Microsoft SQL connection to Sysmac NJ

Quick Start Guide

This Quick Start will show you how to connect to a Microsoft SQL database – it will not show you how to set up the database.

Watch the corresponding video on YouTube: Connect Sysmac NJ to Microsoft SQL Server
http://youtu.be/ys3Mn2zP-wA

Introduction:
The NJ501 series controllers have three new part numbers NJ501-1320, 1420, and 1520 which have the ability to send data directly to a Microsoft SQL database and an Oracle SQL database.
This Quick Start will show you how to connect to a database – it will not show you how to set up the database.

Why the need for database connection?
Assembly lines need easy and fast way to log or get data on the assembly line for production and product data.

Why direct from PLC?
Assembly lines last up to 10 years. Very difficult to maintain a SCADA package for 10 years. The computer will not last that long and it is difficult to find someone to support a 10 year old version of a SCADA package.
Computers take a long time to boot – lost production time.
Computers need updates – IT has to service – no guarantee an update will not adversely affect the performance of the unit.

Why SQL type data bases:
SQL databases accept many connections at once unlike databases like Microsoft Access which only allow one user at a time. Many PLC's can send data at once and many people can get data from the database at once.

Other Features:
The NJ can store the data to a spool file (1Mbyte in NJ memory) if connection to data base is lost.
The NJ can log commmands and responses to files on the NJ SD card which come in handy to debug the logging error.

Versions Required:
Sysmac Studio V1.06 with database patch or V1.07.

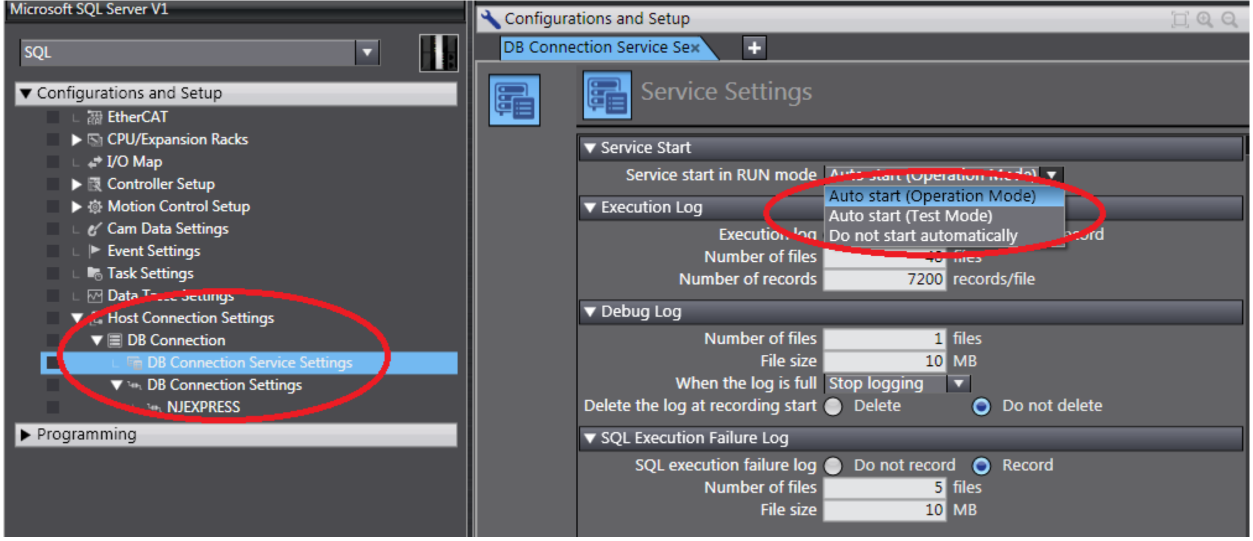
Example of How to Connect to Microsoft SQL

Setup:
In this example my laptop has IP address 192.168.250.9 and has Microsoft SQL installed on it.
NJ uses address 192.168.250.1

Introduction:
The SQL DB connection is setup in the "Configurations and Setup" in Sysmac Studio.
Show " Host Connection Setting" in the setup section of Sysmac Studio. This lets you test and setup your connection to the SQL Host.

- DB Connection Service Settings – specify error files and enable service
- DB Connection Settings – set logon to SQL server

Under DB Connection Service Setting – pick "Auto Start" to make the SQL service start on power up. Test mode will send all SQL commands to SD memory card and all the program instructions will assume connect or data transfer was good – for debugging.



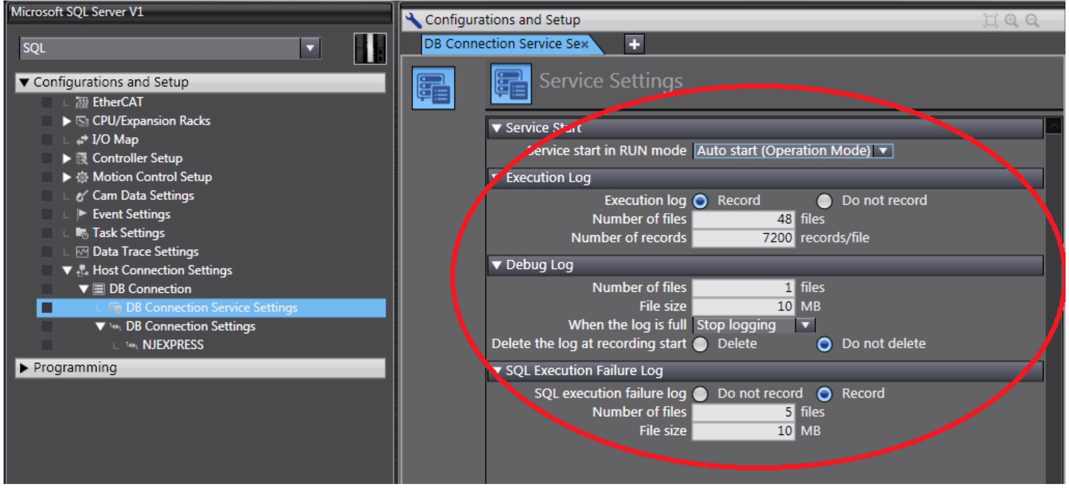
Service Start
Service start in RUN mode: Auto start (Operation Mode)
- Auto start (Operation Mode)
- Auto start (Test Mode)
- Do not start automatically

Execution Log
Execution log: Record
Number of files: 48 files
Number of records: 7200 records/file

Debug Log
Number of files: 1 files
File size: 10 MB
When the log is full: Stop logging
Delete the log at recording start: ○ Delete ● Do not delete

SQL Execution Failure Log
SQL execution failure log: ○ Do not record ● Record
Number of files: 5 files
File size: 10 MB

Execution Log, Debug Log, and SQL Execution Failure Log all go to the SD card.



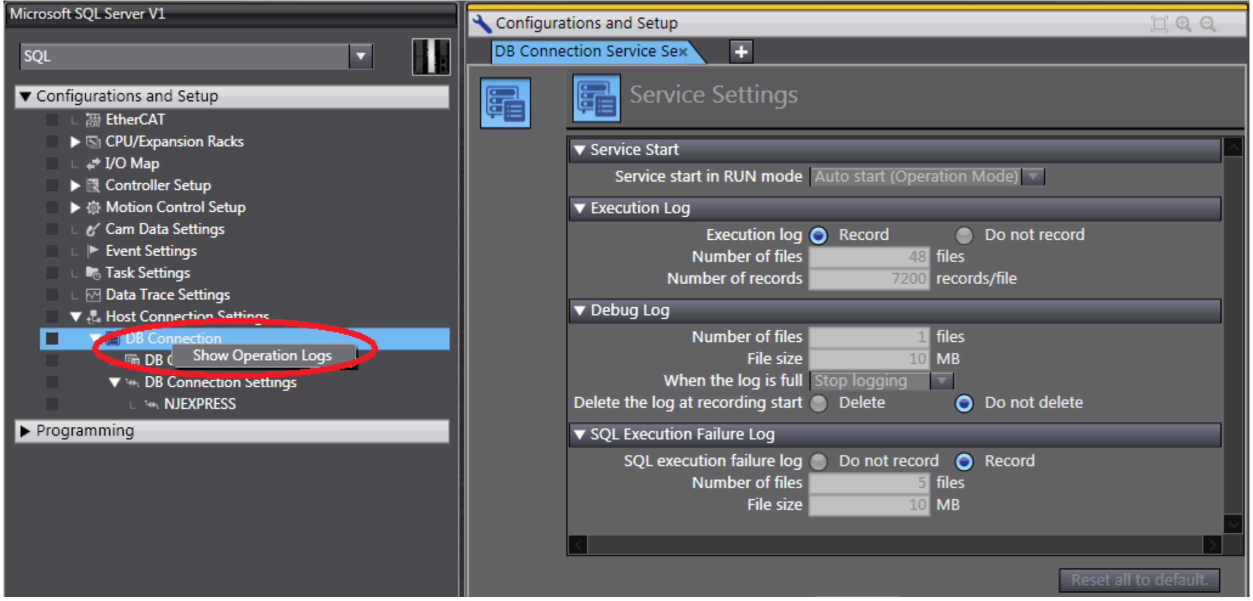
Service Start
Service start in RUN mode: Auto start (Operation Mode)

Execution Log
Execution log: ● Record ○ Do not record
Number of files: 48 files
Number of records: 7200 records/file

Debug Log
Number of files: 1 files
File size: 10 MB
When the log is full: Stop logging
Delete the log at recording start: ○ Delete ● Do not delete

SQL Execution Failure Log
SQL execution failure log: ○ Do not record ● Record
Number of files: 5 files
File size: 10 MB

If you right click on “DB Connection” then you can see the Operation Logs – assuming you have a least tried to log on or safe a record. You must be online to see the files.



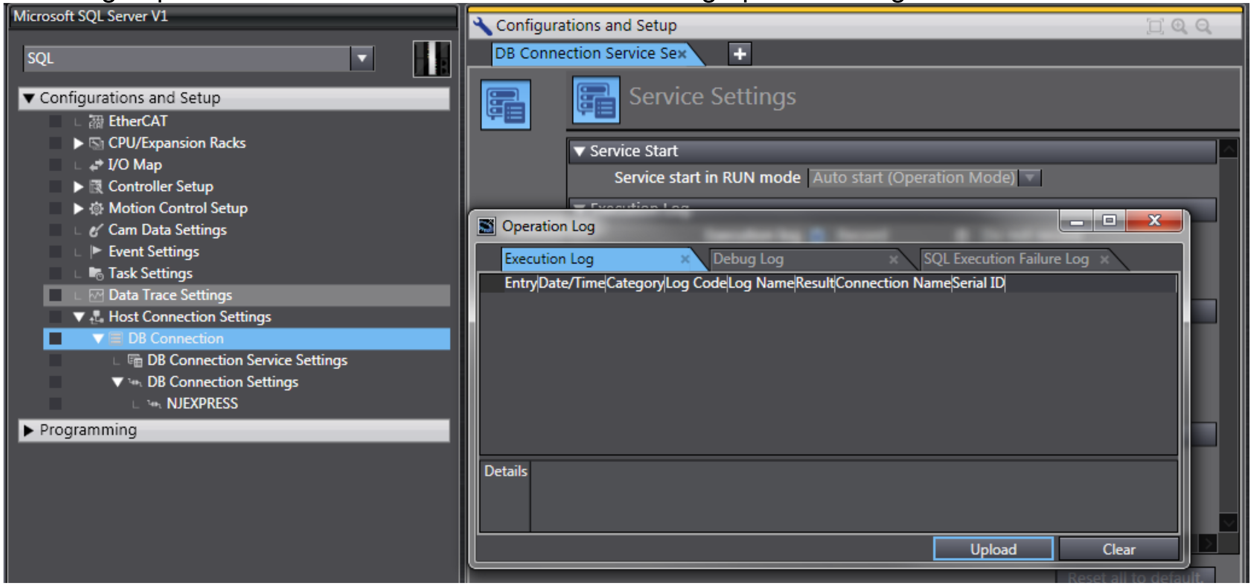
Service Start
Service start in RUN mode: Auto start (Operation Mode)

Execution Log
Execution log ● Record ○ Do not record
Number of files 48 files
Number of records 7200 records/file

Debug Log
Number of files 1 files
File size 10 MB
When the log is full: Stop logging
Delete the log at recording start ○ Delete ● Do not delete

SQL Execution Failure Log
SQL execution failure log ○ Do not record ● Record
Number of files 5 files
File size 10 MB

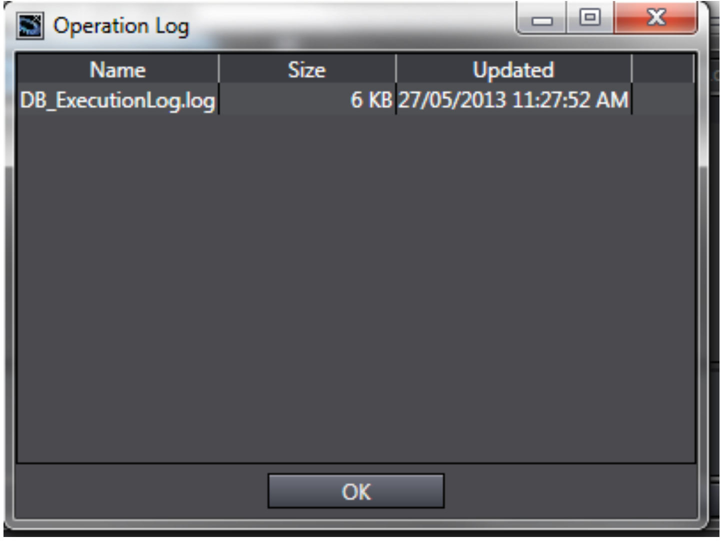
An “Operation Log” Window will open. There are three tabs – one for each of the log types.
Pressing “Upload” at the bottom of the screen will bring up a list of log files that exist on the SD card.



Service Start
Service start in RUN mode: Auto start (Operation Mode)

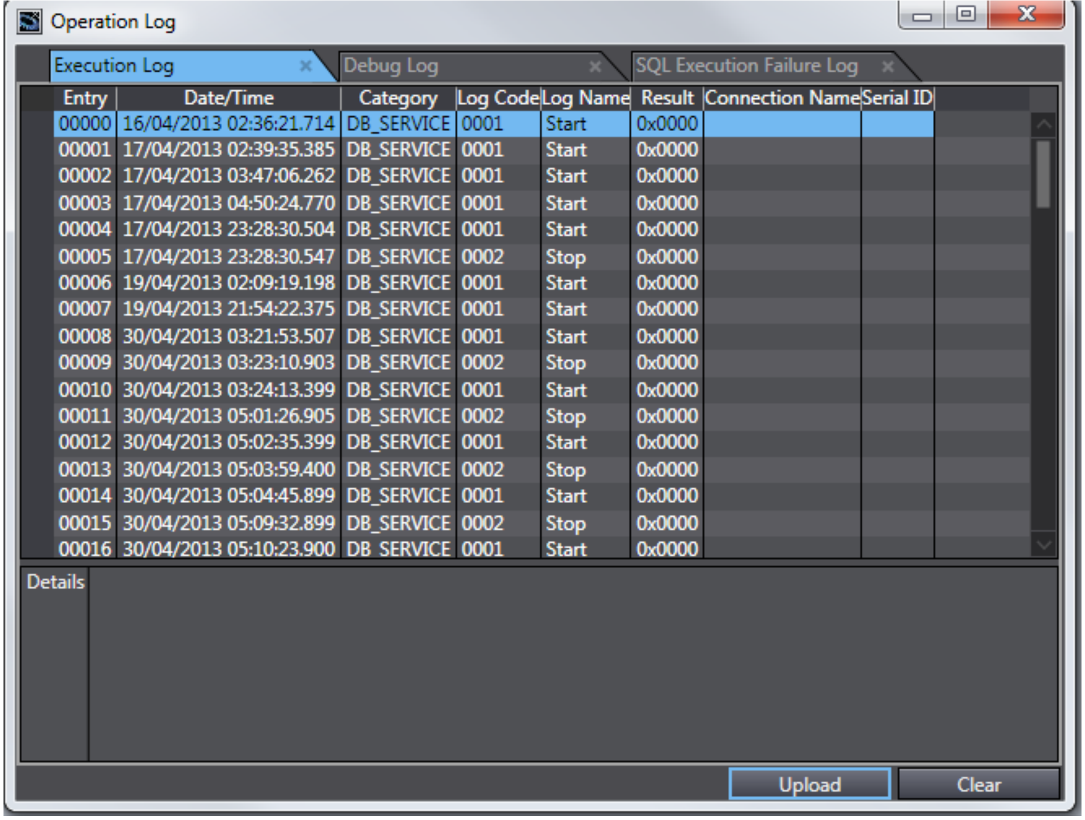
Entry	Date/Time	Category	Log Code	Log Name	Result	Connection Name	Serial ID

Select on of the files an press “OK”



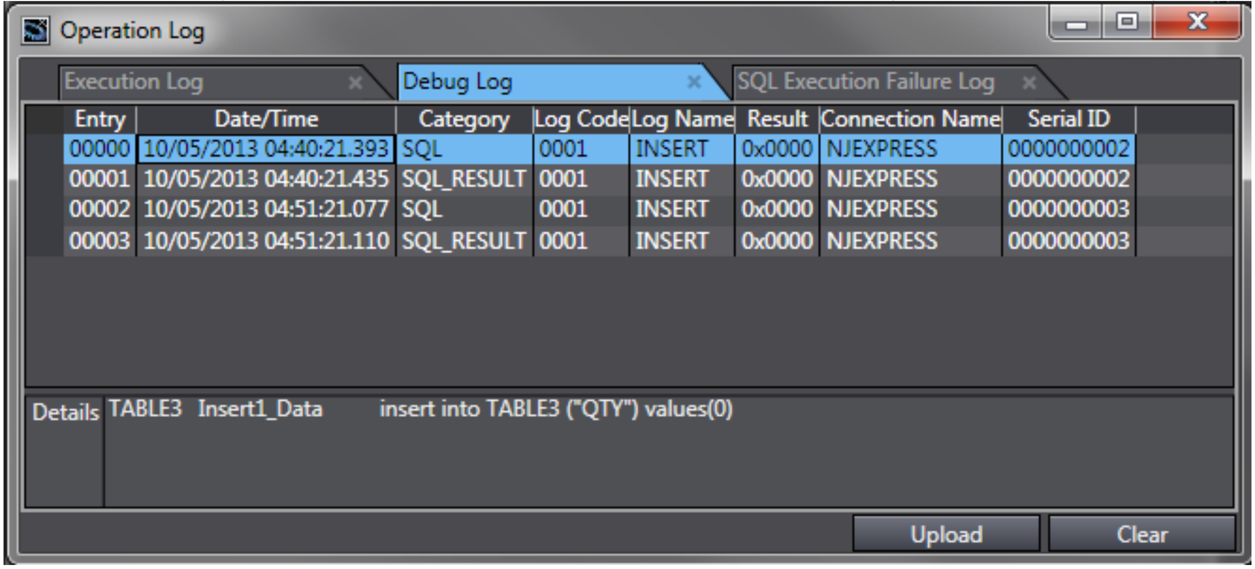
Name	Size	Updated
DB_ExecutionLog.log	6 KB	27/05/2013 11:27:52 AM

The Operation Log is good for seeing when services start and stop.



Entry	Date/Time	Category	Log Code	Log Name	Result	Connection Name	Serial ID
00000	16/04/2013 02:36:21.714	DB_SERVICE	0001	Start	0x0000		
00001	17/04/2013 02:39:35.385	DB_SERVICE	0001	Start	0x0000		
00002	17/04/2013 03:47:06.262	DB_SERVICE	0001	Start	0x0000		
00003	17/04/2013 04:50:24.770	DB_SERVICE	0001	Start	0x0000		
00004	17/04/2013 23:28:30.504	DB_SERVICE	0001	Start	0x0000		
00005	17/04/2013 23:28:30.547	DB_SERVICE	0002	Stop	0x0000		
00006	19/04/2013 02:09:19.198	DB_SERVICE	0001	Start	0x0000		
00007	19/04/2013 21:54:22.375	DB_SERVICE	0001	Start	0x0000		
00008	30/04/2013 03:21:53.507	DB_SERVICE	0001	Start	0x0000		
00009	30/04/2013 03:23:10.903	DB_SERVICE	0002	Stop	0x0000		
00010	30/04/2013 03:24:13.399	DB_SERVICE	0001	Start	0x0000		
00011	30/04/2013 05:01:26.905	DB_SERVICE	0002	Stop	0x0000		
00012	30/04/2013 05:02:35.399	DB_SERVICE	0001	Start	0x0000		
00013	30/04/2013 05:03:59.400	DB_SERVICE	0002	Stop	0x0000		
00014	30/04/2013 05:04:45.899	DB_SERVICE	0001	Start	0x0000		
00015	30/04/2013 05:09:32.899	DB_SERVICE	0002	Stop	0x0000		
00016	30/04/2013 05:10:23.900	DB SERVICE	0001	Start	0x0000		

The operation log shows when the database commands occurred – good for the line programmer.

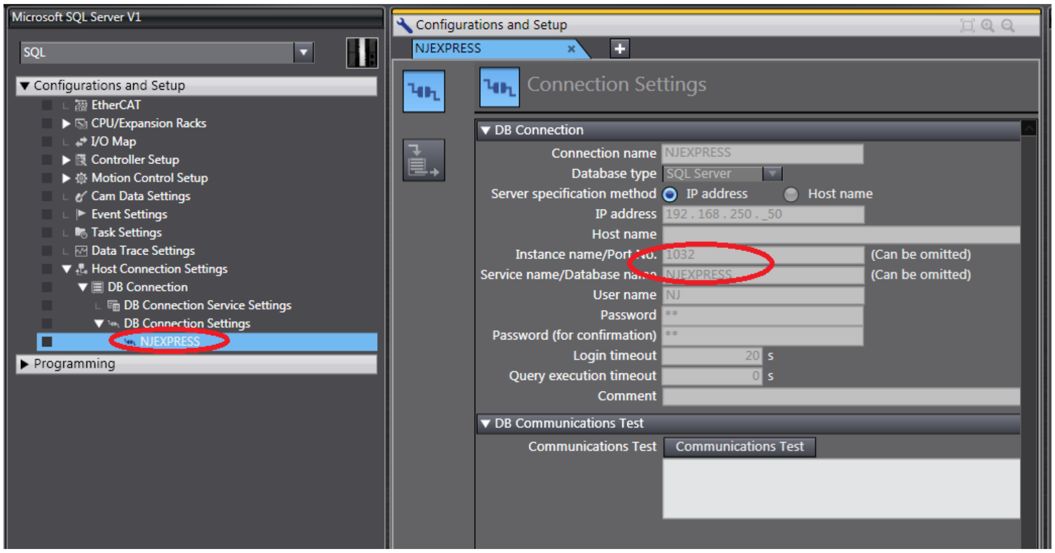


Entry	Date/Time	Category	Log Code	Log Name	Result	Connection Name	Serial ID
00000	10/05/2013 04:40:21.393	SQL	0001	INSERT	0x0000	NJEXPRESS	0000000002
00001	10/05/2013 04:40:21.435	SQL_RESULT	0001	INSERT	0x0000	NJEXPRESS	0000000002
00002	10/05/2013 04:51:21.077	SQL	0001	INSERT	0x0000	NJEXPRESS	0000000003
00003	10/05/2013 04:51:21.110	SQL_RESULT	0001	INSERT	0x0000	NJEXPRESS	0000000003

Details TABLE3 Insert1_Data insert into TABLE3 ("QTY") values(0)

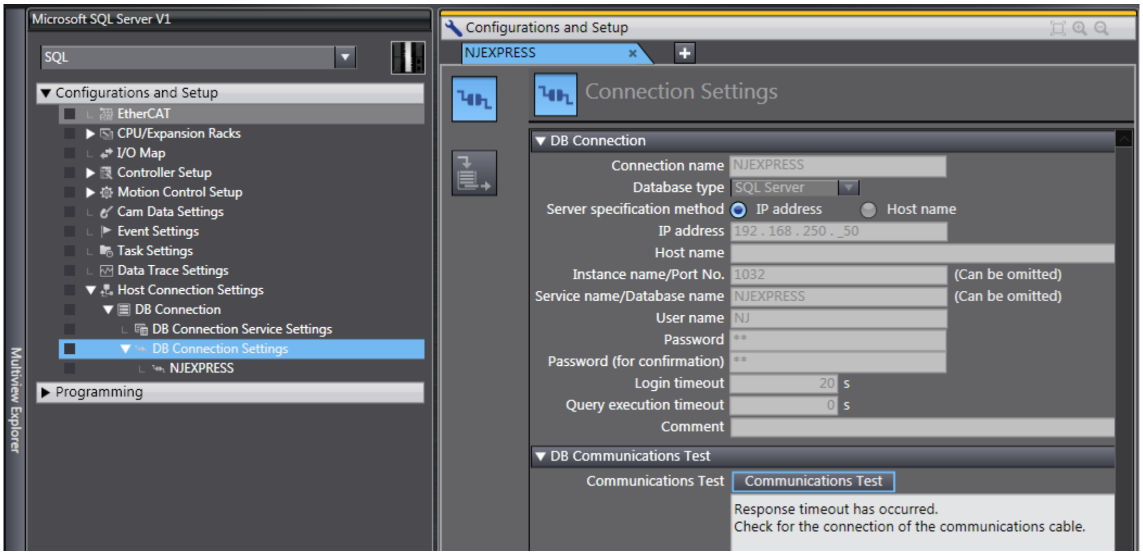
The SQL Execution Failure Log shows the (to and from) SQL commands – good for IT and high level NJ/SQL programmer. These logs are good for showing why a log did not happen – i.e. empty field on a field that cannot be empty.

How to create the connection profile:
Go to “DB Connection Settings – Right Click – “Add” – “DB Connection Settings”. This will create a new connection setting – in this case “NJEXPRESS” is what I labeled this connection setting – “Connection Name”.
Database Type: SQL Server for Microsoft SQL and Oracle for Oracle SQL.
Server Specification Method: IP address or Host name of the computer the SQL server is on.
Instance Name/Port Number: 1433 is the default. You do not need to specify if it is the default.
Service Name/Database Name: You must enter the database name here as specified by IT people. You can omit the name if the user has been set by default to the correct Service/Database within SQL.
You must then enter the Login Name and Password as given by IT people.



Connection name: NJEXPRESS
Database type: SQL Server
Server specification method: ● IP address ○ Host name
IP address: 192 . 168 . 250 . 50
Host name:
Instance name/Port No.: 1032 (Can be omitted)
Service name/Database name: NJEXPRESS (Can be omitted)
User name: NJ
Password: **
Password (for confirmation): **
Login timeout: 20 s
Query execution timeout: 0 s
Comment:
Communications Test: Communications Test

You can now press the Communications Test Button.
“Test OK” will show if successful.
In this case the controller was not connected to the SQL server so an error message is shown.



Connection name: NJEXPRESS
Database type: SQL Server
Server specification method: ● IP address ○ Host name
IP address: 192 . 168 . 250 . 50
Host name:
Instance name/Port No.: 1032 (Can be omitted)
Service name/Database name: NJEXPRESS (Can be omitted)
User name: NJ
Password: **
Password (for confirmation): **
Login timeout: 20 s
Query execution timeout: 0 s
Comment:
Communications Test: Communications Test

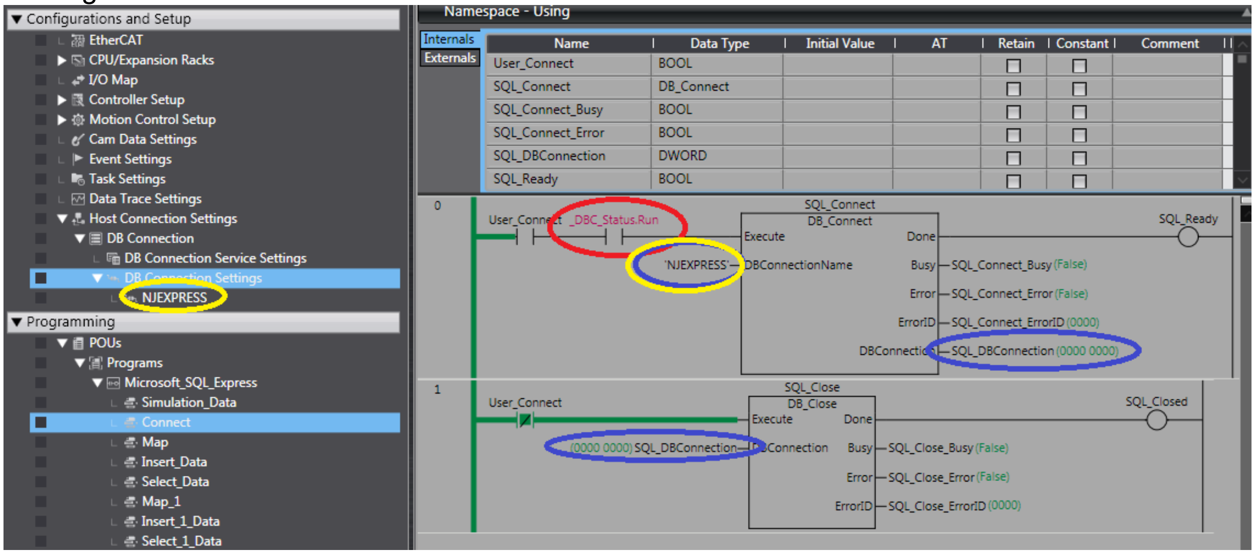
Response timeout has occurred.
Check for the connection of the communications cable.

Ladder Code to send data to SQL:

You have created a connection setup log into the SQL data base and tested it.
Now we want to add ladder code to send data to/from the SQL server.
There is 3 basic parts to this step:

1) DB_Connect and DB_Close instructions to log in and log out of the SQL the database.
2) A DB_CreateMapping instruction to create a map between NJ Tags and SQL fields/Table.
3) DB_Insert/Select/Delete/Update instructions.
    - Insert: to append data to the table
    - Select: which allows you to query data from the table
    - Delete: delete a record.
    - UpDate: modify a record.

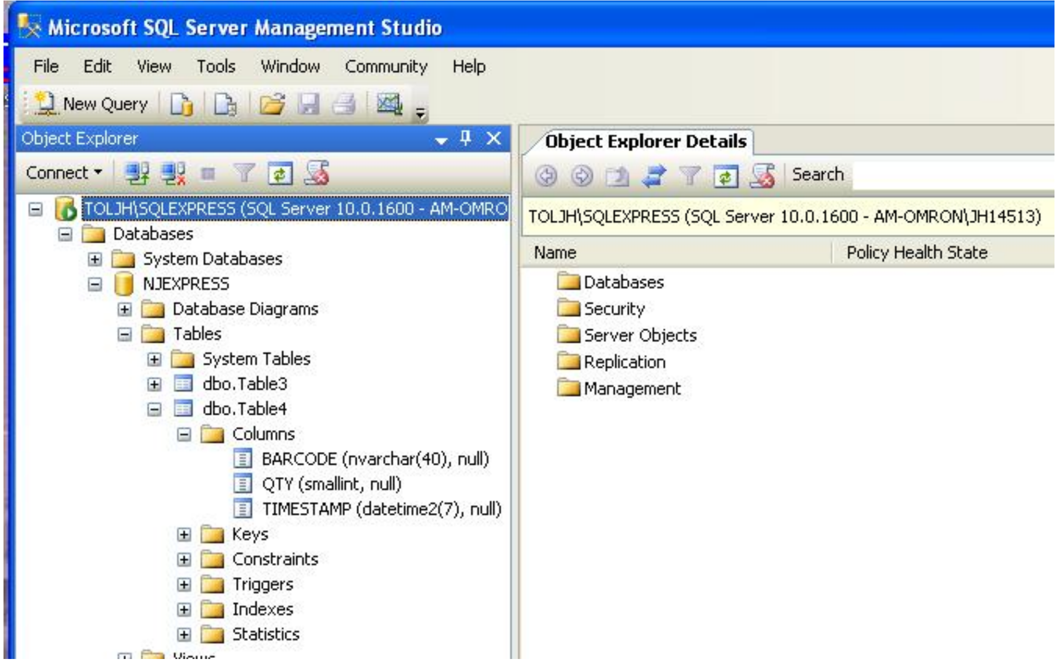
This is an example of how to write the connection program:
The _DBC_Status.Run bit lets you know the DB service has actually started up and is running.
'NJEXPRESS' with single quotes is how you specify the DB Connection you want to connect use.
The "Done" bit lets you know if you successfully connected to the database/table.
The SQL_DBConnection variable is the reference variable to this database connection for all the other instructions including SQL_Close.
It is a good idea to close the database when the controller is about to shut down.



Name	Data Type	Initial Value	AT	Retain	Constant	Comment
User_Connect	BOOL			□	□	
SQL_Connect	DB_Connect			□	□	
SQL_Connect_Busy	BOOL			□	□	
SQL_Connect_Error	BOOL			□	□	
SQL_DBConnection	DWORD			□	□	
SQL_Ready	BOOL			□	□	

The next step is to map the NJ Tags to the Database Tables Fields
DBConnection: This came from the "DB_Connect Instruction" – unique number identifier for each connection.
TableName: This is the name of the Table you want to send data to/from in the SQL server.
SQLType: Enter one of the following: DBC_SQLTYPE_INSERT, DBC_SQLTYPE_SELECT, DBC_SQLTYPE_DELETE, DBC_SQLTYPE_UPDATE.

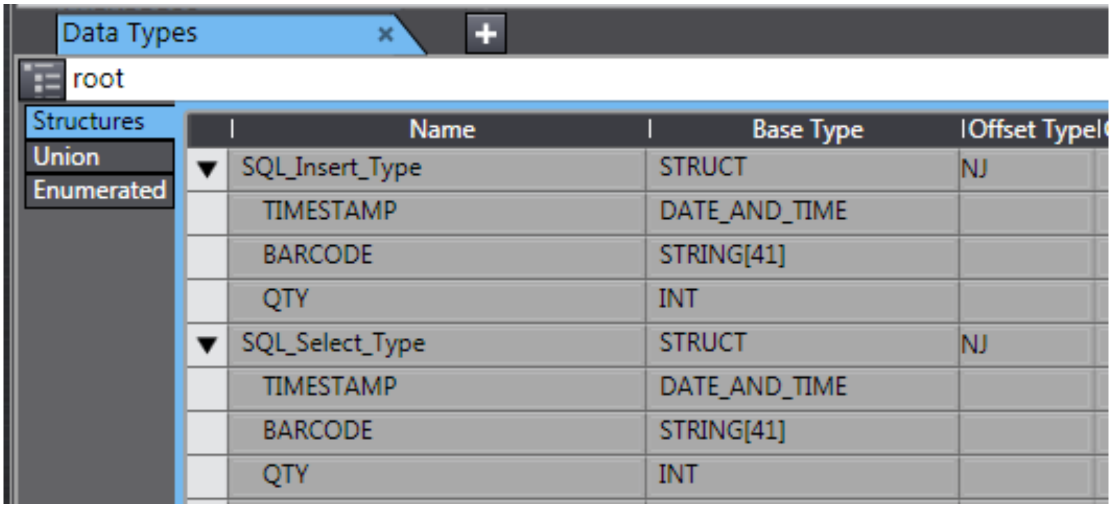
You must enter a unique variable name for “MapVar” even if it is for the same table – or only the last one will work. The MapVar must be of a structure type. Here is an example of an NJ Structure and the SQL table. In this case under Table4 of Database NJEXPRESS you will see Barcode, Qty, and Timestamp.
The SQL_Insert_Type is exactly the same. The member names must match exactly the field names in the SQL database. You do not have to have all the field names – nor do they have to be in order. The variable types will not match exactly and the manual shows you how to match them up. I have included most of the table below.
The “done” output in the DB_Mapping means nothing. No data transfer occurs with the SQL database. It does not mean the mapping was good.



- TOLJH\SQLEXPRESS (SQL Server 10.0.1600 - AM-OMRO
  - Databases
    - System Databases
    - NJEXPRESS
      - Database Diagrams
      - Tables
        - System Tables
        - dbo.Table3
        - dbo.Table4
          - Columns
            - BARCODE (nvarchar(40), null)
            - QTY (smallint, null)
            - TIMESTAMP (datetime2(7), null)
          - Keys
          - Constraints
          - Triggers
          - Indexes
          - Statistics

TOLJH\SQLEXPRESS (SQL Server 10.0.1600 - AM-OMRON\JH14513)

Name	Policy Health State
Databases	
Security	
Server Objects	
Replication	
Management	



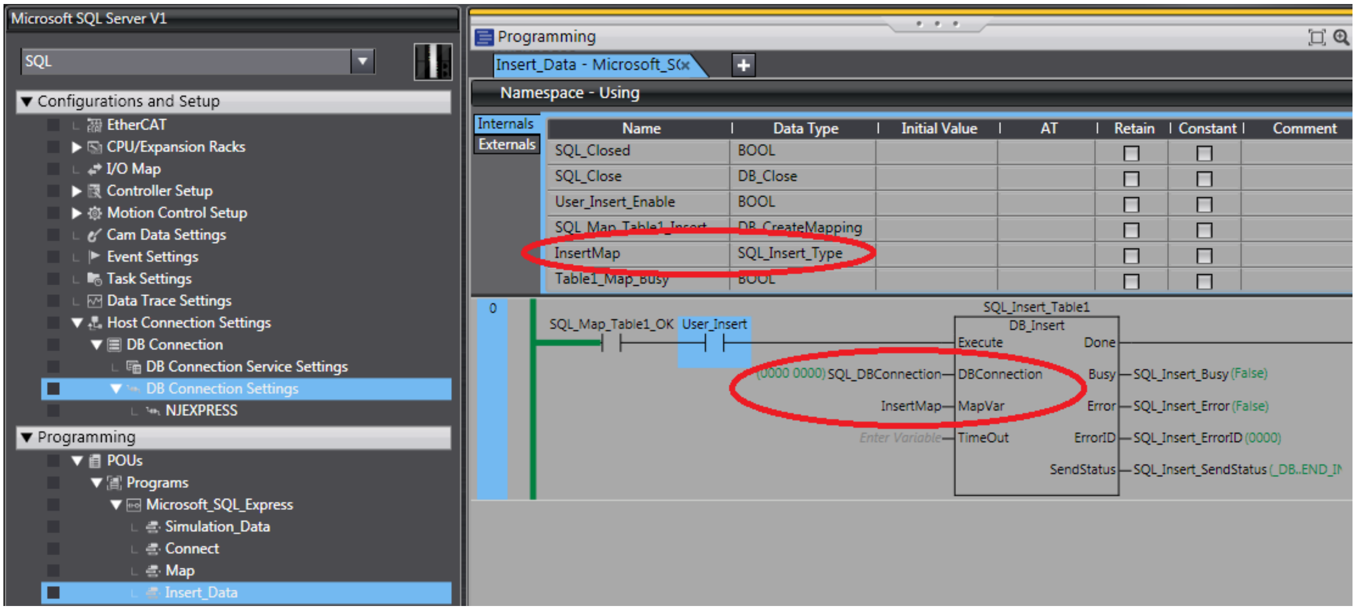
	Name	Base Type	Offset Type
▼	SQL_Insert_Type	STRUCT	NJ
	TIMESTAMP	DATE_AND_TIME	
	BARCODE	STRING[41]	
	QTY	INT	
▼	SQL_Select_Type	STRUCT	NJ
	TIMESTAMP	DATE_AND_TIME	
	BARCODE	STRING[41]	
	QTY	INT	

Data Type	Data Type specified in IEC 61131	Oracle	SQL Server
Boolean / bit string	BOOL	NUMBER(1)	bit
	BYTE	RAW	varbinary
	WORD	RAW	varbinary
	DWORD	RAW	varbinary
	LWORD	RAW	varbinary
Signed integers	SINT	NUMBER(3)	tinyint
	INT	NUMBER(5)	smallint
	DINT	NUMBER(10)	int
	LINT	NUMBER(20)	bigint
Unsigned integers	USINT	NUMBER(3)	smallint
	UINT	NUMBER(5)	int
	UDINT	NUMBER(10)	bigint
	ULINT	NUMBER(20)	decimal
Real	REAL	NUMBER BINARY_FLOAT	real
	LREAL	NUMBER BINARY_DOUBLE	float
Time of day, duration, date	TIME	TIMESTAMP	time
	DATE	TIMESTAMP	date
	TIME_OF_DATE	TIMESTAMP	time
	DATE_AND_TIME	TIMESTAMP	datetime
Text string	STRING	VARCHAR2 NVARCHAR2 <A03>	varchar nvarchar

To Insert Data into the SQL database:

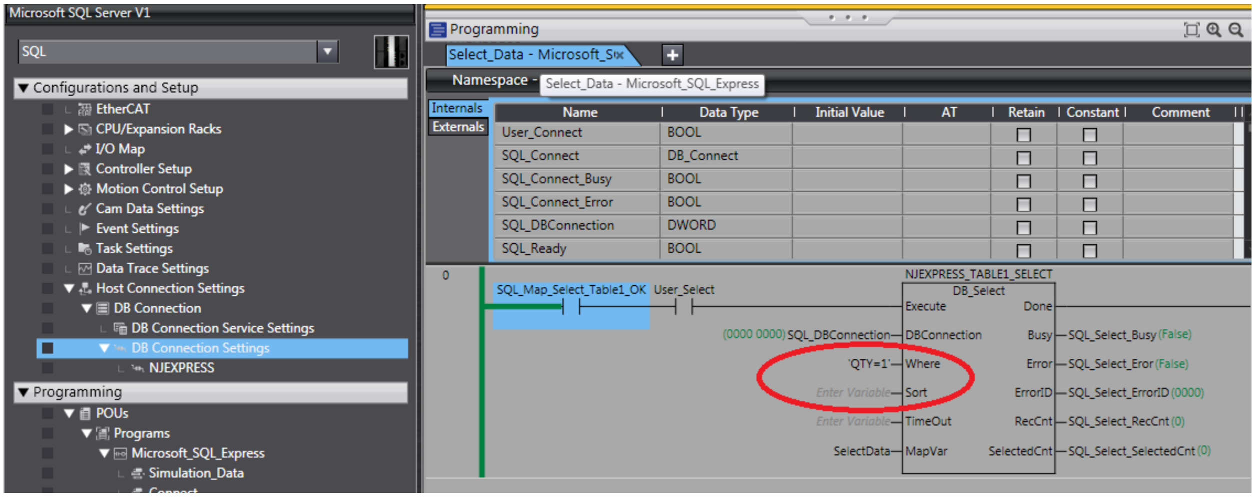
The DB_Insert instruction needs two reference inputs:

- DB_Connection comes from the DB_Connect instruction.
- MapVar is the same variable we used in the DB_CreateMapping instruction.
- With these two pieces of information the insert instruction know which database, table, and fields to add the data too.
- The "done" output does tell you if the data was successfully sent to the SQL database.



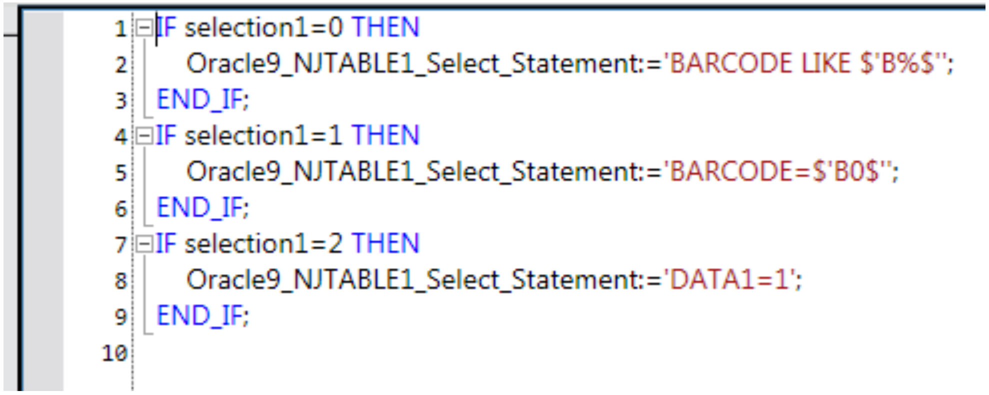
Name	Data Type	Initial Value	AT	Retain	Constant	Comment
SQL_Closed	BOOL			□	□	
SQL_Close	DB_Close			□	□	
User_Insert_Enable	BOOL			□	□	
SQL_Map_Table1_Insert	DB_CreateMapping			□	□	
InsertMap	SQL_Insert_Type			□	□	
Table1_Map_Busy	BOOL			□	□	

To Select Data from the database:
The DB_Select instruction does this for us. It works mostly the same as DB_Insert with two major differences.
The MapVar can be an array. (Notice there is not “[0]” at the end of the variable name when it is an array.
There is a “Where” input to filter which records you are looking for. (There is also a sort input).
There are two counters at the bottom right corner of the instruction to tell you how many records it found and how many it gave to you (if your array was to small it just gives you enough to fill the array.)
Notes on how to use the “Where” are below.



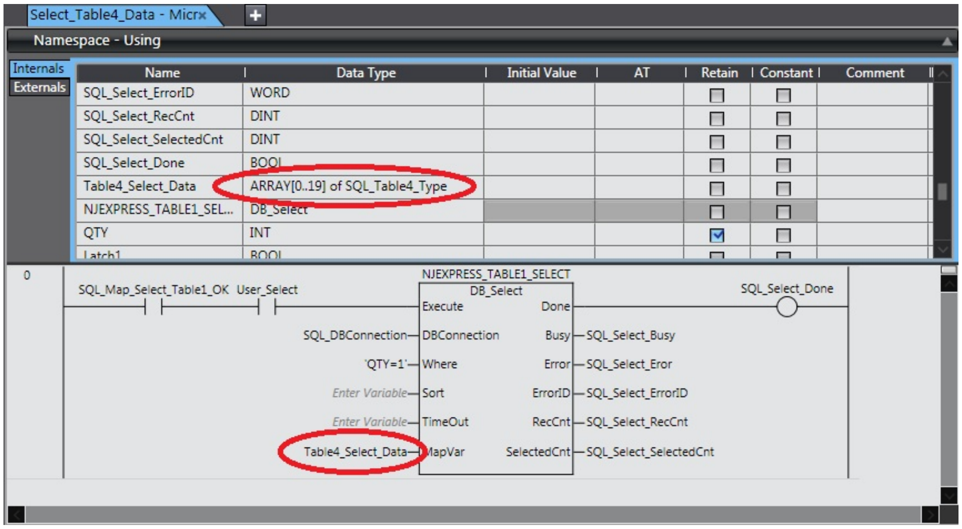
Name	Data Type	Initial Value	AT	Retain	Constant	Comment
User_Connect	BOOL			□	□	
SQL_Connect	DB_Connect			□	□	
SQL_Connect_Busy	BOOL			□	□	
SQL_Connect_Error	BOOL			□	□	
SQL_DBConnection	DWORD			□	□	
SQL_Ready	BOOL			□	□	

Both SQL and Sysmac Studio use ‘ ‘ in their syntex. To let Sysmac Studio the ‘ is for Oracle put a $’.
So the following first example means: BARCODE LIKE ‘B%’ to Oracle. The % is a wildcard to Oracle when the “LIKE” is included.



```
IF selection1=0 THEN
    Oracle9_NJTABLE1_Select_Statement:='BARCODE LIKE $'B%$'';
END_IF;
IF selection1=1 THEN
    Oracle9_NJTABLE1_Select_Statement:='BARCODE=$'B0$'';
END_IF;
IF selection1=2 THEN
    Oracle9_NJTABLE1_Select_Statement:='DATA1=1';
END_IF;
```

Here is an example of the Select type Variable for the answer – notice how it is an array so we can accept more than one record from the select query.



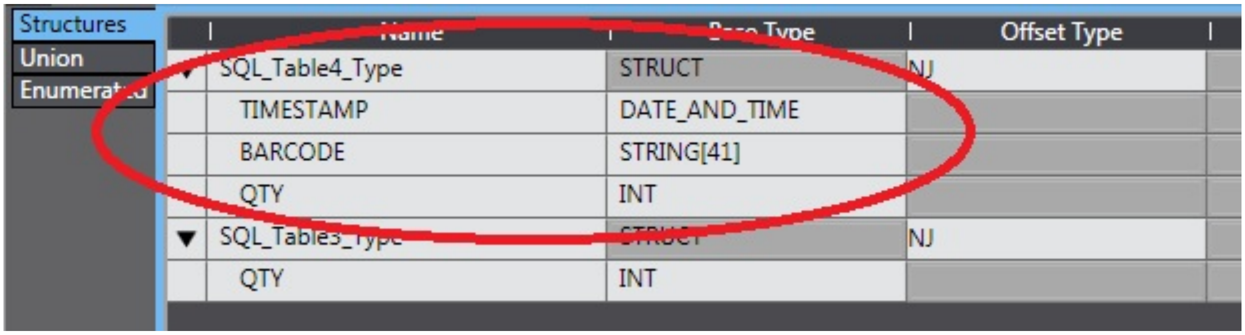
Select_Table4_Data - Micr

Namespace - Using

Internals | Externals

Name	Data Type	Initial Value	AT	Retain	Constant	Comment
SQL_Select_ErrorID	WORD			□	□	
SQL_Select_RecCnt	DINT			□	□	
SQL_Select_SelectedCnt	DINT			□	□	
SQL_Select_Done	BOOL			□	□	
Table4_Select_Data	ARRAY[0..19] of SQL_Table4_Type			□	□	
NJEXPRESS_TABLE1_SEL...	DB_Select			□	□	
QTY	INT			☑	□	
Latch1	BOOL			□	□	

Here is the data type:



Structures | Union | Enumerated

Name	Base Type	Offset Type
▼ SQL_Table4_Type	STRUCT	NJ
TIMESTAMP	DATE_AND_TIME	
BARCODE	STRING[41]	
QTY	INT	
▼ SQL_Table3_Type	STRUCT	NJ
QTY	INT	

This completes the Quick Start for the
Connect Sysmac NJ to Microsoft SQL Server

Please visit our YouTube Channel for Omron Quick Tip and other videos:
https://www.youtube.com/user/OmronAutomationTech