Installation Manual

PLCSQL link

SQL Client in a
Siemens S7 PLC

Contents

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH

Introduction

You can operate the PLCSQL link, with different databases.
But in order to get it to work, it is important to setup the Database, to fit with the design for PLCSQL link.

- Have you chosen to operate the PLCSQL link, together with Microsoft SQL server?
  You also need to follow tutorial "MS SQL Installation Manual_2014".

If you still have questions after reading this manual, please send them to info@plcsql.com

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

ALSMATIK

Page 3 of 32

Software requirements.

This example project is based on following software tools:

        PLC Program: Simatic Step 7 V5.5 +SP4 + HF8

        HMI: Simatic WinCC Flexible 2008 SP3 upd7

The HMI is an option, you don't need to use a HMI to get the system running, but it is much easier to look on an HMI rather than on an VAT table.

If you don't have WinCC Flexible, you can request an HMI to use in the TIA Portal.

Please send you request with the version of you TIA portal to info@plcsql.com

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

How does the system work?

We have tried to make a system where it is possible to communicate with a SQL data base without being an "SQL expert", nor being an "PLC expert" regarding communication etc. the system contains a "standard" PLC program, and a "standard" SQL data base that "fits" together.

In the PLC we are using the basic tag types:

Bool.	Is stored in an "Bool" table in the data base.
Int.	Is stored in an "Int" table in the data base.
Dint.	Is stored in an "Dint" table in the data base.
Real	Is stored in an "Real" table in the data base.
String	Is stored in an "String" table in the data base.

To distinguish between the different tags, every tag has a specific number.
In the PLC there is an "Array" that contains all the tags, and in the database the different tables contain the corresponding data types and numbers as in the PLC, so you have complete control with the tags.

Now, somebody will ask, "but we have to connect an existing data base", yes that is no problem, you can easily interconnect between different data bases, so we strongly recommend to "make" a data base exclusively for PLCSQL, so you easily can check where the problem could be when something is not working.

On the following pages there is a schematic view of the layout and the possibilities you have with the PLCSQL system.

Please note the following.

Parameter 10001, 15001, and 30001 is used internally in the "Log" parameters and in the "Recipe" parameters. DON'T write to these parameters.

10001:	SetCount
15001:	SetID
30001:	DateTimeStamp.

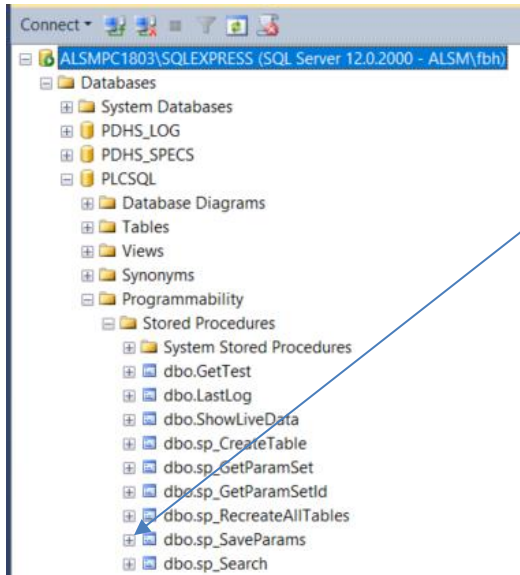| Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

How does PLCSQL work, Log, write to SQL server

In Mssql we want to store the Value 2.3009 in the ParamID[1].
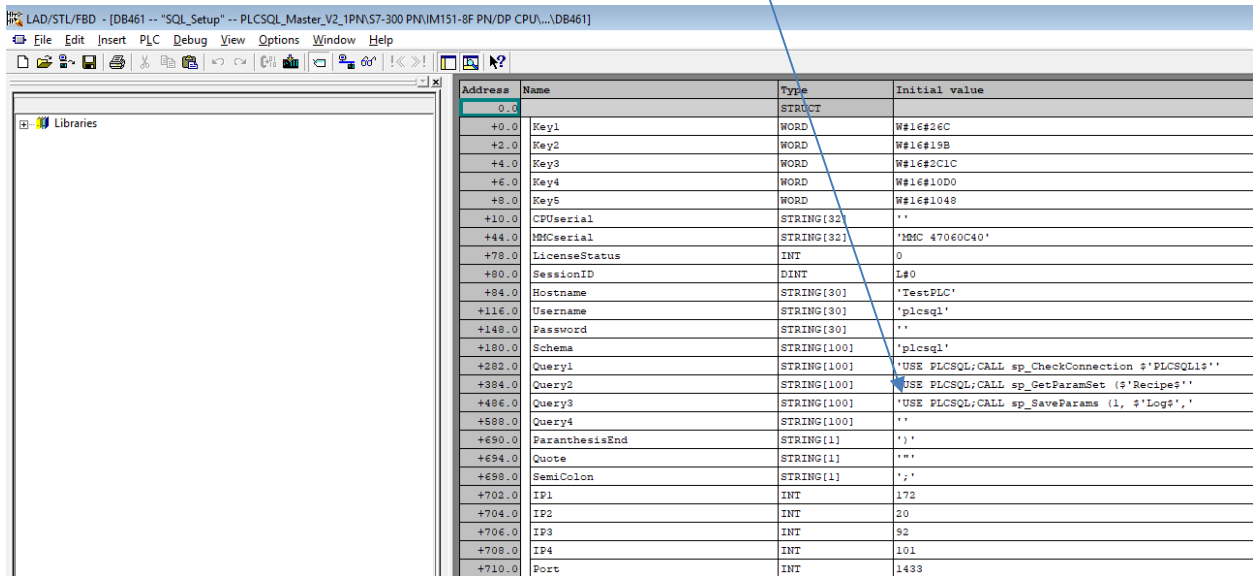
First we need to setup the Call in the PLC for the stored procedure in Mssql.

Open DB461.
In DB461 you setup the Query3 to match the procedure in Mssql:
Query3 = 'CALL sp_SaveParams (1, $'Log$', ';

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH

In the PLC we move the value 2.3009 to the SQL DB
Open FC462 to view example.
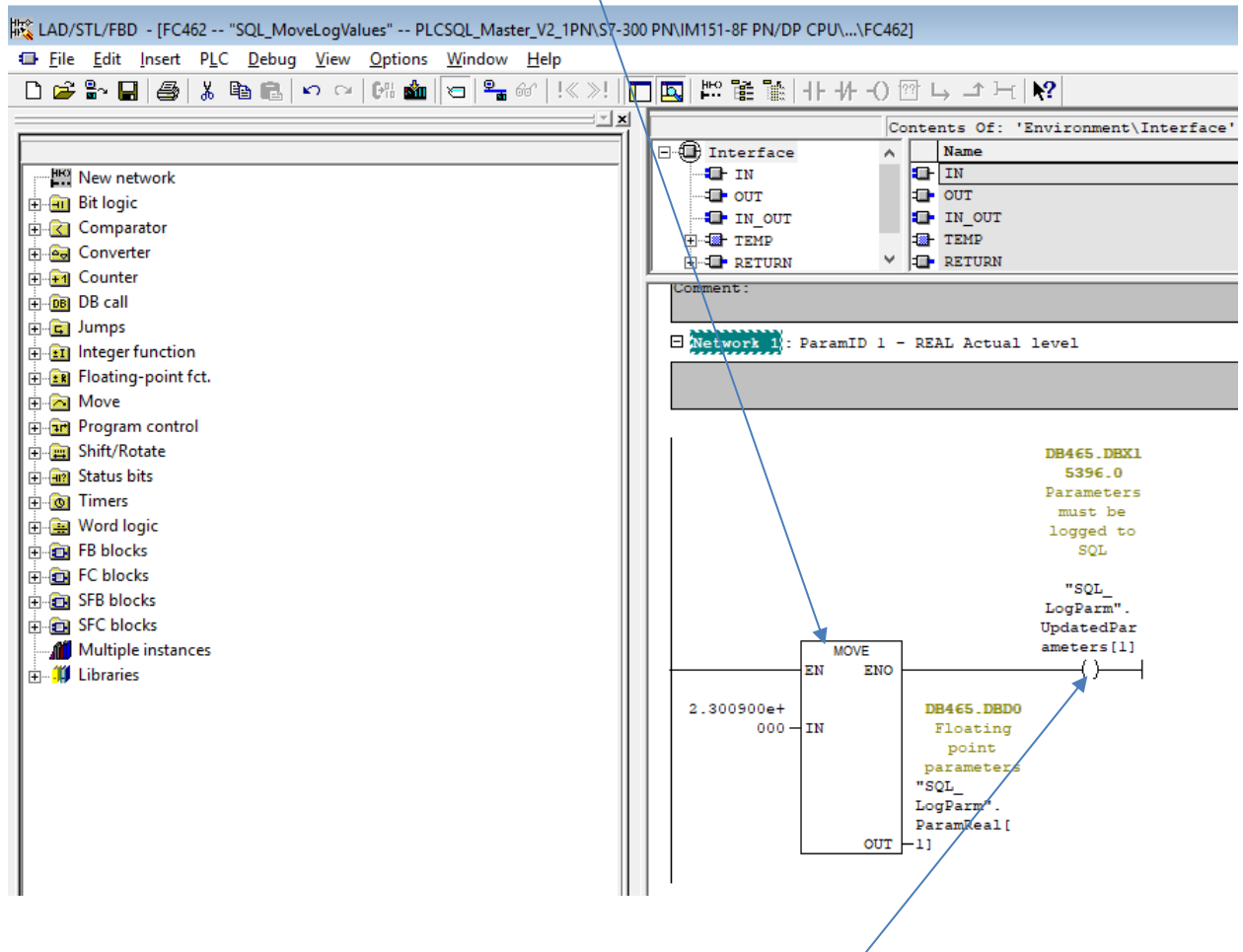
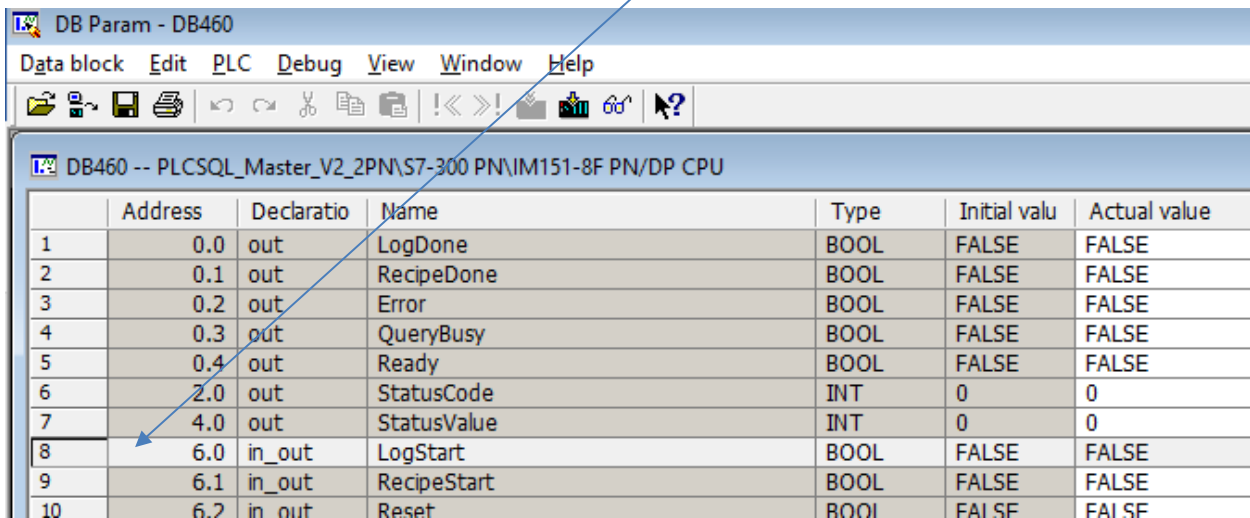And we tell the system that there is an updated value on ParamID[1].

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH

Last we set the bit Sql_Status.LogStart (see DB460) and the value is stored in mssql.

In Mssql we can see that the Value 2.3009 is stored in ParamID[1].

Right click on dbo.Log_REAL and select top 1000 rows to see the view.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

ParamId[1] is defined as a REAL type in both the PLC and Mssql. See DB465.
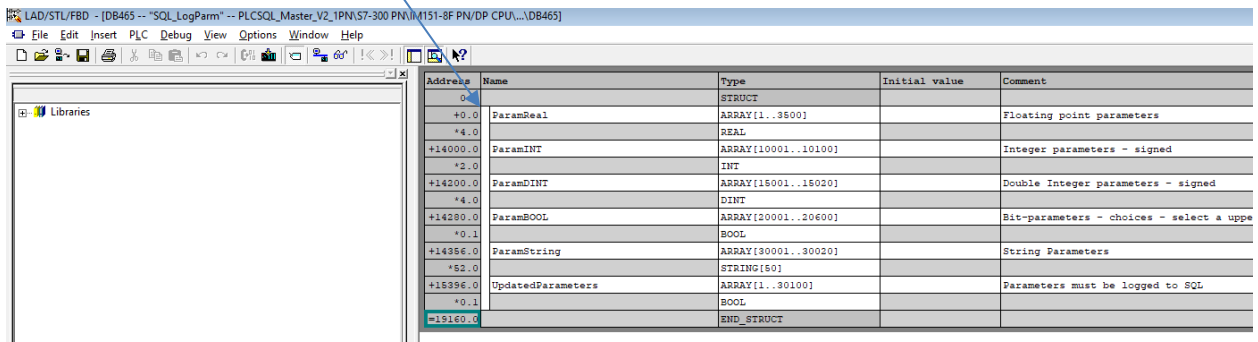
How does PLCSQL work, Recipe, read from SQL server

In Mssql we want to read the stored value in ParamID[3].

First we need to setup the Call in the PLC for the stored procedure in Mssql.
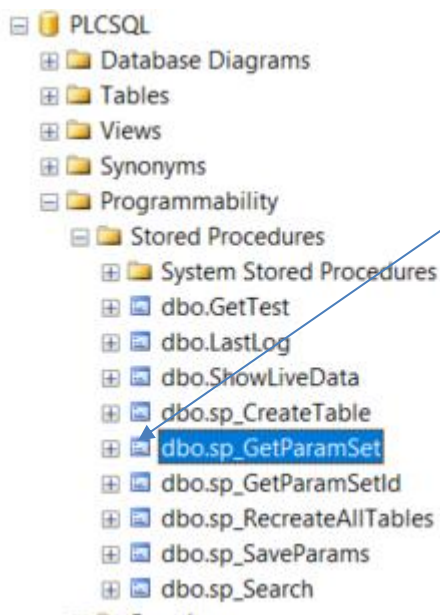
Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Page 9 of 32

Installation Manual: For PLCSQL link

In DB461 you setup the Query2 to match the procedure in Mssql:
Query2 = 'CALL sp_GetParamSet ($'Recipe$',';

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	Key1	WORD	W#16#26C
+2.0	Key2	WORD	W#16#19B
+4.0	Key3	WORD	W#16#2C1C
+6.0	Key4	WORD	W#16#10D0
+8.0	Key5	WORD	W#16#1048
+10.0	CPUserial	STRING[32]	''
+44.0	MMCserial	STRING[32]	'MMC 47060C40'
+78.0	LicenseStatus	INT	0
+80.0	SessionID	DINT	L#0
+84.0	Hostname	STRING[30]	'TestPLC'
+116.0	Username	STRING[30]	'plcsql'
+148.0	Password	STRING[30]	''
+180.0	Schema	STRING[100]	'plcsql'
+282.0	Query1	STRING[100]	'USE PLCSQL;CALL sp_CheckConnection $'PLCSQL1$''
+384.0	Query2	STRING[100]	'USE PLCSQL;CALL sp_GetParamSet ($'Recipe$''
+486.0	Query3	STRING[100]	'USE PLCSQL;CALL sp_SaveParams (1, $'Log$','
+588.0	Query4	STRING[100]	''
+690.0	ParanthesisEnd	STRING[1]	')'
+694.0	Quote	STRING[1]	'"'
+698.0	SemiColon	STRING[1]	';'
+702.0	IP1	INT	172
+704.0	IP2	INT	20
+706.0	IP3	INT	92
+708.0	IP4	INT	101
+710.0	Port	INT	1433

Move the received data from Mssql to your variable in the PLC. Open FC461
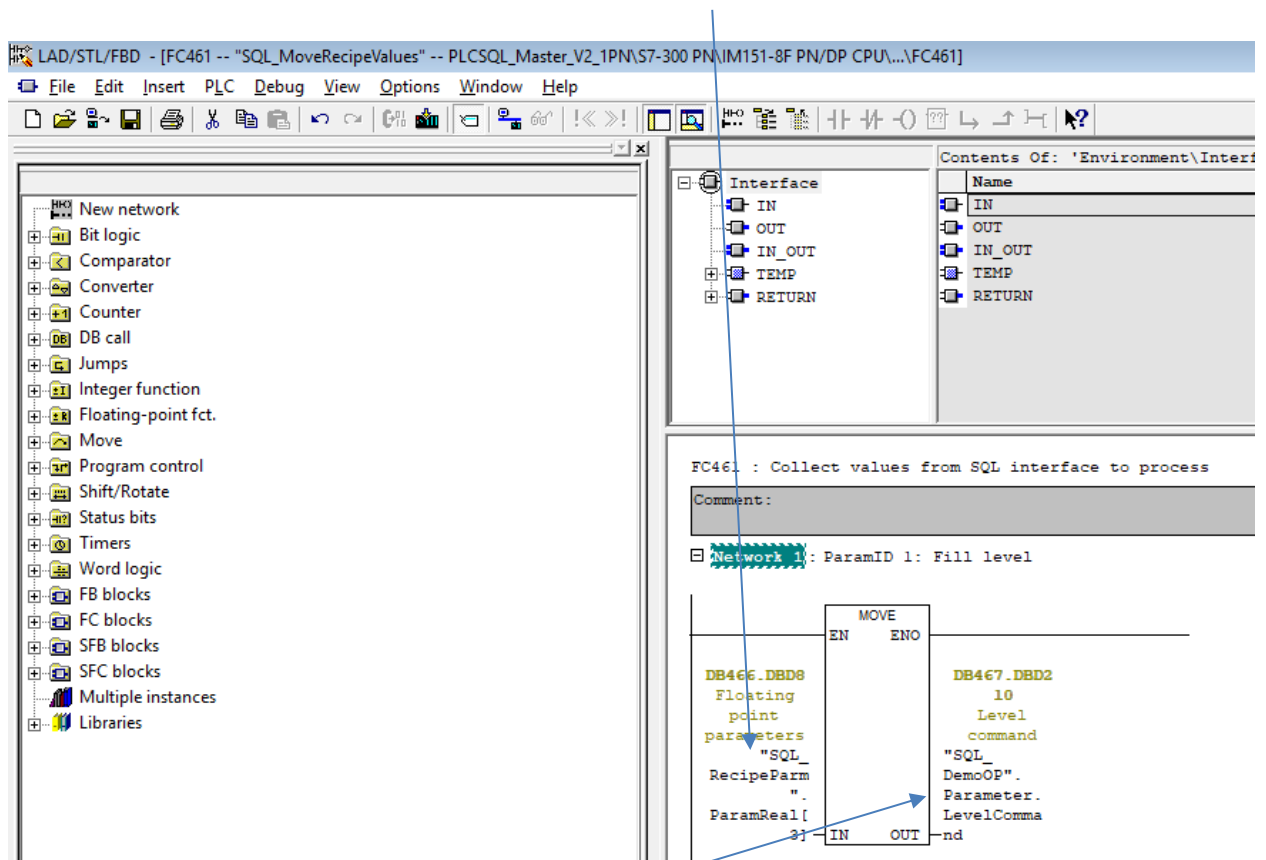
Your local Tag to store the value in.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	ALSMATIK

In the Plc we need to trigger the Sql_Status.RecipeStart, in DB460.

In Mssql we want to read a Real value with ParamID 3 from Recipe

Right click on dbo.Log_REAL and select Top 1000 rows to see the view.

Subject .	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	ALSMATIK

ParamId[3] is defined as a REAL type in both the PLC and Mssql.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Page 12 of 32

Setting up PLCSQL Link

Retrieving the software.

You will receive the software as an "Archive".
You will have to "Retrieve" the software inside from Step 7 v 5.5

If you have told us the type of the PLC when ordering PLCSQL, then, if possible, the correct PLC will be in the hardware configuration.
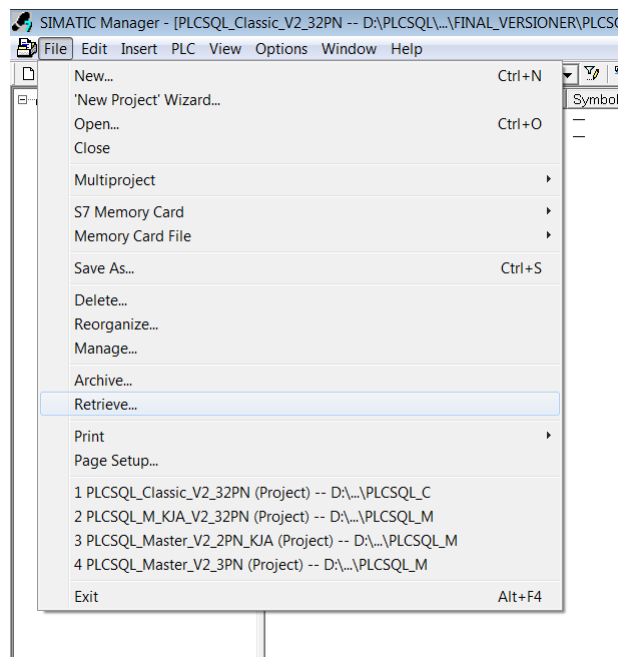
If the selected hardware is not the hardware you are using, you have to make you own project with the correct hardware, and then copy the PLCSQL software to this project.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Setting the PLC IP address.
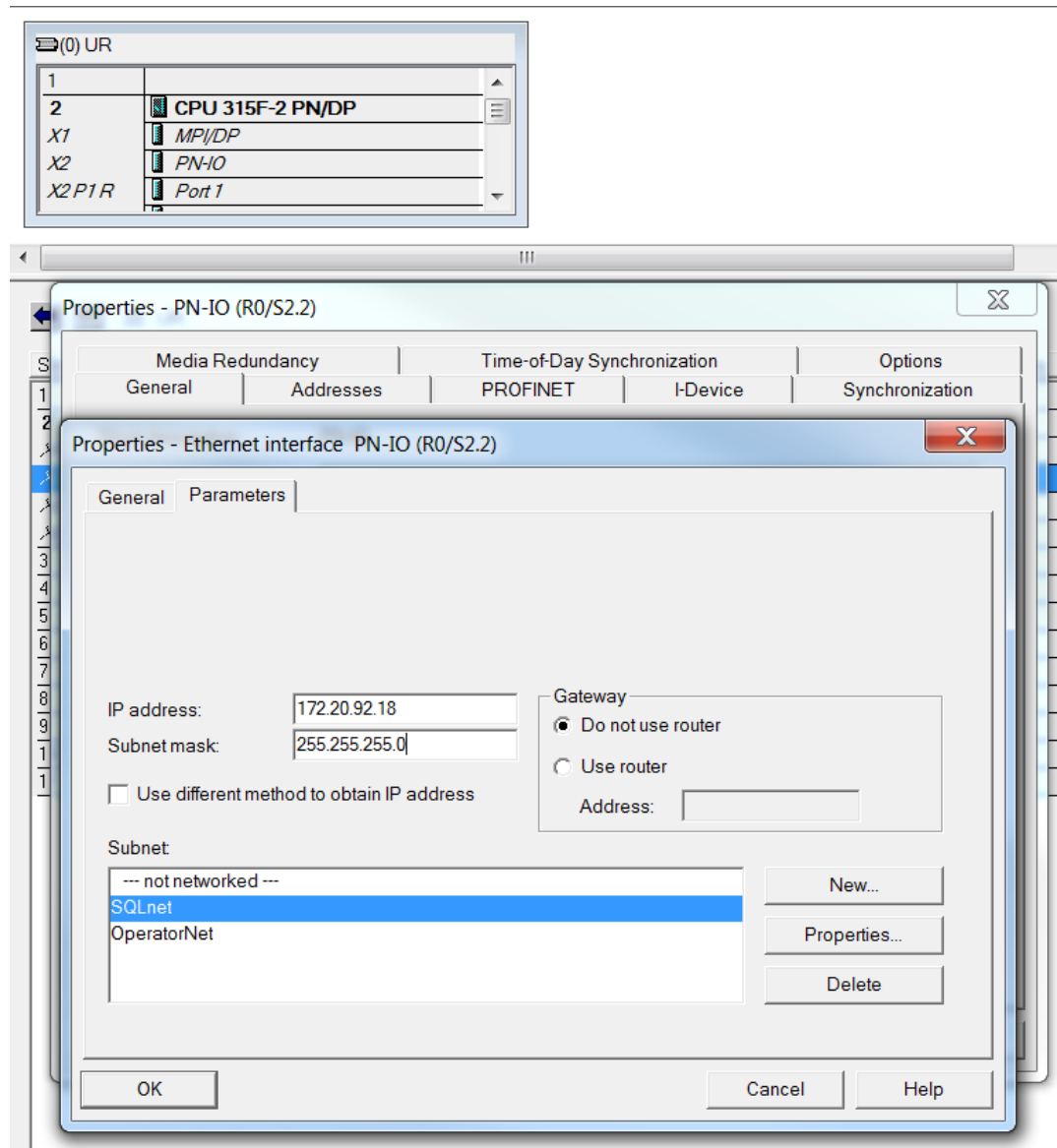
The IP address of the PLC is set in the hardware configuration.
If you need to use a "Router", it is also here you set the router address.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH

Page 14 of 32

Blocks used.

Program in the Step 7 v 5.5 with WinCC Flexible 2008 installed.

Overview of the PLC blocks used by PLCSQL Link.
The main blocks are FB460, FC460, FC461, FC462, DB460 … DB466.
FC460 is "Know how protected", the rest of the blocks are "open".

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Call from OB1.

⊟ **Network 5**: PLCSQL for direct SQL Server connection

```
                                DB460
                              Instance
                               DB for
                               PLCSQL
                                Link
                             controller
                               "SQL_
                               Status"

                                FB460
                              PLCSQL Link
                         Controller handles
                            logon, query
                          generation and
            M2.1          data processing
           "True"         "SQL_Controller"
             ┤ ├──────────┤EN              ENO├──────────────
                    ...─┤LogStart      LogDone├─ ...
                        │RecipeSta    RecipeDon│
                    ...─┤rt                   e├─ ...
                    ...─┤Reset           Error├─ ...
                        │            QueryBusy├─ ...
                        │                Ready├─ ...
                        │            StatusCod│
                        │                    e├─ ...
                        │            StatusVal│
                        │                   ue├─ ...
```
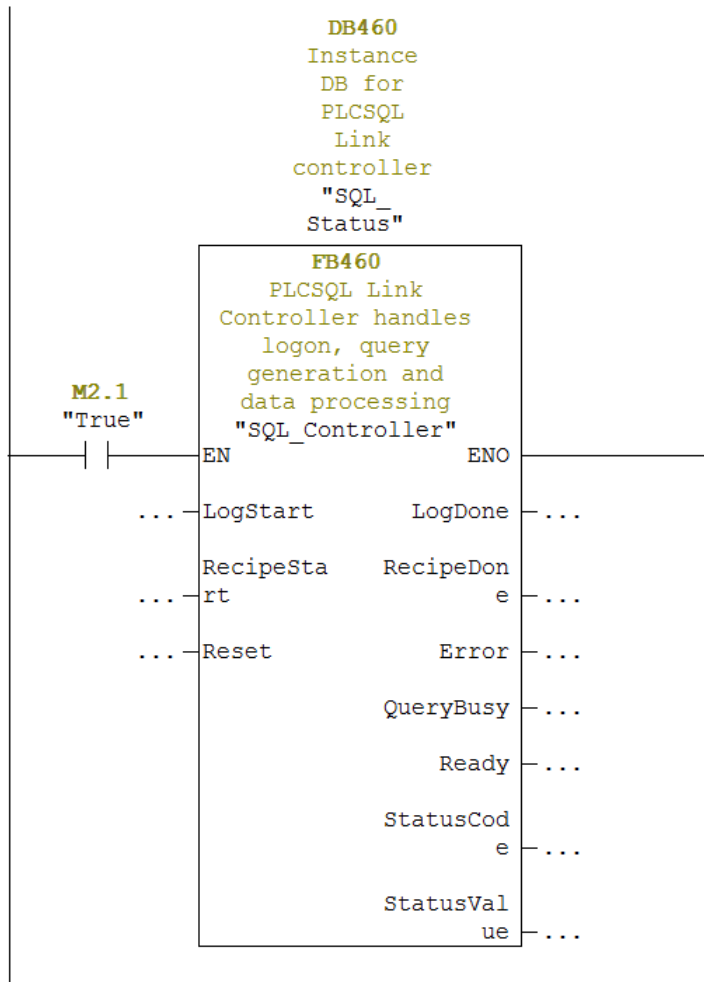
The call of the main block FB460, all other necessary blocks are called inside FB460.

If you define local parameters at "LogStart", RecipeStart", and "Reset", the HMI will no longer work properly with these commands, address instead the parameters directly in "SQL_Status" DB with "Set" command, then you can control the PLCSQL from the PLC code and from the HMI.

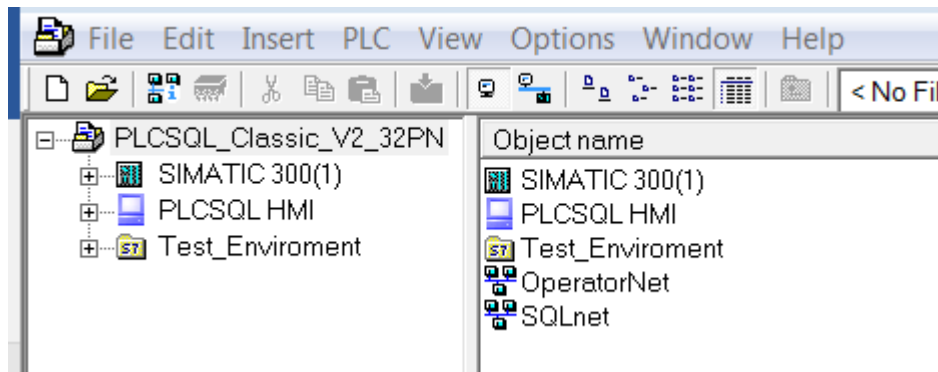| Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH

ALSMATIK

Page 16 of 32

```
⊟ Network 4 : Initialize PLCSQL
```



```
                        FC466
                  "Initialize_PLCSQL"
              EN                    ENO
```
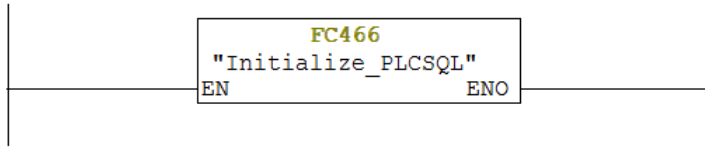
FC466 is not mandatory, you don't need to use this block.
Inside this block you can set the "Licens key", the IP address of the SQL server, and what you else think could be necessary.

System parameters.

There are 3 "system parameters" placed in "SQL_LogParm" and "SQL_RecipeParm" DB's
In both DB's it is the following 3 parameters.

10001:	SetCount	Data type "Int"
15001:	SetID	Data type "Dint"
30001:	DateTimeStamp.	Data type "String"

The "SetCount" parameter 10001 contains the number of parameters excluding parameter 10001,15001, and 30001 in "this" actual "dataset".

The "SetID" parameter 15001, contains the unique number that every "dataset" get when something is stored in the SQL database. The "SetID" number changes only when something is stored in the SQL database, e.g. you trigger the "Log data" function from the PLC or you generate a new "Recipe" in the SQL database. Every time you save a "dataset" in the SQL database, all parameters in this specific "dataset" will get the same "SetID" number, it is the "SetID" number that "connects" all the parameters in this specific "dataset" together.

The "DateTimeStamp" contains the date and time when this "dataset" was stored in the SQL database.

For further explanation, please take a look in section "HMI, SQL Setup" in this manual.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Page 17 of 32

Setup DB.

The following will describe the most important parameters of the setup DB.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	Key1	WORD	W#16#26C
+2.0	Key2	WORD	W#16#1AA
+4.0	Key3	WORD	W#16#2C3E
+6.0	Key4	WORD	W#16#1583
+8.0	Key5	WORD	W#16#116B
+10.0	CPUserial	STRING[32]	''
+44.0	MMCserial	STRING[32]	''
+78.0	LicenseStatus	INT	0
+80.0	SessionID	DINT	L#0
+84.0	Hostname	STRING[30]	'TestPLC'
+116.0	Username	STRING[30]	'plcsql'
+148.0	Password	STRING[30]	'link'
+180.0	Schema	STRING[100]	'plcsql'
+282.0	Query1	STRING[100]	''
+384.0	Query2	STRING[100]	'USE PLCSQL;EXEC sp_GetParamSet '
+486.0	Query3	STRING[100]	'USE PLCSQL;EXEC sp_SaveParams 1, $'Log$', '

Here is shown a section of the "Setup" DB.

You can type the same parameters from the HMI, but every time you download to the PLC, the typed values will be replaced with the "Initial values",

The "Key1….Key5" is the license key that you got to your memory card, or to the PLC serial number. Key1 is the most left number in the license key. Leading "0" is not shown.

"Hostname"	Freely definable.
"Username"	The name of the user in the database, default "plcsql"
"Password"	The password of the user in the database, default "link".
"Schema"	The name of the database, default, "plcsql".
"Query2"	Query to read data to the PLC from the database.
"Query3"	Query to write data from the PLC to the database.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Page 18 of 32

+702.0	IP1	INT	172	
+704.0	IP2	INT	20	
+706.0	IP3	INT	92	
+708.0	IP4	INT	101	
+710.0	Port	INT	1433	
+712.0	DeviceID	BYTE	B#16#2	
+714.0	ConnectionID	INT	1	

"IP1...IP4"   is the IP address of the Server / PC with the database.

"Port"   Is the port trough the database communicates, depends on the SQL server used, Microsoft SQL is default 1433.

"Device ID" Tells the communication blocks witch type of connection to be used, see list below.

B#16#00: CP 443-1EX4x (with S7-400, however, only with connection_type = B#16#12), won't work with SQL
B#16#01: IE interface in interface slot 1 (IF1) with WinAC RTX (TCP only) or IM151 PN
B#16#02: Integrated IE interface with CPUs 315-2 PN/DP and 317-2 PN/DP
B#16#03: Integrated IE interface on the CPU 319-3 PN/DP
B#16#05: Integrated IE interface with CPUs 414-3 PN/DP, 416-3 PN/DP and 416-3F PN/DP
B#16#06: IE interface in interface slot 2 (IF2) with WinAC RTX (TCP only)
B#16#0B. IE interface in interface slot 3 (IF3) with WinAC RTX (TCP only)
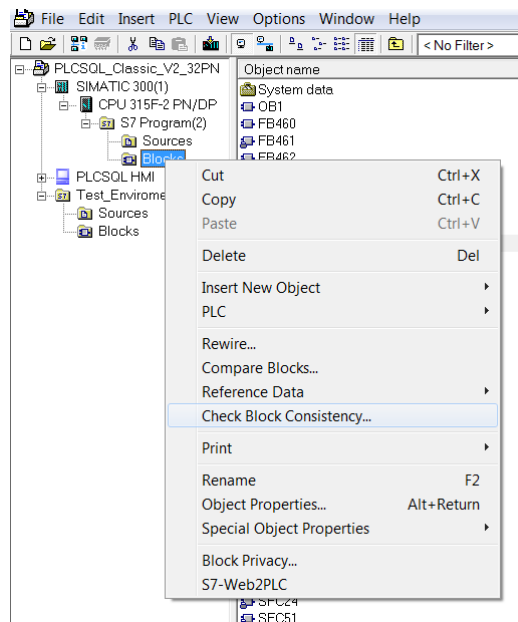B#16#0F: IE interface in interface slot 4 (IF4) with WinAC RTX (TCP only)

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Page 19 of 32

+734.0	LogSizes	STRUCT	
+0.0	MaxUpdatedValues	INT	100
+2.0	MinREAL	INT	1
+4.0	MaxREAL	INT	0
+6.0	MinINT	INT	10001
+8.0	MaxINT	INT	0
+10.0	MinDINT	INT	15001
+12.0	MaxDINT	INT	0
+14.0	MinBOOL	INT	20001
+16.0	MaxBOOL	INT	0
+18.0	MinSTRING	INT	30001
+20.0	MaxSTRING	INT	0
+22.0	StringSize	INT	0
=24.0		END_STRUCT	
+758.0	RecipeSizes	STRUCT	
+0.0	MaxUpdatedValues	INT	100
+2.0	MinREAL	INT	1
+4.0	MaxREAL	INT	0
+6.0	MinINT	INT	10001
+8.0	MaxINT	INT	0
+10.0	MinDINT	INT	15001
+12.0	MaxDINT	INT	0
+14.0	MinBOOL	INT	20001
+16.0	MaxBOOL	INT	0
+18.0	MinSTRING	INT	30001
+20.0	MaxSTRING	INT	0
+22.0	StringSize	INT	0

The sizes for the "Log" parameters and the "Recipe" parameters.
All parameters with "0" is calculated automatically, that is all "Max" parameters.
All "Min" parameters have to have the shown values, don't change.

If / when you change the size of the "SQL_LogParm" or the "SQL_RecipeParm", then remember to run "Check Block Consistency" before you download the program.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH

SQL_LogParm DB.

Address	Name	Type	In
0.0		STRUCT	
+0.0	ParamReal	ARRAY[1..50]	
*4.0		REAL	
+200.0	ParamINT	ARRAY[10001..10050]	
*2.0		INT	
+300.0	ParamDINT	ARRAY[15001..15050]	
*4.0		DINT	
+500.0	ParamBOOL	ARRAY[20001..20160]	
*0.1		BOOL	
+520.0	ParamString	ARRAY[30001..30010]	
*42.0		STRING[40]	
+940.0	UpdatedParameters	ARRAY[1..30999]	
*0.1		BOOL	
=4816.0		END_STRUCT	

Delivery settings for the DB "SQL_LogParm".
REAL   = 50.
INT   = 49, 10001 is used internally.
DINT   = 49, 15001 is used internally.
BOOL   = 160.
STRING   = 9, length 40 char, 30001 is used internally

"UpdatedParamters covers the maximum size of the parameters.

The total size of the DB may not exceed 65534 bytes.

Remember "Check Block Consistency", before download.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

SQL_RecipeParm DB.

Address	Name	Type	I:
0.0		STRUCT	
+0.0	ParamReal	ARRAY[1..50]	
*4.0		REAL	
+200.0	ParamINT	ARRAY[10001..10050]	
*2.0		INT	
+300.0	ParamDINT	ARRAY[15001..15050]	
*4.0		DINT	
+500.0	ParamBOOL	ARRAY[20001..20160]	
*0.1		BOOL	
+520.0	ParamString	ARRAY[30001..30010]	
*42.0		STRING[40]	
+940.0	UpdatedParameters	ARRAY[1..30999]	
*0.1		BOOL	
=4816.0		END_STRUCT	

Delivery settings for the DB "SQL_RecipeParm".
REAL   = 50.
INT   = 49, 10001 is used internally.
DINT   = 49, 15001 is used internally.
BOOL   = 160.
STRING   = 9, length 40 char, 30001 is used internally

"UpdatedParamters covers the maximum size of the parameters.

The total size of the DB may not exceed 65534 bytes.

Remember "Check Block Consistency", before download.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

SQL_Send, SQL_Recieve, SQL_Query DB.

When you change the size of "SQL_Logparm" and "SQL_RecipeParm" the you also have to adjust the size of the following 3 data blocks.

"SQL_Send"   max size 32766 byte.
"SQL_Recieve"   max size 32766 byte.
"SQL_Query"   max size 32766 byte, query data available 32764 bytes.

You can calculate the approx. size of the DB's with help of the following table.

The size of 1 variable is approx. as follows (average sizes on the "safe" side)
REAL   20 bytes, (can be bigger with large numbers)
INT   12 bytes
DINT   12 bytes, (can be bigger with large numbers)
BOOL   9 bytes
STRING   6 bytes + number of chars.

Typical it is the size of the "SQL_Query" that sets the maximum number of variables to "handle" in 1 telegram (Query).
It is possible to log 500 Real, 500 Int, 500 Dint, 160 Bool, and 400 Strings 20 Char long in one "shot", the query length for these about 2000 variables is about 32000 bytes.

The PLC may go to "Stop" (default) if any of the DB's is to short.

Remember "Check Block Consistency", before download.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

HMI, SQL Setup

Preconditions, you have set up the database with our standard script.

In the "SQL Setup" picture, you have to do the following.

1: Type the "SQL client license", and press "Reset SQL client".
2: Select the correct PLC type in "DeviceID PN controller".
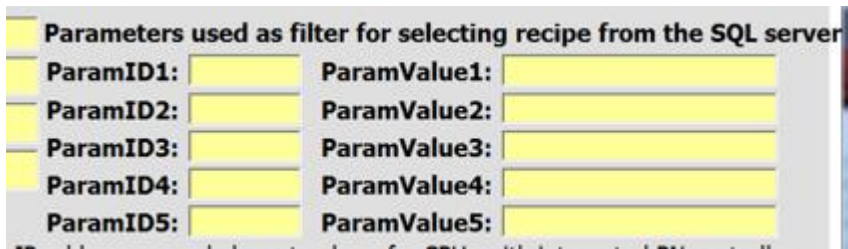3: Type the "SQL Server IP"

If you not used the standard for
"SQL Username"   (plcsql)
"SQL Password"   (link)
"SQL Database"   (plcsql)
Then you also have to change these settings.

Subject .	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

If you read the manual for the SQL server installation, then you maybe know that it is the "SetID" parameter that is the "key" that "connects" the different parameters together.

Every time you trigger a "Log", or "insert" a new "Recipe", the actual parameters you are handling at this moment will get a unique "SetID" number.

I will try to explain how it works, take a look at the following table. "NU" means "No use" nothing is saved in the particular parameter.

	Value	Value	Value	Value	Value
SetID	1	2	3	4	5
P[10]	2.2	0.8	1.0	NU	NU
P[10100]	-10	-10	8	NU	-10
P[15020]	123456789	NU	NU	NU	14445
P[20009]	True	False	NU	NU	NU
P[30005]	NU	'link'	NU	'Empty'	'PLCSQL'
SetCount	4	4	2	1	3
DATETIME	12.00	12.02	12.08	12.10	12.15

In the table, we have "SetID" 1 to 5, the "values" that belongs to the single "SetID" is placed below the "SetID" and has the same color.

If we take a look at the parameters with "SetID" = 1 then we have a
REAL = 2.2
INT = -10
DINT = 123456789
BOOL = True
SetCount = 4
DATETIME = 12.00

If you now write "10" in "ParamID1:" and "2.2" in "ParamValue1:" and trigger a "Read", then you will get the values we just have looked at, placed in the corresponding parameters in the "SQL_RecipeParm" DB.

If we now try "ParamID1:" = 10100, "ParamValue1:" = -10, then we will get the values with "SetID" = 5, why? we have 3 sets with the value "-10", because "SetID" 5 has the "newest" "Time stamp".

You can use up to 5 different parameters as "filter" for you "search" in the "Recipe" tables, the parameters are combined with the "AND" logic.
If you just trigger a "Read", the you will get the newest dataset.
If there is no data or wrong "filter", you will get an error, and the system will restart.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

HMI, SQL Status

PLCSQL Link	SQL Status	

SQL Client

Request	Reset	Get data from SQL	Log data to SQL	Get busy	Log busy

SQL Client Sta	232	Log Data was succesfully accepted and saved in SQL server. Ready for new request

Last Param	10001	**RetVal**	0000	0000	**Offset:**	192	**LogSetID:**	76
DB Numb	0	**Size**	0		**Length:**	0	**RecipeSetID:**	0

Controller Step Nr:	230	230 - Query Request was Successfully executed

SQL info text:	

TCP Network communication

Connection Status:	Connection established	Login OK	Ready	No error

Last Connect Status:	0000	0000 - Connection was established successfully
Last Send Status:	0000	0000 - Send job completed without error
Last Receive Status:	7002	7002 - Follow-on call, receive job being processed. Note: during this processing the FB writes data to the r

SQL Data

Query:	**Length:**	203	Byte (maximum 32764)

Current time	1/25/2017 12:49:31 PM	**Duration**	0	milliseconds
Last log request	1/25/2017 12:49:22 PM	**Duration**	52	milliseconds
Last recipe request	1/1/1990 12:00:00 AM	**Duration**	0	milliseconds
Last reset time	1/25/2017 12:40:40 PM			
Last error time	1/25/2017 12:40:08 PM			
Retry counter	0			

No.	Time	Date	Status	Text	GR

SQL Setup	SQL Status	SQL DB Sizes	SQL Test	Exit Runtime

If you press the button "Log data to SQL" you should get a view like above
(First time you press the button, you will properly get an error, wait until the "Timeout" is gone, and then try again.)

Subject .	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH

ALSMATIK

Page 26 of 32

| PLCSQL Link | | SQL Status | | | | |

SQL Client

Request		Reset	Get data from SQL	Log data to SQL	Get busy	Log busy
SQL Client Sta	3106	SQL_ReadDataStream - No data found in database for the current request				
Last Param	30001	**RetVal** 0000 0000	**Offset:** 260	**LogSetID:** 77		
DB Numb	0	**Size** 0	**Length:** 0	**RecipeSetID:** 0		
Controller Step Nr:	220	220 - Query Request Send completed - waiting for reply from server				
SQL info text:						

TCP Network communication

Connection Status:	Connection established	Login OK	Not ready	Error
Last Connect Status:	0000	0000 - Connection was established successfully		
Last Send Status:	0000	0000 - Send job completed without error		
Last Receive Status:	7002	7002 - Follow-on call, receive job being processed. Note: during this processing the FB writes data to the r		

SQL Data

Query:	**Length:**	45	**Byte (maximum 32764)**		
Current time	1/25/2017 1:14:57 PM	**Duration**	3536	milliseconds	
Last log request	1/25/2017 1:13:38 PM	**Duration**	54	milliseconds	
Last recipe request	1/25/2017 1:14:53 PM	**Duration**	3536	milliseconds	
Last reset time	1/25/2017 1:14:36 PM				
Last error time	1/25/2017 1:14:53 PM				
Retry counter	1				

No.	Time	Date	Status	Text		GR

| SQL Setup | SQL Status | SQL DB Sizes | | SQL Test | | Exit Runtime |

If you press the button "Get data from SQL" you should get a view like above, because there is no "recipe data" in the database

When the "Time before TimeOut" is gone, the PLCSQL system will be reset and log in to the database again.

SQL request queries

Query 1 (request data): USE PLCSQL;EXEC sp GetParamSet

Query 2 (log data): USE PLCSQL;EXEC sp SaveParams 1, 'Log',

SQL request queries

Query 1 (request data): USE PLCSQL;EXEC sp GetParamSet

Query 2 (log data): USE PLCSQL;EXEC sp SaveParams 1, 'Recipe',

If you change the "Query 2", to use the "Recipe" table instead of the "Log" table.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Page 27 of 32

PLCSQL Link   **SQL Status**

SQL Client

| Request | Reset | Get data from SQL | Log data to SQL | Get busy | Log busy |

SQL Client Sta   231   Recipe/Setpoint Data from SQL server values was succesfully transferred to user program. Ready for new requ

Last Param 30008   **RetVal** 0000 0000   **Offset:** 375   **LogSetID:** 1

DB Numb 0   **Size** 0   **Length:** 0   **RecipeSetID:** 1

Controller Step Nr: 230   230 - Query Request was Successfully executed

SQL info text:

TCP Network communication

Connection Status: | Connection established | Login OK | Ready | No error |

Last Connect Status: 0000   0000 - Connection was established successfully

Last Send Status: 0000   0000 - Send job completed without error

Last Receive Status: 7002   7002 - Follow-on call, receive job being processed. Note: during this processing the FB writes data to the r

SQL Data

Query: **Length:** 45   **Byte (maximum 32764)**

Current time	1/25/2017 1:23:57 PM	**Duration**	0	milliseconds
Last log request	1/25/2017 1:23:42 PM	**Duration**	319	milliseconds
Last recipe request	1/25/2017 1:23:48 PM	**Duration**	17	milliseconds
Last reset time	1/25/2017 1:15:13 PM			
Last error time	1/25/2017 1:14:53 PM			
Retry counter	0			

No.	Time	Date	Status	Text	GR

| SQL Setup | SQL Status | SQL DB Sizes | | SQL Test | | Exit Runtime |

Then you will be able to "Log" and "Get" data from the database.

| Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

HMI, SQL DB sizes

Default settings of the data blocks by delivery.

Here you get an overview of the DB's number and sizes.

Remember to run "Check block consistency" before download to the PLC.

Subject .	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH

Page 29 of 32

HMI, SQL test.
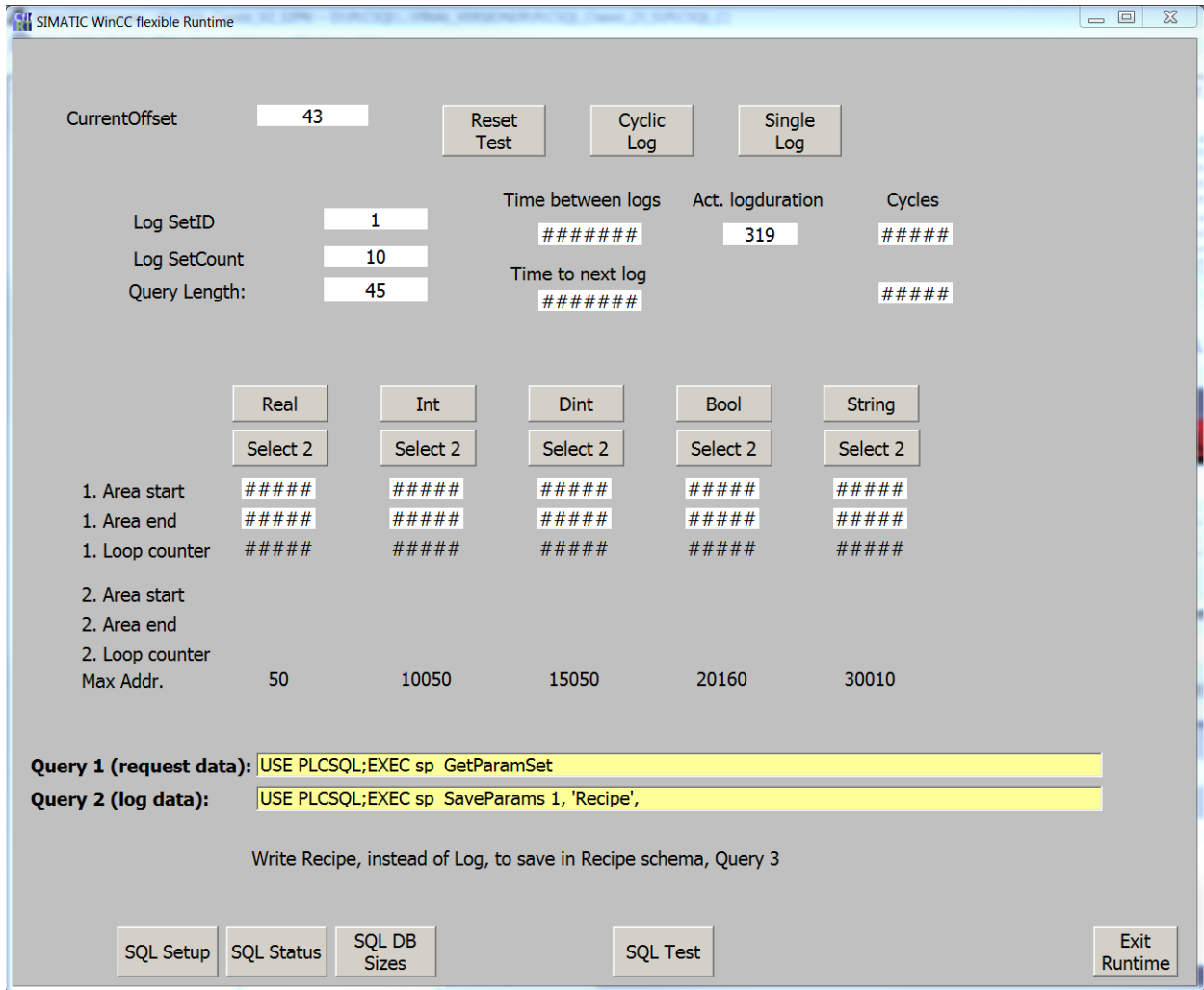
This page is intended for test purposes, if you don't need the picture you can delete it. The code for the test program is not placed in the main program, but in an "S7" program container.

You call FB28 in OB1, and FB29 in FC462
The source code for the blocks are also available in the "Sources" folder.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	ALSMATIK
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Changing the block numbers.

	Statu	Symbol	Address	Data type	Comment
1		BLKMOV	SFC 20	SFC 20	Copy variables
2		Debug	DB 470	DB 470	
3		DI_STRNG	FC 464	FC 464	Double Integer to String
4		False	M 2.0	BOOL	
5		FILL	SFC 21	SFC 21	Initialize a Memory Area
6		I_STRNG	FC 465	FC 465	Integer to String
7		Initialize_PLCSQL	FC 466	FC 466	
8		R_STRNG	FC 463	FC 463	Real To String
9		RD_REC	SFC 59	SFC 59	Read a Data Record
10		RDREC	SFB 52	SFB 52	Read a Process Data Record
11		RDSYSST	SFC 51	SFC 51	
12		READ_CLK	SFC 1	SFC 1	Read System Clock
13		RecipeStr1	VAT 2		
14		SQL_Client	FC 460	FC 460	PLCSQL Client - handles TCP communication to SQL Server
15		SQL_Controller	FB 460	FB 460	PLCSQL Link Controller handles logon, query generation and data processing
16		SQL_DemoOP	DB 467	DB 467	Demonstration of implementing process values - can be deleted
17		SQL_LogParm	DB 465	DB 465	Process values to be logged to SQL
18		SQL_MoveLogVal...	FC 462	FC 462	Move log values to SQL db
19		SQL_MoveRecipe...	FC 461	FC 461	Test parameters from SQL db
20		SQL_Query	DB 464	DB 464	SQL Request query to server
21		SQL_Receive	DB 463	DB 463	TCP Receive buffer for SQL network communication
22		SQL_RecipeParm	DB 466	DB 466	Recipe/setpoint/order values from SQL
23		SQL_Send	DB 462	DB 462	TCP Send buffer for SQL network communication
24		SQL_Setup	DB 461	DB 461	Settings for SQL connection
25		SQL_Status	DB 460	FB 460	Instance DB for PLCSQL Link controller
26		SQL_Test	FC 1	FC 1	
27		String1	VAT 1		
28		String2	VAT 3		
29		TCON	FB 463	FB 463	Connect
30		TDISCON	FB 464	FB 464	Disconnect
31		Test_Data	DB 468	DB 468	
32		TEST_DB	SFC 24	SFC 24	Test Data Block
33		TON	SFB 4	SFB 4	Timer On Delay
34		TP	SFB 3	SFB 3	Generate a Pulse
35		TRCV	FB 462	FB 462	Receive Data
36		True	M 2.1	BOOL	
37		TSEND	FB 461	FB 461	Send Data
38					

In the symbol table, just change the NUMBERS of the block's you want to give another number, save the symbol table, and run the "famous" "Check block consistency", before you download to the PLC.

Subject.	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx	
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH	

Page 31 of 32

Empty page

Subject .	PLC program. For PLCSQL	Document:	PLCSQL PLC Inst Classic Manual_V1_04.docx
Ref.	MIP Version 1.04	Revision:	2019-03-12 by FBH