Documentation PLCcom interface for SQL databases

Version 2.0

Indi.Systems GmbH

Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de

Tel + 49 421-989703-30
Fax + 49 421-989703-39

Table of contents

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

What is the PLCcom interface for SQL databases?

The SQL database interface is provided by the PLCcom-Library in conjunction with the PLCcom-Dataserver.

With this interface the user can access following functionalities:

1. Data logging
   Log all data from devices connected to the database.

2. Data image
   A complete image of all variables with current values and timestamp of last change will be saved into one file per datasource.

For example, an external application can access data (historic data as well) of the software in use via this interface.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Possible use cases for the filesystem interface

There are countless possible applications where you could use this interface. You can always use the interface if you need to access the PLCcom system from external applications.

Furthermore you can use the provided data for external analysis, charts, reports, or just for archiving purposes.

This way, an external application could access and analyze data from PLCcom.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

What do I need to use the SQL interface?

To use the PLCcom SQL interface you'll need an up-to-date customer-provided and administrated database instance.

Depending on the amount of data as well as the storage period it's also possible to use a free of charge database system like Oracle XE or SQL Server Express.

The database must be accessible by TCP/IP. If necessary configure your firewall so that PLCcom can communicate with the database. The network connection between PLCcom and the database should be at least capable to transmit 100Mbit/s. Lower speeds may lead to performance issues.

Please note:
PLCcom doesn't provide any database server or licenses. PLCcom just allows connecting to an existing database. Needed databases and/or licenses must be provided by the customer/user.

Also note:
The archiving processes are highly performant and were tested with standard hardware and a data volume of 1000 variable changes every 100ms without any problems.

Should however, for some reason, the PLCcom system provide more data than the archiving process can handle, it will buffer data into the main memory. The process will work through these records one by one. This will only happen until a threshold of 50.000 records is reached. After this, no new records will be saved, but discarded instead.

This problem may, for example, occur in otherwise used hardware or too little resources provided for the archiving process (e.g. virtual server).

For this reason, we recommend checking the load of the system before going into productive use. You should also provide enough free disk space. The needed space highly depends on the number of variables as well as the update interval. For this reason it's difficult to give an accurate recommendation, but 20GB should be a good start.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

How to prepare the interface?

Before getting started you must provide a set of database objects for the PLCcom library. For this reason we provided you with several SQL examples for different database systems in the subfolder **Scripts**.

These objects must be created in the schema corresponding to the parameterized connection of the QuickHMI project. You may need administrator rights to execute these scripts. Please make sure that all scripts finish without any failures. All database objects must be accessible by the interface.

The following objects must exist:

- PLCCOM_DATAIMAGE table
  - Primary key PK_PLCCOM_DATAIMAGE
  - Index IDX_PLCCOM_DATAIMAGE_DATASO_NAME
- PLCCOM_DATALOG table
  - Primary key PK_PLCCOM_DATALOG
  - Index IX_PLCCOM_DATALOG_DATASOURCE
  - Index IX_PLCCOM_DATALOG_TS

Depending on the database type you may have to create trigger and sequences to auto increment ID fields. Details can be found in the enclosed SQL scripts.

The used database connection needs rights to insert, update and delete records in the created tables. If these aren't granted the interface process can't work properly.

Especially delete permissions should be granted. Without this rights the process can't reorganize old records and therefore free disk space. This can lead to your system running out of space, and even crash your operating system. The same applies to other processes blocking the database or single tables.

Please check the remaining disk space continually, even during operation.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Configure the interface

You can make the following settings for the interface:

1. Handing over a connection to the target database
2. Enable or disable archiving
3. Enable or disable the creation of a complete variable image

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Which SQL databases are supported by PLCcom?

The PLCcom library supports basically all SQL connections inheriting from the base types **System.Data.Common.DbConnection** (Dotnet) or **java.sql.connection** (Java).

PLCcom itself doesn't provide any database provider. Needed references for database providers must be added to the project by the developer himself. The initialized database connector then must be handed over to PLCcom. To avoid any deadlock situation you need to provide a single database connection for every archiving and image functionality you want to use.

PLCcom was tested with the following databases:

- Oracle
- MS SQL-Server
- MySQL
- PostgresSQL
- Firebird
- SQLite
- ODBC-Connection (only DotNet, not tested with Java)
- OLE-Connection (only DotNet, not tested with Java)

If you're using a file based database system (like SQLite) please consider the following:

Some file based databases lock the used file as soon as a connection is established. In this case both given connections must work on two different databases.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Code examples for setting up a PLCcom-Dataserver with logging functionality

CSharp

```
PLCcom.PLCComDataServer.PLCComDataServer myDataServer = null;
PLCcomDevice Device = null;

//Enter your Username + Serial here
Console.WriteLine("Please enter your user name");
authentication.User = Console.ReadLine();
Console.WriteLine("Please enter your user serial key");
authentication.Serial = Console.ReadLine();

Console.WriteLine("Start Connect to TCPIP device...");
//Create an PLCcom-Device instance
Device = new TCP_ISO_Device("192.168.1.100", 0, 2, ePLCType.S7_300_400_compatibel);

//set autoconnect to true and idle time till disconnect to 10000 milliseconds
Device.setAutoConnect(true, 10000);

//Create an instance depending on the device type
Console.WriteLine("Create DataServer PLCDataServerTCP1...");
myDataServer = new PLCcom.PLCComDataServer.PLCComDataServer_TCP("PLCDataServerTCP1", (TCP_ISO_Device)Device, 500);

//register incoming events
//register events
myDataServer.OnConnectionStateChange += new
PLCcom.PLCComDataServer.PLCComDataServer.ConnectionStateChangeEventHandler(myDataServer_OnConnectionStateChange);
myDataServer.OnReadDataResultChange += new
PLCcom.PLCComDataServer.PLCComDataServer.ReadDataResultChangeEventHandler(myDataServer_OnReadDataResultChange);
myDataServer.OnIncomingLogEntry += new
PLCcom.PLCComDataServer.PLCComDataServer.OnIncomingLogEntryDelegate(myDataServer_OnIncomingLogEntry);

//define new request
Console.WriteLine("Create new Request Read 4 Bytes from DB1 at address 0 ...");
ReadDataRequest RequestItem1 = new ReadDataRequest(eRegion.DataBlock,          //Region
                                                   1,                          //datablock
                                                   0,                          //startAdress
                                                   eDataType.BYTE,             //target data type
                                                   4);                         //Quantity

//add new request to plccom data server
myDataServer.AddReadDataRequest(RequestItem1, "Request1");

//define new request
Console.WriteLine("Create new Request Read 10 DWord from Flags_Markers at address 4 ...");
ReadDataRequest RequestItem2 = new ReadDataRequest(eRegion.Flags_Markers,      //Region
                                                   0,                          //datablock
                                                   4,                          //startAdress
                                                   eDataType.DWORD,            //target data type
                                                   4);                         //Quantity

//add new request to plccom data server
myDataServer.AddReadDataRequest(RequestItem2, "Request2");


//add on or more Loggingkonnektoren with logging and writing of a data image into filesystem or database
//in this case create a new DatabaseConnector instance
//Create two valid MS SQL Connections and refer it to new database connector
//A valid database connection for writing 'Log-Archive' based of a DbConnection object. If the value is null, the flag
'isWriteLogActive' will be disabled
System.Data.SqlClient.SqlConnection SQLConLogArchive = new System.Data.SqlClient.SqlConnection(<valid Connectionstring>);
//A valid database connection for writing 'Image-Archive' based of a DbConnection object. If the value is null, the flag
'isWriteImageActive' will be disabled
System.Data.SqlClient.SqlConnection SQLConImageArchive = new System.Data.SqlClient.SqlConnection(<valid Connectionstring>);

//Create LoggingConnector instance
PLCcom.ExternalLogging.LoggingConnector con = new PLCcom.ExternalLogging.DataBaseConnector(SQLConLogArchive,
                                              SQLConImageArchive,
                                              "Connector1",                //unique connector name
                                              true,                        //activate progressive logging
                                              true);                       //activate image writing
//add Connector to Dataserver
myDataServer.AddOrReplaceLoggingConnector(con);

//start PLCcom data server
Console.ReadLine();
```

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Visual Basic

```
Dim myDataServer As PLCcom.PLCComDataServer.PLCComDataServer = Nothing
Dim Device As PLCcomDevice = Nothing
'Enter your Username + Serial here
Console.WriteLine("Please enter your user name")
authentication.User = Console.ReadLine()
Console.WriteLine("Please enter your user serial key")
authentication.Serial = Console.ReadLine()

Console.WriteLine("Start Connect to TCPIP device...")
'Create an PLCcom-Device instance
Device = New TCP_ISO_Device("192.168.1.100", 0, 2, ePLCType.S7_300_400_compatibel)

'set autoconnect to true and idle time till disconnect to 10000 milliseconds
Device.setAutoConnect(True, 10000)

'Create an instance depending on the device type
Console.WriteLine("Create DataServer PLCDataServerTCP1...")
myDataServer = New PLCcom.PLCComDataServer.PLCComDataServer_TCP("DataServerTCP1", DirectCast(Device, TCP_ISO_Device), 500)

'register incoming events
AddHandler myDataServer.OnConnectionStateChange, AddressOf myDataServer_OnConnectionStateChange
AddHandler myDataServer.OnReadDataResultChange, AddressOf myDataServer_OnReadDataResultChange
AddHandler myDataServer.OnIncomingLogEntry, AddressOf myDataServer_OnIncomingLogEntry

Console.WriteLine("Create new Request Read 4 Bytes from DB1 at address 0 ...")
'Parameter:
'-Region
'-datablock
'-startAdress
'-target data type
'-Quantity
Dim RequestItem1 As New ReadDataRequest(eRegion.DataBlock, 1, 0, eDataType.[BYTE], 4)
'add new request to plccom data server
myDataServer.AddReadDataRequest(RequestItem1, "Request1")

Console.WriteLine("Create new Request Read 10 DWord from Flags_Markers at address 4 ...")
'Parameter:
'-Region
'-datablock
'-startAdress
'-target data type
'-Quantity
Dim RequestItem2 As New ReadDataRequest(eRegion.Flags_Markers, 0, 4, eDataType.DWORD, 4)
'add new request to plccom data server
myDataServer.AddReadDataRequest(RequestItem2, "Request2")

'add on or more Loggingkonnektoren with logging and writing of a data image into filesystem or database
'in this case create a new DatabaseConnector instance
'Create two valid MS SQL Connections and refer it to new database connector
'A valid database connection for writing 'Log-Archive' based of a DbConnection object. If the value is null, the flag
'isWriteLogActive' will be disabled
Dim SQLConLogArchive As New System.Data.SqlClient.SqlConnection(<valid Connectionstring>)
'A valid database connection for writing 'Image-Archive' based of a DbConnection object. If the value is null, the flag
'isWriteImageActive' will be disabled
Dim SQLConImageArchive As New System.Data.SqlClient.SqlConnection(<valid Connectionstring>)

'create LoggingConnector instance
Dim con As PLCcom.ExternalLogging.LoggingConnector = New PLCcom.ExternalLogging.DataBaseConnector(SQLConLogArchive, _
                                                     SQLConImageArchive, _
                                                     "Connector1", _
                                                     true, _
                                                     true)
'add Connector to Dataserver
myDataServer.AddOrReplaceLoggingConnector(con)

'start PLCcom data server
myDataServer.StartServer()
Console.ReadLine()
'stop PLCcom data server
myDataServer.StopServer()
```

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Java

```
PLCComDataServer myDataServer = null;
PLCcomDevice Device = null;

BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
//Enter your Username + Serial here
System.out.println("Please enter your user name");
authentication.User(input.readLine());
System.out.println("Please enter your user serial key");
authentication.Serial(input.readLine());

System.out.println("Start Connect to TCPIP device...");
//Create an PLCcom-Device instance
Device = new TCP_ISO_Device("192.168.1.100", 0, 2, ePLCType.S7_300_400_compatibel);

//set autoconnect to true and idle time till disconnect to 10000 milliseconds
Device.setAutoConnect(true, 10000);

//Create an instance depending on the device type
System.out.println("Create DataServer PLCDataServerTCP1...");
myDataServer = new PLCComDataServer_TCP("PLCDataServerTCP1", (TCP_ISO_Device)Device, 500);

//register incoming events
// register Connection state change event
myDataServer.connectionStateChangeNotifier = new ConnectionStateChangeNotifier(this);
// register incoming log event
myDataServer.incomingLogEntryEventNotifier = new IncomingLogEntryEventNotifier(this);
// register change ReadDataResult event
myDataServer.readDataResultChangeNotifier = new ReadDataResultChangeNotifier(this);

//define new request
System.out.println("Create new Request Read 4 Bytes from DB1 at address 0 ...");
ReadDataRequest RequestItem1 = new ReadDataRequest(eRegion.DataBlock,          //Region
                                                   1,                          //datablock
                                                   0,                          //startAdress
                                                   eDataType.BYTE,             //target data type
                                                   4);                         //Quantity

//add new request to plccom data server
myDataServer.addReadDataRequest(RequestItem1, "Request1");

//define new request
System.out.println("Create new Request Read 10 DWord from Flags_Markers at address 4 ...");
ReadDataRequest RequestItem2 = new ReadDataRequest(eRegion.Flags_Markers,      //Region
                                                   0,                          //datablock
                                                   4,                          //startAdress
                                                   eDataType.DWORD,            //target data type
                                                   4);                         //Quantity

//add new request to plccom data server
myDataServer.addReadDataRequest(RequestItem2, "Request2");

//add on or more Loggingkonnektoren with logging and writing of a data image into filesystem or database
//in this case create a new DatabaseConnector instance
//create two valid JDBC Connections and refer it to new database connector. If the value is null, the flag 'isWriteLogActive' will be
disabled
Connection SQLConLogArchive = DriverManager.getConnection(<valid Connectionstring>);
Connection SQLConImageArchive = DriverManager.getConnection(<valid Connectionstring>);

LoggingConnector  con = new DataBaseConnector(SQLConLogArchive,
                                              SQLConImageArchive,
                                              "Connector1",                //unique connector name
                                              true,                        //activate progressive logging
                                              true);                       //activate image writing

((DataBaseConnector) con).setTableName_DATAIMAGE("PLCCOM_DATAIMAGE");
((DataBaseConnector) con).setTableName_DATALOG("PLCCOM_DATALOG");

//add Connector to Dataserver
myDataServer.addOrReplaceLoggingConnector(con);

//start PLCcom data server
myDataServer.startServer();
input.readLine();
```

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Data archiving

When using the data archiving functionality all changes of variables will be logged in the **PLCCOM_DATALOG** archiving table. You can use this data for any purpose in any way.

#	ID	TS	DATASOURCE	DATATYPE	VARNAME	VALUE	QUALITY
1	251	09.01.2015 17:07:45	modbus	UShort	Register2	500	OK
2	253	09.01.2015 17:08:39	modbus	UShort	Register2	504	OK
3	252	09.01.2015 17:07:45	modbus	Boolean	CoilTest	True	OK
4	254	09.01.2015 17:08:39	modbus	Boolean	CoilTest	False	OK
5	260	09.01.2015 17:10:11	SPS1200	Boolean	E1	False	OK
6	261	09.01.2015 17:10:11	SPS1200	Boolean	Ausgang01	False	OK
7	263	09.01.2015 17:10:26	SPS1200	Boolean	E1	False	OK
8	265	09.01.2015 17:10:31	SPS1200	Boolean	E1	True	OK
9	269	09.01.2015 17:10:42	SPS1200	Boolean	E1	False	OK
10	240	09.01.2015 17:04:35	SPS1200	Boolean	E1	False	OK
11	257	09.01.2015 17:10:11	modbus	UShort	Register2	534	OK
12	258	09.01.2015 17:10:11	modbus	Boolean	Regist	False ···	OK

The following information is saved for each record:

Column	Column type	Information
ID	NUMBER(*, 0) Not Null	Sequential ID, Primary key
TS	TIMESTAMP(6) Not Null	Timestamp
DATASOURCE	VARCHAR2(50) Not Null	Datasource name
DATATYPE	VARCHAR2(50) Not Null	Data type
VARNAME	VARCHAR2(100) Not Null	Variable name
VALUE	VARCHAR2(2048) Null	Variable value
QUALITY	VARCHAR2(50) Null	Quality (GOOD if the communication between PLCcom and device works fine or BAD if problems occurred (e.g. timeouts)

Current data image

With the aid of a data image, you can access the variable pool of the PLCcom system from an external application.

For this purpose the data will be saved in the **PLCCOM_DATAIMAGE** table.

#	TS	DATASOURCE	DATATYPE	VARNAME	VALUE	QUALITY
1	09.01.2015 17:10:39	modbus	Boolean	Coil100	True	OK
2	09.01.2015 17:10:11	modbus	Short	Register1	28	OK
3	09.01.2015 17:10:11	modbus	Short	Register2	534	OK

For each variable there's a record in the table. The following information is saved for each variable:

Column	Column type	Information
TS	TIMESTAMP(6) Not Null	Timestamp
DATASOURCE	VARCHAR2(50) Not Null	Datasource name
DATATYPE	VARCHAR2(50) Not Null	Datatype
VARNAME	VARCHAR2(100) Not Null	Variable name
VALUE	VARCHAR2(2048) Null	Variable value
QUALITY	VARCHAR2(50) Null	Quality (GOOD if the communication between PLCcom and device works fine or BAD if problems occurred (e.g. timeouts)

Any questions?

Please write an email to support@indi-systems.de.

We will process your request promptly or respond to you directly.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39