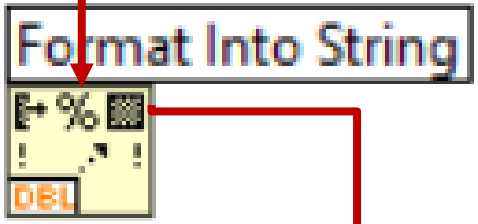




Format Into String

Format String:

%.; insert into TAGDATA (TagValue, TagTimestamp) values (%2.1f, getdate())

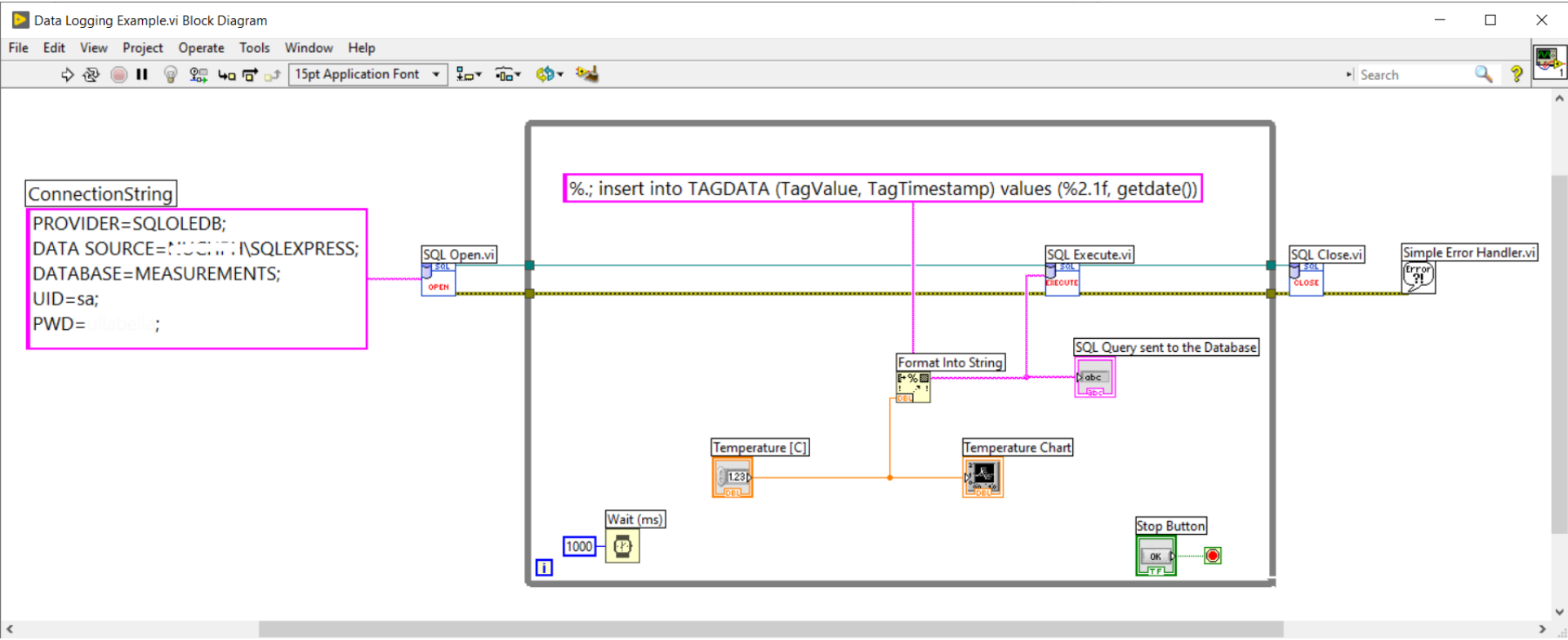


Number

Resulting String:

insert into TAGDATA (TagValue, TagTimestamp) values (22.5, getdate())

Final Solution



Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog

YouTube: https://www.youtube.com/IndustrialITandAutomation