

Technical Workshop

Topic: TwinCAT Database Server

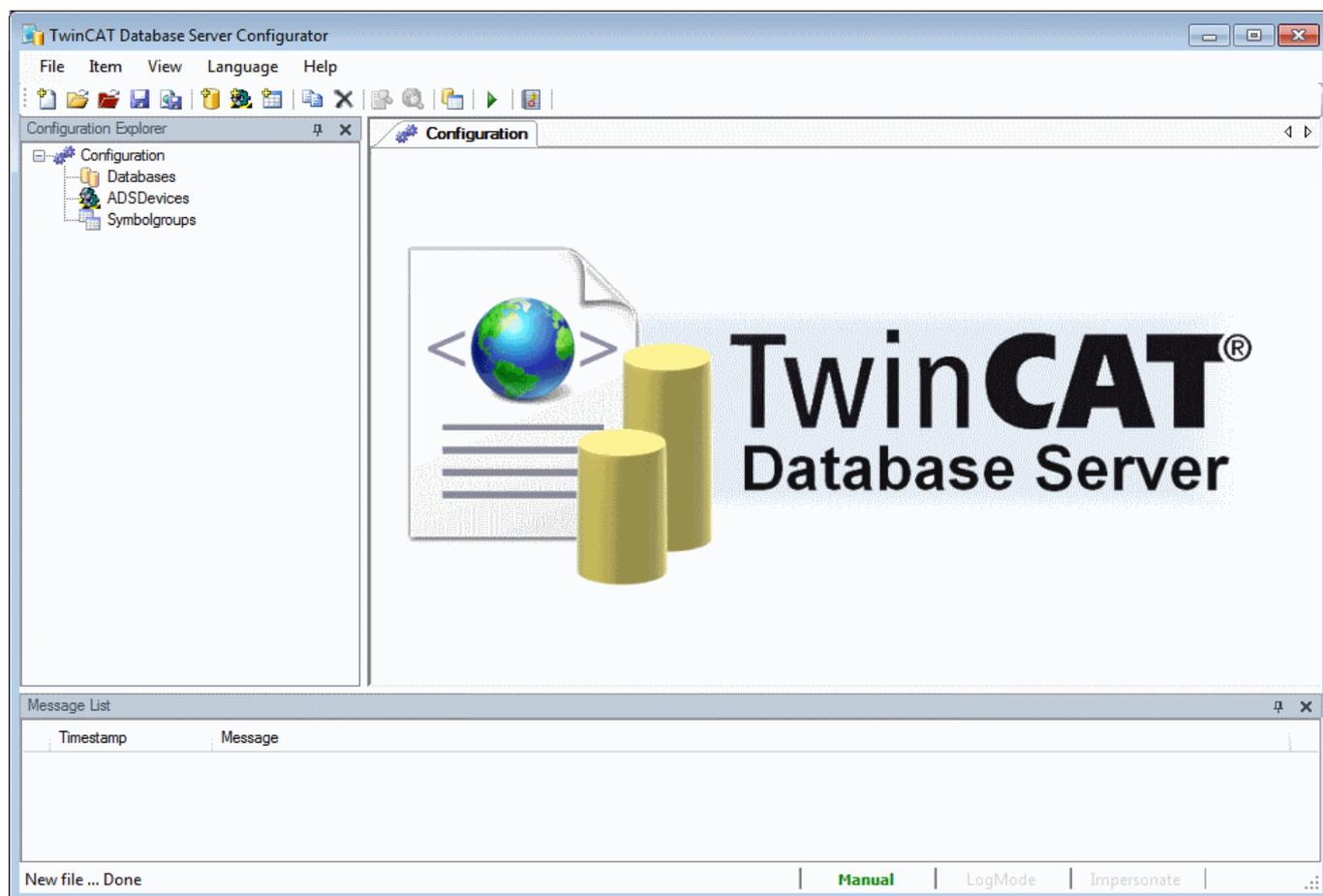
Step by step introduction for hands on with TwinCAT Database Server.

1. Foreword

There are two possibilities to configure the TwinCAT Database Server - on the one hand out of the PLC Control and on the other hand with TwinCAT Database Server XML Configuration File Editor. Today we discuss a small example for the configuration with XML Editor, but there is also an instruction for the second way for testing yourself.

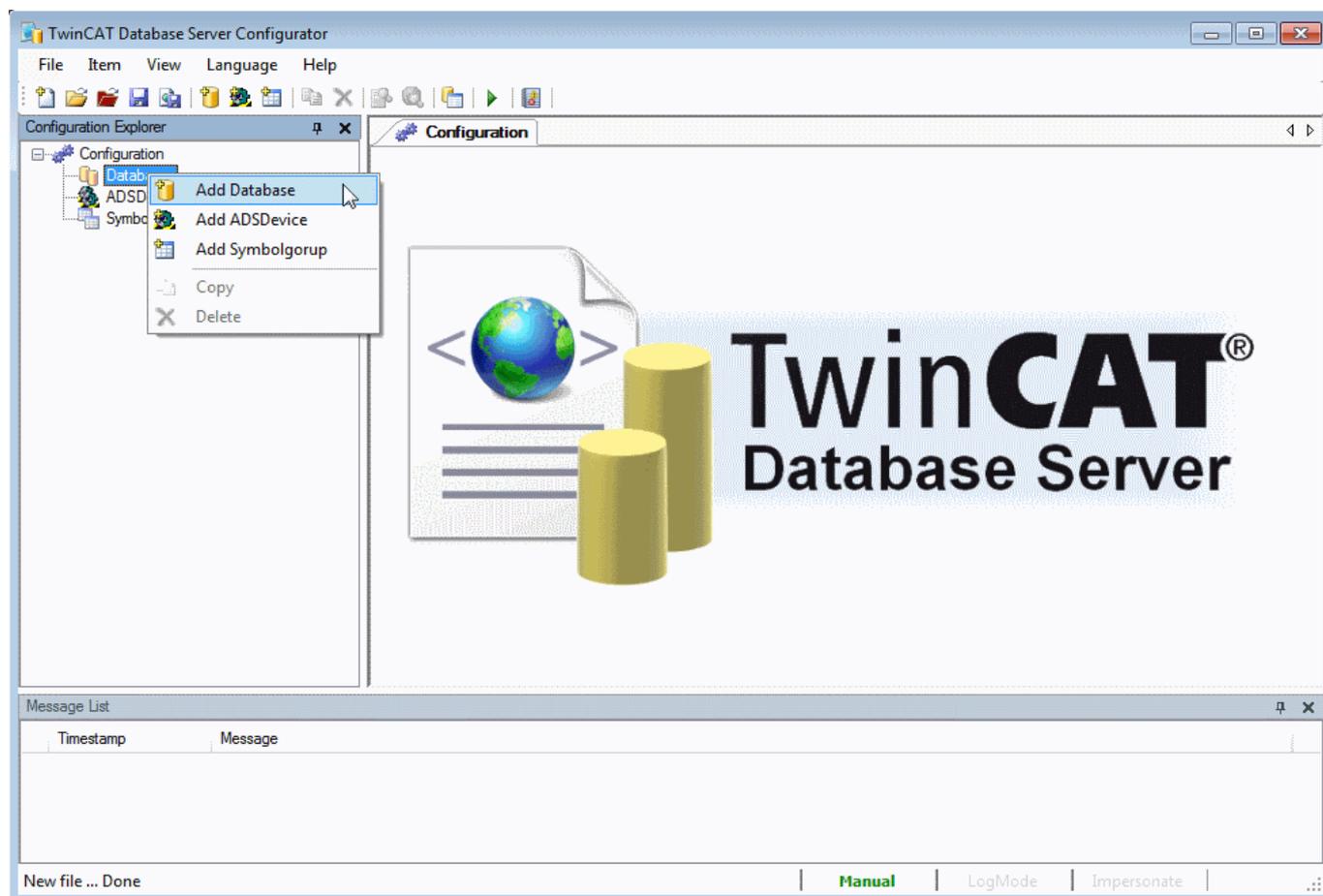
2. DataBaseServer Configuration with XML Editor

Open the Editor under Start -> All Programs -> TwinCAT System -> TwinCAT DataBase Server -> XML Configuration File Editor



BECKHOFF

At first you have to add a new database. Right click on Database as it is shown in the picture or by the symbols in the tool bar.



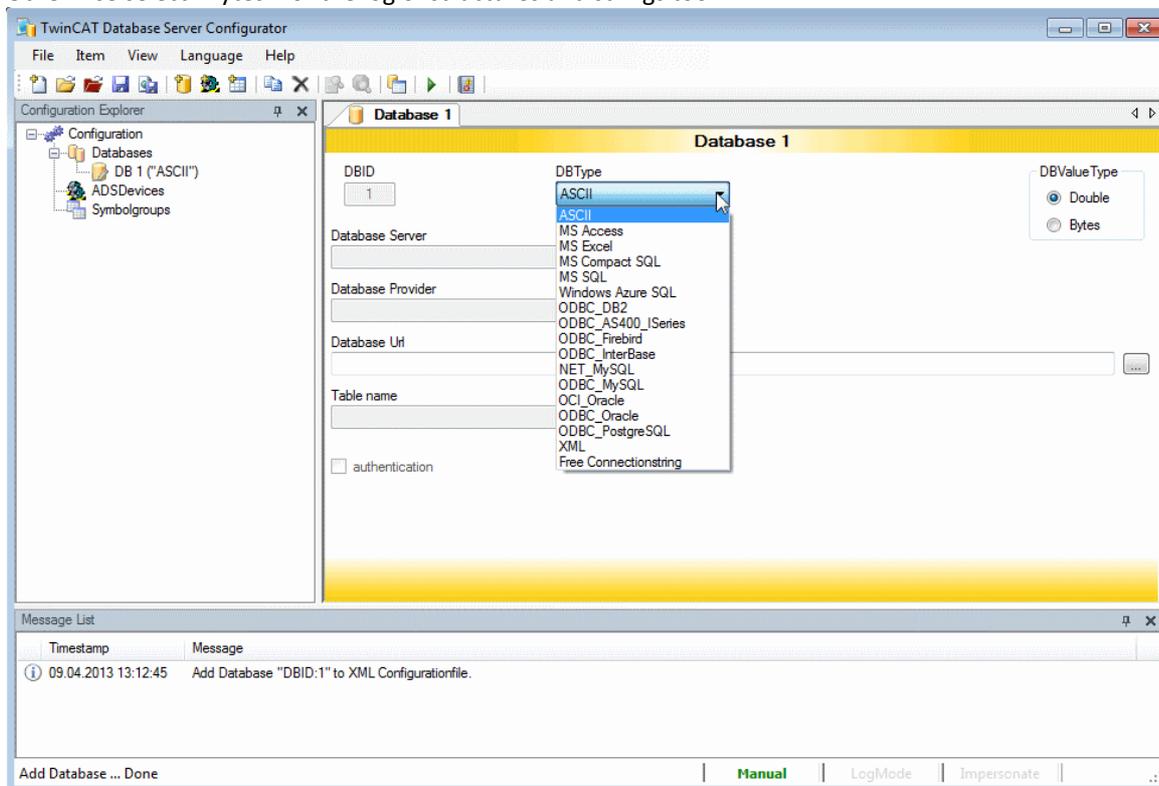
In the next step you must configure your database. Not everybody has got the great databases on his system, that's why we choose the ASCII database.

You can find the declaration of all different database types, which we support, in the Information System.

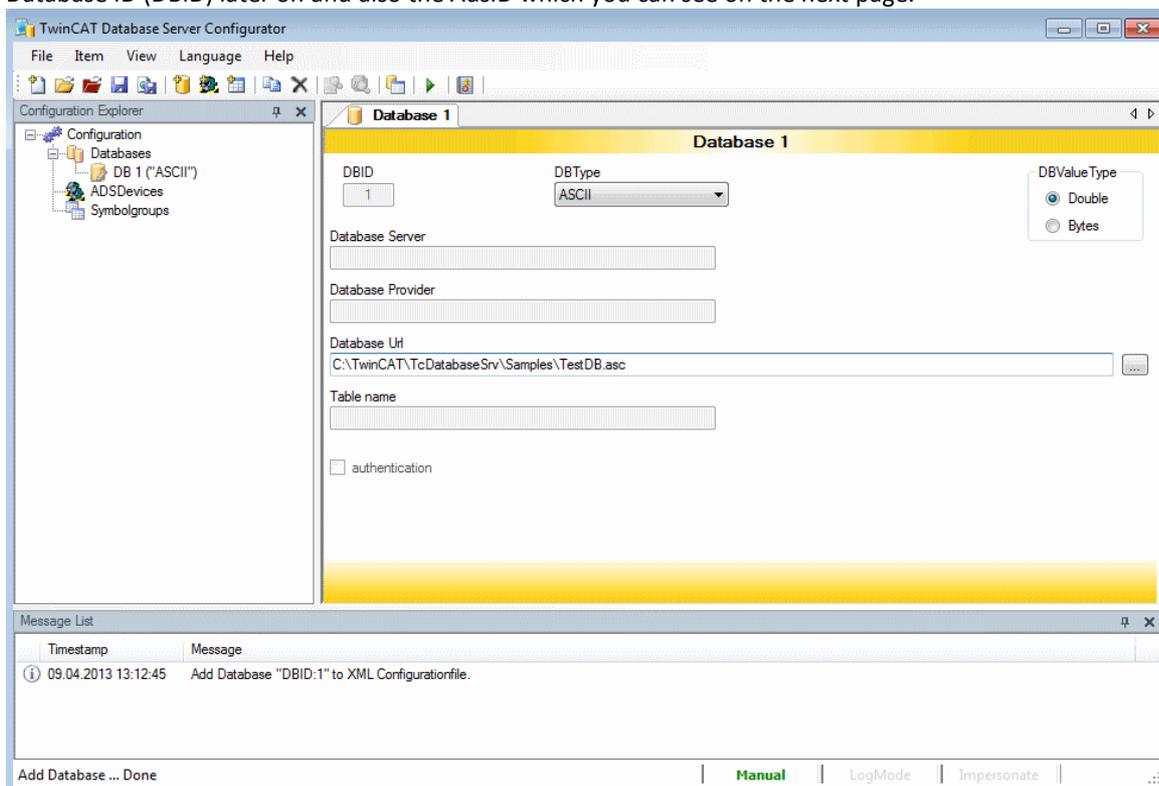
http://infosys.beckhoff.com/content/1031/tcdbserver/html/tcdbserver_dbdeclaration.htm

BECKHOFF

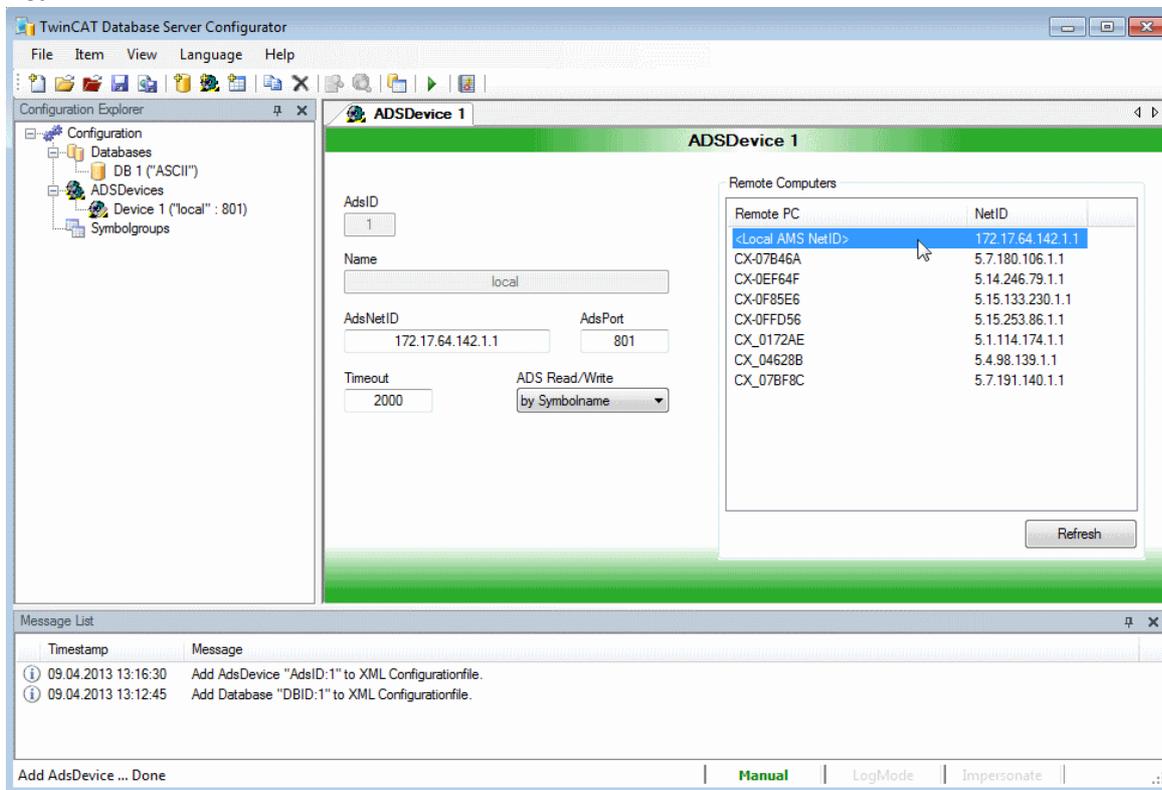
Choose the DBType ASCII. If you only will log alphanumeric data types and Boolean choose "Double" for DBValueType. Otherwise select "Bytes" for the log of structures and strings too.



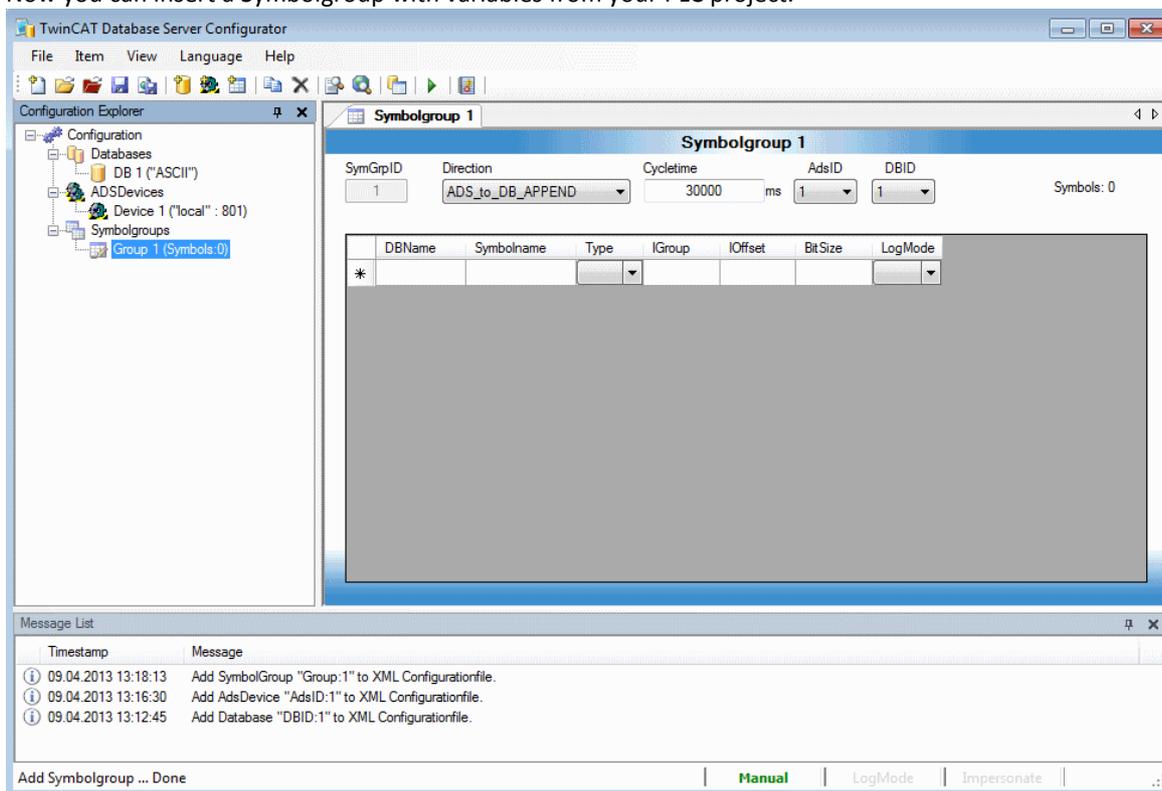
For an ASCII database you must indicate only the database URL, like it is shown in the next picture. You still need the Database ID (DBID) later on and also the AdSID which you can see on the next page.



Add an AdsDevice and insert your AdsNetID, also your AdsPort. If you use the local system, you don't need to type your NetID.

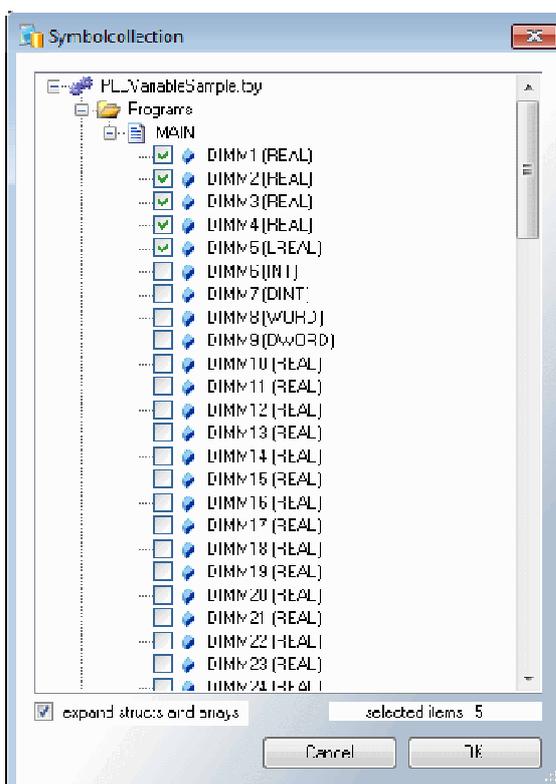
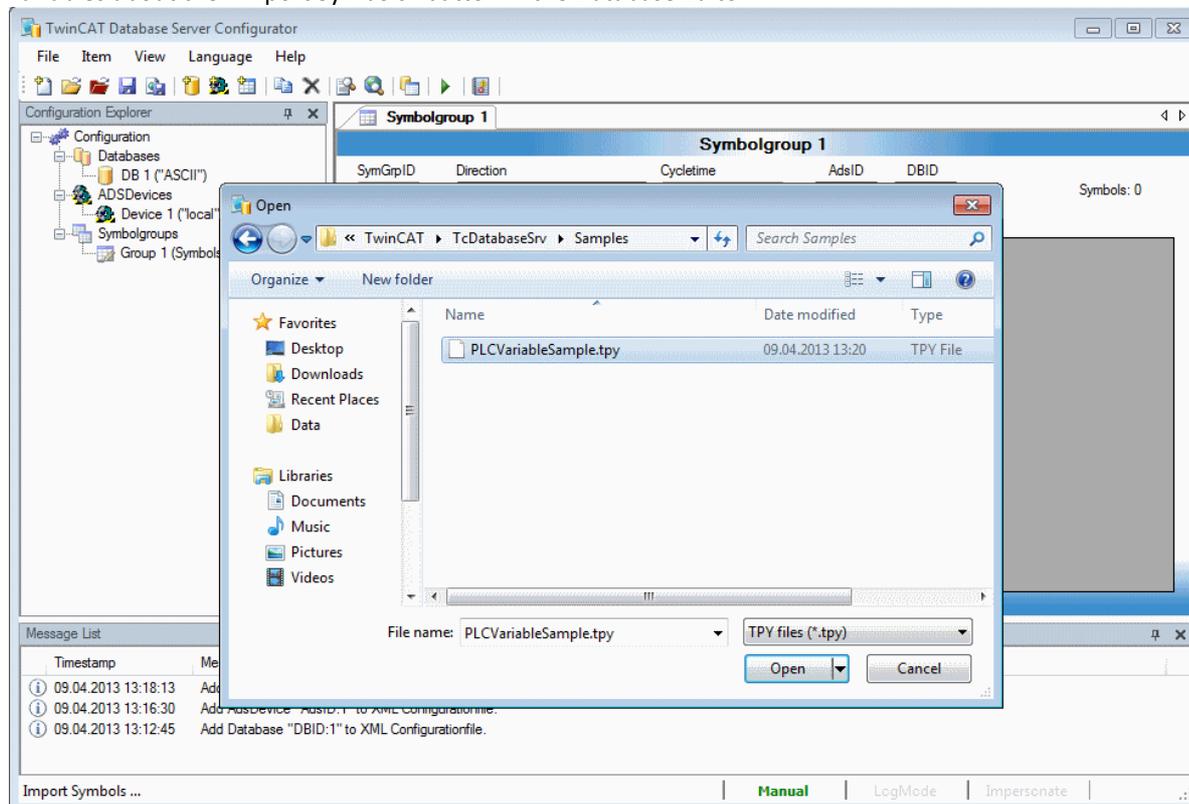


If you create a configuration for a remote system, you have the possibility to choose your target in the table on the right side. Now you can insert a Symbolgroup with variables from your PLC project.



BECKHOFF

Before you configure your Symbolgroup, you have to "Build" your PLC project. Open TwinCAT PLC Control with PLCVariableSample.pro file and then Project -> Build. This is important for the generation of the tpy file. You can search for variables about the "Import Symbols" button in the Database Editor.



Beckhoff Automation GmbH
Eiserstraße 5
33415 Verl
Germany

Postfach 11 42
33398 Verl

Telefon: +49 (0) 52 46/9 63-0
Fax Zentrale: -1 49
Fax Vertrieb: -1 98
Fax Anlagentechnik: -3 79
Fax Service: -4 79
E-Mail: info@beckhoff.de
www.beckhoff.de

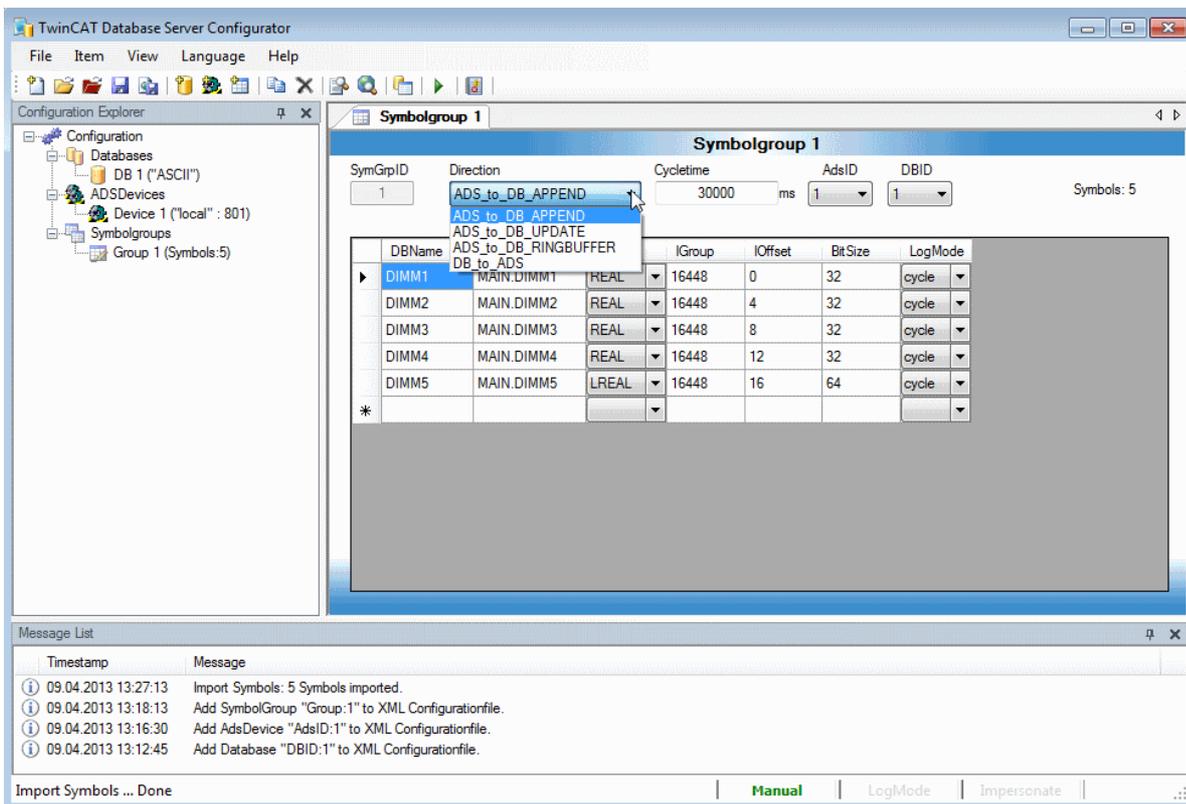
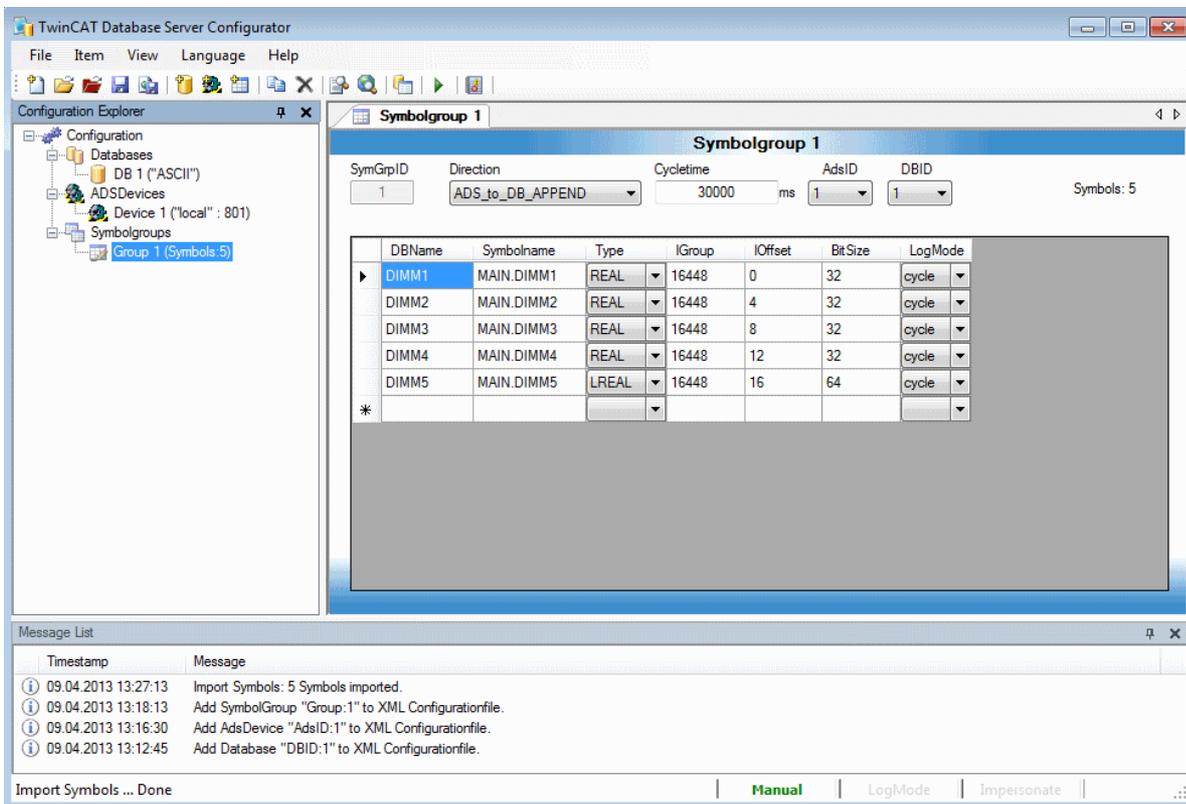
Geschäftsführer:
Dipl. Phys. Hans Beckhoff
Arnold Beckhoff
Registergericht: Gütersloh HRB 1803
Ust.-Id.-Nr.: DE 126787444
Finanzamt Wiedenbrück
St.-Nr. 347/5819/0027

Kreissparkasse Verl
BLZ 478 535 20
Kto.Nr. 4 000 766
SWIFT: WELADED1WDB
IBAN: DE114785352
00004000766

Deutsche Bank
BLZ 480 700 43
Kto.Nr. 371701400
SWIFT: DEUTDE33489
IBAN: DE9348070043
0371701400



For example you choose the first five variables of the Symbolcollection. You can change the LogMode for each variable, like it is shown in the next picture. You have to set the AdsID and the DBID, which you know from the sides before.



BECKHOFF

You have the possibility to choose the communication direction. In this case ADS_to_DB_APPEND is selected.

ADS_to_DB_APPEND:

This option appends the new data entries to the old in the database.

ADS_to_DB_UPDATE:

This option updates the available entries in the database.

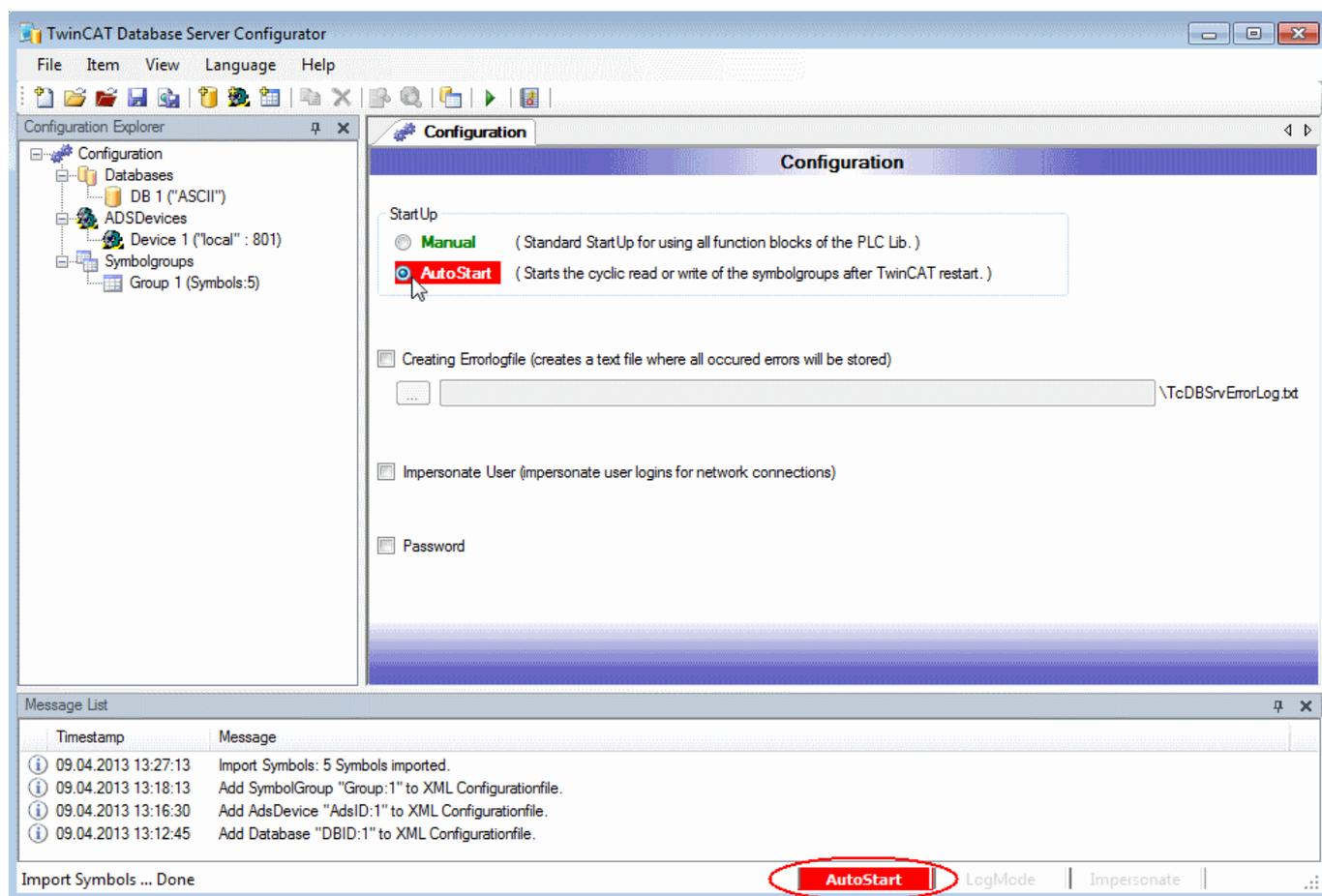
ADS_to_DB_RINGBUFFER:

With this write mode you can limit the count or the age of datasets at databasetables.

DB_to_ADS:

This communication direction describes the reading of values from the database into the PLC.

If the Database Server should be used without the PLC (only logging from PLC variables), the option AutoStart must be set. So the Database Server immediately starts creating the connections to the declared database and ADS-devices after a TwinCAT start. Therefore you have to create a Bootproject from PLCVariableSample.pro. Finally you have to save your DB configuration in the Editor under C:\TwinCAT\Boot.



4. Configuration test

Go to PLC Control and start the PLC, after that create a Bootproject and make a TwinCAT restart. Look for the test database "TestDB" under C:\TwinCAT\TcDatabaseSrv\Samples.

This was a small introduction for the Beckhoff Database Server. Have fun with it. Questions? Then write me an email! For professionals follows a small instruction for the configuration of the Database Server from the PLC.

5. DataBaseServer Configuration from the PLC

Task:

Create a Microsoft SQL Compact database from the PLC Control, add a database connection and insert a new table in your database. Finally use the write function block with the ring buffer mode "RingBuffer_Count" to write 100 times a variable into the table of your database.

At first insert a R_TRIG function block to start your program with a rising edge. After that create a state machine for the call of your database function blocks, like it is shown in the picture below.

```

0001 PROGRAM MAIN
0002 VAR
0003     fbDBCreate           : FB_DBCreate;
0004     fbDBConAdd          : FB_DBConnectionAdd;
0005     fbDBTableCreate     : FB_DBTableCreate;
0006     fbDBWrite           : FB_DBWrite;
0007
0008     state                : INT := 0;
0009     R_Edge                : R_TRIG;
0010     bExecute             : BOOL;
0011 END_VAR
0012
0013 (* Take care for the test with FB_DBCreate that the database do not exist -> for a second run delete the DB *)
0014 R_Edge (CLK := bExecute);
0015 IF R_Edge.Q THEN
0016     state := 1;
0017 END_IF
0018
0019 CASE state OF
0020     0: (* idle state *)
0021     ;
0022     1: (* Create a Database *)
0023     ;
0024     11: (* Is the FB_DBCreate busy? *)
0025     ;
0026     2: (* Add a connection to your Database *)
0027     ;
0028     21: (* Is the FB_DBConnectionAdd busy? *)
0029     ;
0030     3: (* Create a table for your Database *)
0031     ;
0032     31: (* Is the FB_DBTableCreate busy? *)
0033     ;
0034     4: (* Write cyclic values into your Database *)
0035     ;
0036     41: (* Is the FB_DBWrite busy? *)
0037     ;
0038 END_CASE

```

BECKHOFF

Now declare all other variables that you need. Especially the structure of the table. You can find a description how to do this for Microsoft SQL Compact the Information System:

http://infosys.beckhoff.com/english.php?content=content/1033/tcdbserver/html/TcDBServer_MobileDatabase.htm

An AutoID is generated in the "ID" column. The value in this column is always increased by 1. The "Timestamp" column stores the time at which the data record was saved. And the name of the variable is stored in the third column. Under "Value" you find the values of the variables.

You can take the remaining variables from the next picture.

```

0001 PROGRAM MAIN
0002 VAR
0003     fbDBCreate           : FB_DBCreate;
0004     fbDBConAdd          : FB_DBConnectionAdd;
0005     fbDBTableCreate     : FB_DBTableCreate;
0006     fbDBWrite           : FB_DBWrite;
0007
0008     state                : INT := 0;
0009     R_Edge                : R_TRIG;
0010     bExecute             : BOOL;
0011
0012     bError                : BOOL;
0013     uErrID                : UDINT;
0014
0015     uDbID                 : UDINT;           (* Database ID *)
0016     uAdsID                : UDINT := 1;     (* Set your ADS ID *)
0017
0018     (* Table structure: *)
0019     tablestrc: ARRAY [0..3] OF ST_DBColumnCfg :=
0020         (sColumnName := 'ID', sColumnProperty := 'IDENTITY(1,1)', eColumnType := EDBCOLUMN_BIGINT),
0021         (sColumnName := 'Timestamp', eColumnType := EDBCOLUMN_DATETIME),
0022         (sColumnName := 'Name', eColumnType := EDBCOLUMN_NTEXT),
0023         (sColumnName := 'Value', eColumnType := eDBCcolumn_Integer);
0024
0025     iLogVariable           : INT;           (* Log variable *)
0026 END_VAR

```

The next steps.

BECKHOFF

FB_DBCreate:

```

0007 CASE state OF
0008   0:  (* idle state *)
0009   ;
0010
0011   1:  (* Create a Database *)
0012   fbDBCreate( sNetID      := ,
0013               sPathName  := 'C:\TwinCAT\TcDatabaseSrv\Samples',
0014               sDBName    := 'DB_ITW',
0015               eDBType    := eDBType_Mobile_Server,
0016               sSystemDB  := ,
0017               sUserId    := ,
0018               sPassword  := ,
0019               bExecute   := TRUE,
0020               tTimeout   := T#20s,
0021               bBusy     => ,
0022               bError    => bError,
0023               nErrID    => uErrID);
0024   state := 11;
0025
0026   11:
0027   fbDBCreate(bExecute := FALSE);
0028   IF NOT fbDBCreate.bBusy AND NOT fbDBCreate.bError THEN
0029     state := 2;
0030   END_IF

```

In state number one you have to insert the FB_DBCreate. The path of the database must be given to this function block. But you must provide that this database does not exist yet. Otherwise there is an error message. In state 11 you have to wait until the function block is not busy.

FB_DBConnectionAdd:

```

0032 2: (*Add a connection to your Database*)
0033 fbDBConAdd(sNetID      := ,
0034            eDBType     := eDBType_Mobile_Server,
0035            eDBValueType := eDBValue_Double,
0036            sDBServer   := ,
0037            sDBProvider := ,
0038            sDBUrl      := 'C:\TwinCAT\TcDatabaseSrv\Samples\DB_ITW.sdf',
0039            sDBSystemDB := ,
0040            sDBUserId   := ,
0041            sDBPassword := ,
0042            sDBTable    := 'ITW_Table',
0043            bExecute    := TRUE,
0044            tTimeout    := T#20s,
0045            bBusy      => ,
0046            bError     => bError,
0047            nErrID    => uErrID,
0048            hDBID     => );
0049   state := 21;
0050
0051   21:
0052   fbDBConAdd(bExecute:=FALSE, hDBID=> uDbID);
0053   IF NOT fbDBConAdd.bBusy AND NOT fbDBConAdd.bError THEN
0054     state := 3;
0055   END_IF

```

Beckhoff Automation GmbH
Eiserstraße 5
33415 Verl
Germany

Postfach 11 42
33398 Verl

Telefon: +49 (0) 52 46/9 63-0
Fax Zentrale: -149
Fax Vertrieb: -198
Fax Anlagentechnik: -379
Fax Service: -479
E-Mail: info@beckhoff.de
www.beckhoff.de

Geschäftsführer:
Dipl. Phys. Hans Beckhoff
Arnold Beckhoff
Registergericht: Gütersloh HRB 1803
Ust.-Id.-Nr.: DE 126787444
Finanzamt Wiedenbrück
St.-Nr. 347/5819/0027

Kreissparkasse Verl
BLZ 478 535 20
Kto.Nr. 4 000 766
SWIFT: WELADED1WDB
IBAN: DE114785352
00004000766

Deutsche Bank
BLZ 480 700 43
Kto.Nr. 371701400
SWIFT: DEUTDE33489
IBAN: DE9348070043
0371701400

BECKHOFF

In this function block you have to set the DBType and the DBValueType (Bytes or Double). And you already have to give your table a name. The other settings for Microsoft SQL Compact databases are in the Information System:

http://infosys.beckhoff.com/content/1033/tcdbserver/html/tcdbserver_dbdeclaration.htm

FB_DBTableCreate:

```

0055 3:  (* Create a table for your Database *)
0056 fbDBTableCreate(sNetID      := ,
0057                hDBID       := uDbID,
0058                sTableName  := 'ITW_Table',
0059                cbTableCfg  := SIZEOF(tablestrc),
0060                pTableCfg   := ADR(tablestrc),
0061                bExecute    := TRUE,
0062                tTimeout    := #20s,
0063                bBusy       => ,
0064                bError      => bError,
0065                nErrID      => uErrID,
0066                sSQLState   => );
0067 state := 31;
0068
0069 31:
0070 fbDBTableCreate(bExecute := FALSE);
0071 IF NOT fbDBTableCreate.bBusy AND NOT fbDBTableCreate.bError THEN
0072     state := 4;
0073 END_IF

```

One of the inputs of this function block is "hDBID". You give over the value which you get from the FB_DBConnectionAdd. Furthermore you give over the size and the address of your table structure.

FB_DBWrite:

```

0075 4: (*Write cyclic values into your Database *)
0076 fbDBWrite( bExecute := FALSE);
0077 fbDBWrite( sNetID := ,
0078           hDBID := uDbID,
0079           hAdsID := uAdsID,
0080           sVarName := 'MAIN.iLogVariable',
0081           sDBVarName := 'SPEED [km/h]',
0082           eDBWriteMode := eDBWriteMode_RingBuffer_Count,
0083           tRingBufferTime := ,
0084           nRingBufferCount := 100,
0085           bExecute := TRUE,
0086           tTimeout := #20s,
0087           bBusy => ,
0088           bError => bError,
0089           nErrID => uErrID,
0090           sSQLState => );
0091 state := 41;
0092
0093 41:
0094 fbDBWrite(bExecute := FALSE);
0095 IF NOT fbDBWrite.bBusy AND NOT fbDBWrite.bError THEN
0096     state := 4;
0097 END_IF
0098
0099 END_CASE
0100
0101 iLogVariable := iLogVariable + 1;
0102 IF iLogVariable > 2000 THEN
0103     iLogVariable := 0;
0104 END_IF

```

With the FB_DBWrite you can write the current values of the variable “iLogVariable” into the database all the time. But there will be stored only 100 values of the variable, if you set the nRingBufferCount to 100.

At the end of the program code you can see the variable which is increasing by1 every PLC cycle.

Important: If you test the sample program take care that the database does not exist!