Machine Controls Traceability Interface

Siemens PLC

(For use with Traceability Application v11.9.0 or newer and Siemens Logic v2.0.0 or newer)

Global Common

SD-1056

ISSUED August 27, 2021
REVISED

Table of Contents

1 Scope and Purpose 3
2 Siemens Traceability v2.0 4

1 Scope and Purpose

1.1 Scope

1.1.1 This specification describes the traceability application configuration and PLC logic design requirements for Nexteer Automotive facilities utilizing Nexteer's traceability System.

1.1.2 This specification applies to the equipment requiring traceability communication for process flow, electronic error proofing, and data collection. Refer to the Manufacturing Engineer's written specification for details regarding traceability requirements.

1.1.3 This specification has associated PLC logic routines and HMI screens that reflect the requirements of this specification. In addition, the logic library provides the required routines and examples that may be applied to new equipment designs. All files are available at www.nexteerdataexchange.com.

1.1.4 The use of the word "shall" indicates requirements and the use of the word "should" indicates recommendations. The use of the word "may" indicates permission or allowance and the use of the word "can" indicates a possibility.

1.2 Purpose and Objectives

1.2.1 The purpose of this specification is to provide Nexteer requirements and guidance to Original Equipment Manufacturers (OEM) for use in their PLC logic designs to interface with Nexteer's traceability System.

1.2.2 The objective of this specification is to provide common, maintainable, and cost-effective traceability controls systems that enhance both the productivity and ease-of-use of the system, while ensuring the quality of Nexteer products produced. The application of this specification will result in common traceability controls systems.

1.2.3 The Nexteer traceability systems are integrated at the machine, cell (group of machines), or asynchronous assembly line level. Depending on the configuration of the traceability system, it may cover multiple cells and / or multiple asynchronous assembly lines.

1.2.4 The Nexteer traceability system uses a traceability computer, which runs the Nexteer traceability application and interfaces with the SQL Server traceability database.

2 Siemens Traceability v2.0

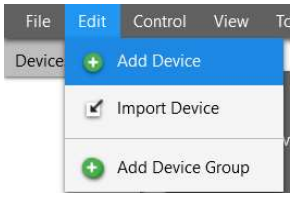
This section describes how to configure the Siemens v2.0 traceability plugin and corresponding logic templates. A template is provided for both Simatic Step 7 and TIA Portal 15 and the following PLC hardware is supported.

- S7-300
- S7-400
- S7-1200
- S7-1500

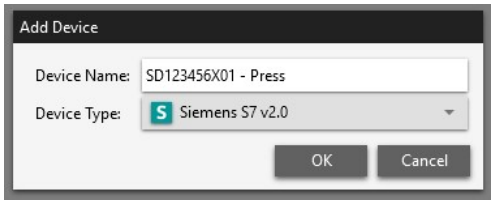
2.1 Traceability Application Configuration

2.1.1 Add a device connection.

1. Create a new device by clicking menu Edit > Add Device , or by right clicking the Devices list on the left side of the app and using the context menu.

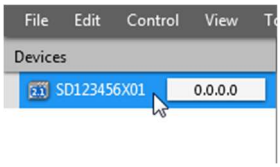


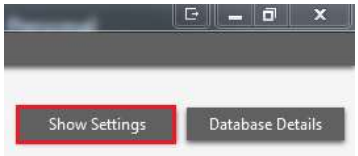
2. A dialog that is used to configure the new device connection will appear.
   a. Enter a name for the device. (For example: SD123456X01 - Press)
   b. Set the device type to Siemens v2.0 .
   c. Click OK to finish adding the device connection.



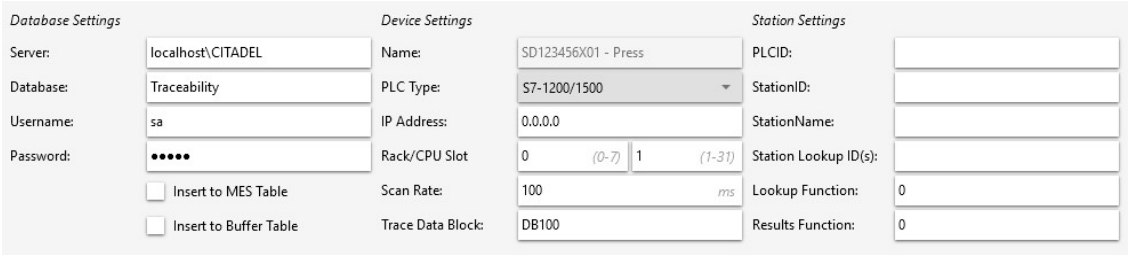
2.1.2 Device settings panel.

1. Ensure that the newly added device connection is selected by clicking on it in the Devices list on the left side of the app. Click the Show Settings button to display the device settings.



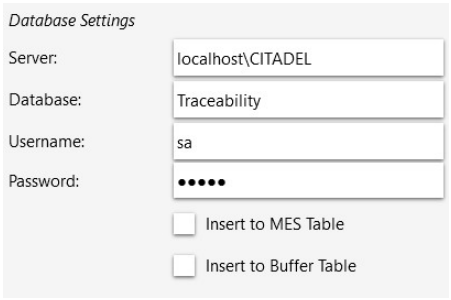


2. The following panel will be displayed. It is used to configure the database and PLC connection settings.



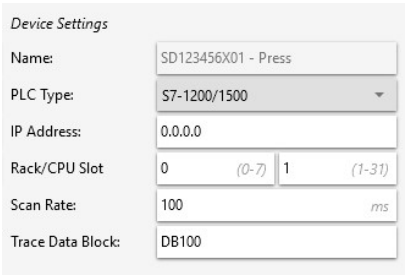
2.1.3 Configure Database Settings

The default database configuration settings normally do not require modification.



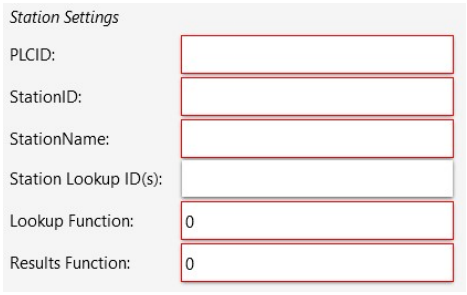
1. Server: The default Microsoft SQL server name is “localhost\CITADEL” which contains the standard Nexteer traceability database.
2. Database : The default database name is “Traceability”.
3. Username/Password : This contains the credentials for the authorized database user. The default username is “sa” and password is “admin”.
4. Insert to MES Table : Enable this checkbox if records from this device also need to be inserted into the dbo._status_mes table. These records are intended to be processed by an external MES system.
5. Insert to Buffer Table : Enable this checkbox if records from this device also need to be inserted into the dbo._status_buffer table. These records are intended to be moved to downstream database prior to a part arriving for permissions to work correctly.

2.1.4 Configure Device Settings



1. PLC Type : Select the type of Siemens S7 PLC. The application supports the following controllers.
   a. S7-300
   b. S7-400
   c. S7-1200/1500
2. IP Address : Enter the IP Address of the PLC.
3. Rack/CPU Slot : Enter the PLC rack and CPU slot numbers.
4. Scan Rate : The default scan rate of how often the traceability application polls the machine for new data is “100” milliseconds. This should not require modification.
5. Trace Data Block : Address of the traceability data block in the PLC. Explained in section 2.3.3 for TIA Portal and section 0 for Simatic Step 7 .

2.1.5 Configure Station Settings

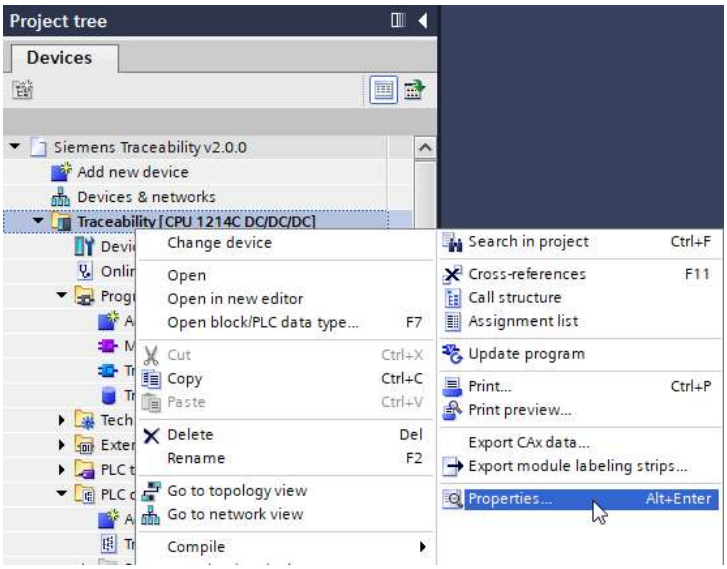


1. PLCID : Enter the machines SD number for the PLC ID. For example: SD123456X.
2. StationID : Enter a unique Station ID for the device configuration. For example: SD123456X01, SD123456X51, SD123456OP010.1, SD123456ST020, etc…
3. StationName : Enter a description for the station. The recommended station name should include the machine / cell or line / operation or station / description. For example: BSI_Line 1_OP10_Load Station
4. Station Lookup ID(s): Enter a comma separated list of StationID values, with no spaces, that are used for filtering function (2) and function (4) lookup requests. See the following lookup function descriptions for more detail.

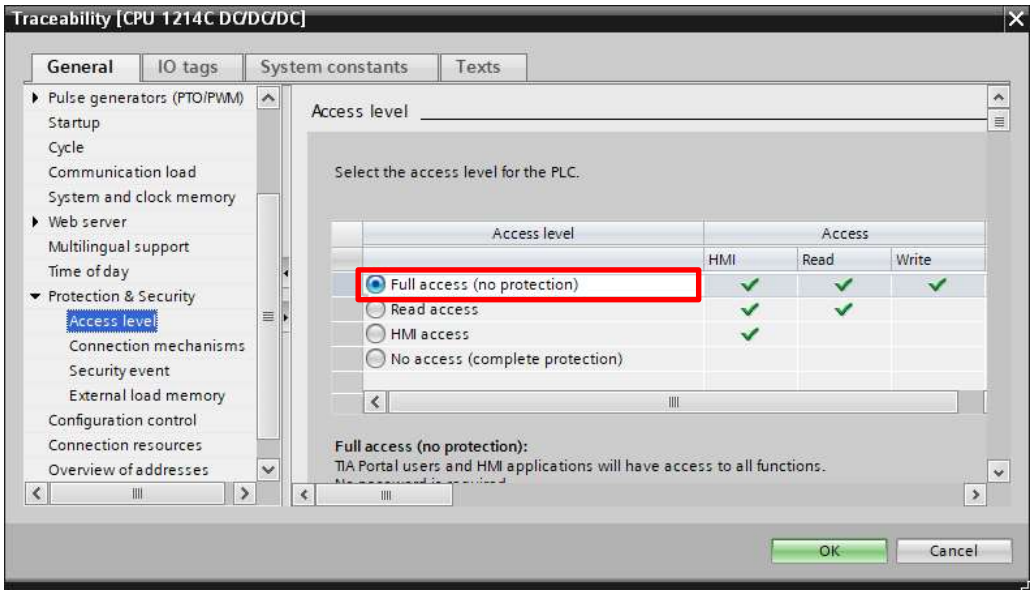
5. Lookup Function : Enter a numeric lookup function value from the following list.
    a. Function (1): Return the latest database record header information to the machine. Name/Data column information is not returned. This function is used primarily to check that a part was processed by the correct upstream machine and has an appropriate status to run.
    b. Function (2): Return the latest database record to the machine, filtered by the list of StationIDs specified by the Station Lookup ID(s) setting. This function is used to return data from a specific list of upstream stations and is not to be use for permissions.
    c. Function (3): Return latest record full record data to the machine. This function works the same as a function 1, but also includes Name/Data column information from the previous station.
    d. Function (4): Perform two queries on the database to find the header information from the latest record for permissions, and Name/Data information filtered by the list of StationIDs specified by the Station Lookup ID(s) setting. The header data from the first query, and the Name/Data information for the second query are combined into a single record and returned to the machine. The purpose of this function is to perform both function (1) and function (2) type requests at the same time using a single lookup for efficiency.
6. Results Function : Enter a numeric results function value from the following list. Function (14) is reserved for CSV import processes and function (15) is reserved for In Process requests.
    a. Function (10): Standard end of cycle results insert that contains part status and process data that needs to be inserted into the database.
    b. Function (12): Same functionality as function (10) and specifies that one or more subcomponent serial numbers exist in the Name/Data information.
    c. Function (20): Used for collecting large amounts of data that is intended to be sent directly to the Nextrace reporting system. The full data record is inserted into the dbo._nextrace_data table and a small record is also insert into the dbo._status table that contains station and status information to use for permissions.
    d. Function (21): Used for collecting large amounts of data that is intended to be sent directly to the Nextrace reporting system. The full data record is inserted into the dbo._nextrace_data table only.

2.2 TIA Portal 15 - General PLC Configuration

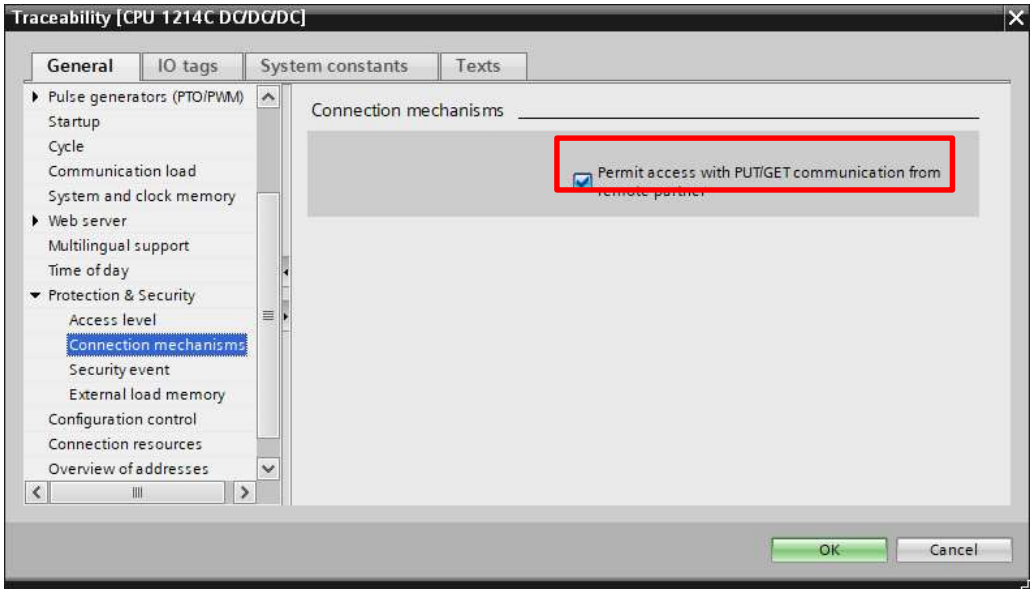
2.2.1 Configure the PLC to allow the traceability application to read/write tags. Right click on the PLC and open the Properties window.



1. Set the Access level for the PLC to Full access (no protection) .

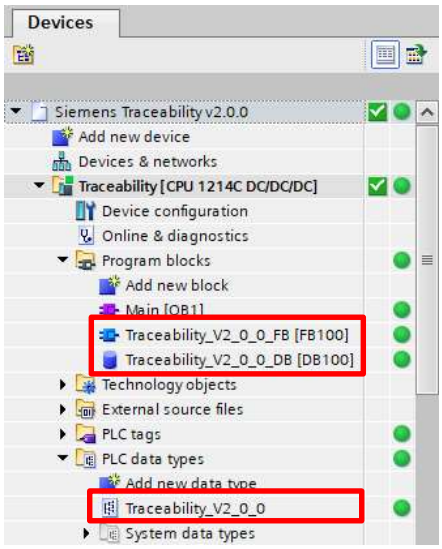


2. Enable option Permit access with PUT/GET communications from remote partner .

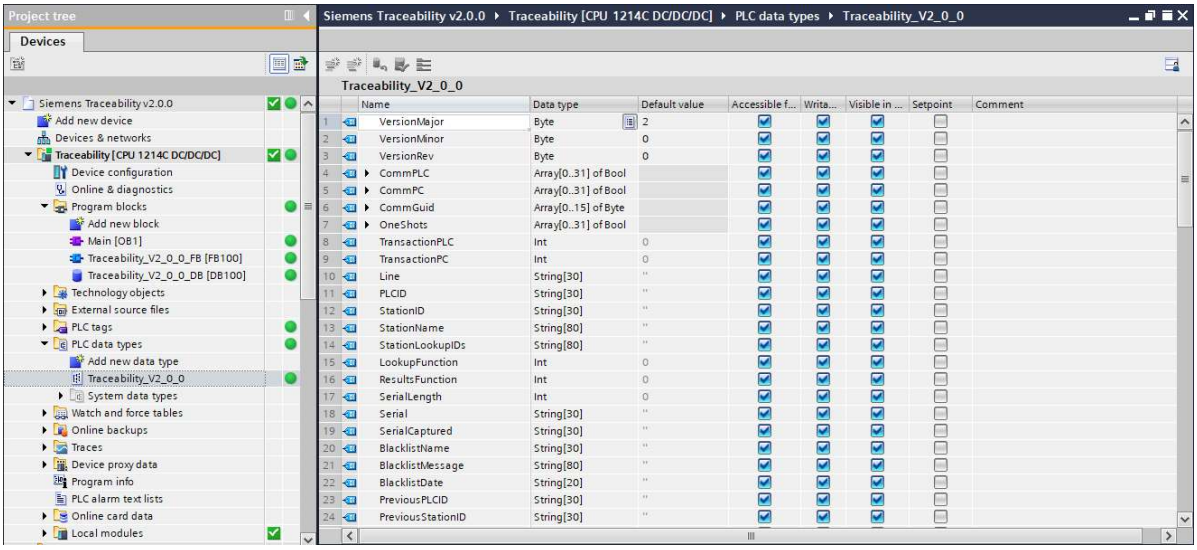


2.3 TIA Portal 15 - Logic Structure

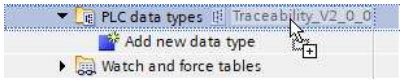
The Siemens traceability logic is comprised of a LAD Function Block and a corresponding Data Block to expose the Function Block's tags to the traceability application. The Function Block utilizes an instance of a common user defined type (UDT) to specify the required communication tags. The logic networks are required to be modified to meet the requirements of the application. See section 2.10 for a detailed timing chart showing logic and traceability application communication.



2.3.1 The traceability function block utilizes an instance of the user defined type (UDT) Traceability_V2_0_0 to specify the required communication tags.

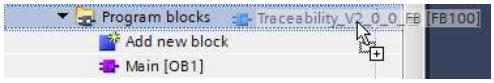


1. The UDT is shared by each traceability function block and can be copied from the template to the target program by dragging & dropping.

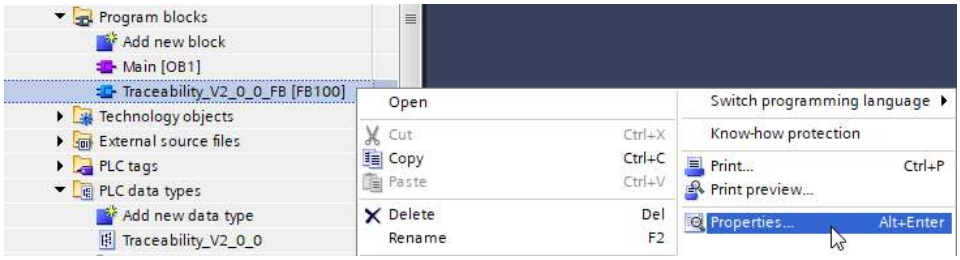


2.3.2 The LAD function block Traceability_V2_0_0_FB contains the logic to perform traceability operations for a single part during the cycle. The function block can be copied from the template to the target programming by dragging & dropping. A unique copy of the function block is required for each part that requires traceability functions to be performed during the cycle.

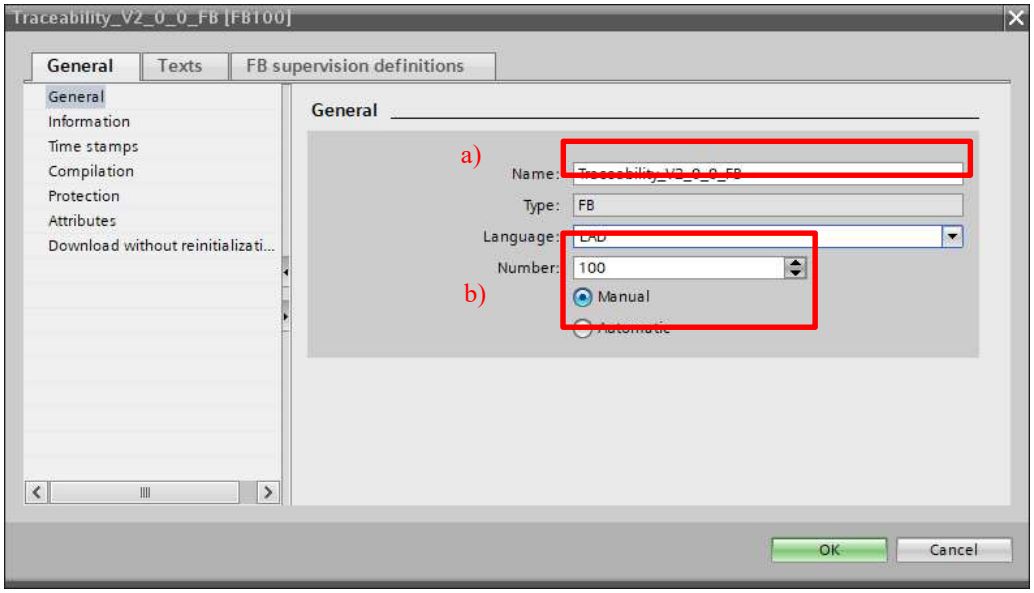
1. Drag and drop the function block from the template to the target program.



2. Right click on the imported function block and open Properties .



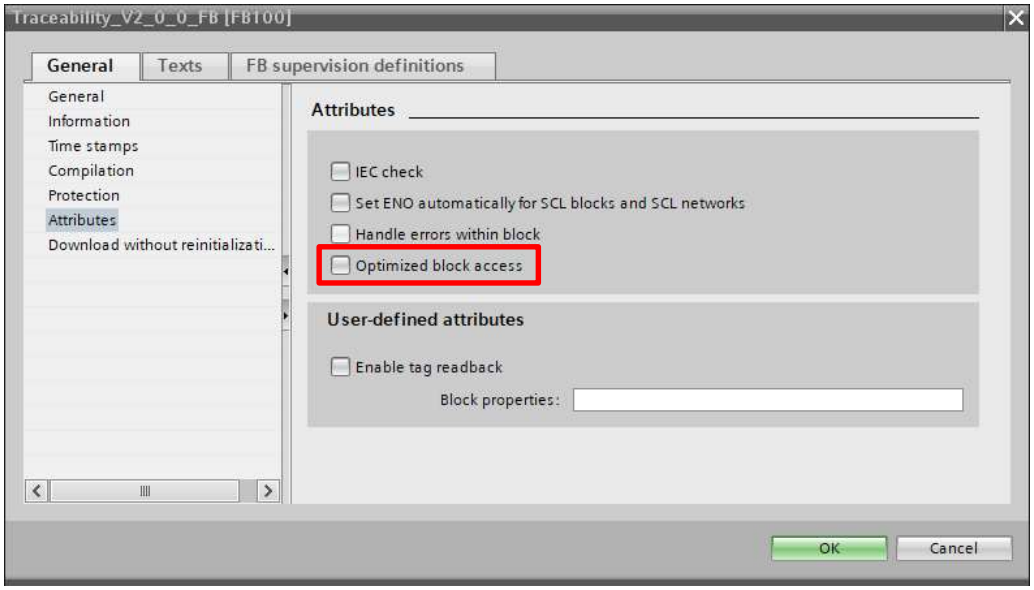
3. Configure the General settings of the function block.



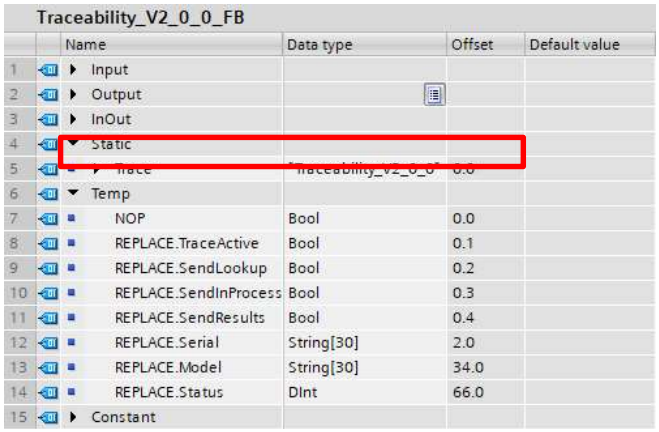
   a. If more than one traceability function block is required to handle multiple parts during the machine cycle, it is suggested to give each function block a meaningful name such as Traceability_Housing_FB or Traceability_Pinion_FB to describe the function that is being provided by each.

   b. The function block number can be set to Manual or Automatic per application.

4. Disable the option Optimized block access on the Attributes tab. This is required for the traceability software to access the tags by their block reference.



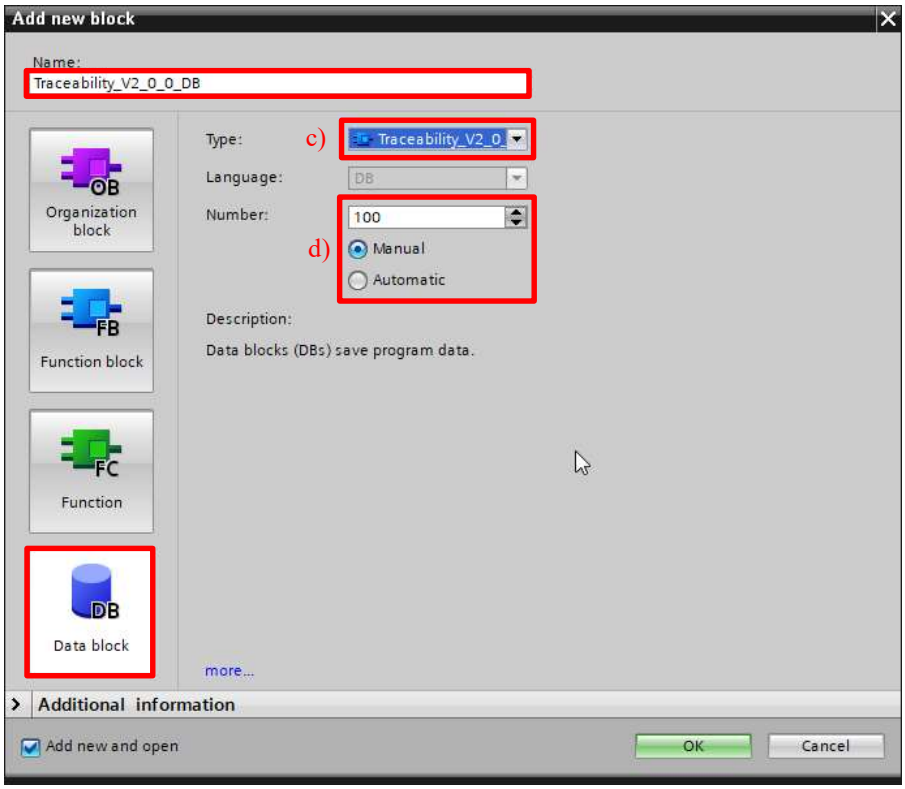
5. The Function Block contains a Static tag named Trace that is based on the UDT Traceability_V2_0_0 and must be the first tag in the list with an Offset of (0.0). There are also several Temp tags that are intended to be replaced or ignored in the logic as explained in section 2.5. Verify that tag references with UDT without errors.



2.3.3 The traceability application cannot access the tags inside the Function block directly, so a Data Block must be created to expose the tags to external connections.

1. Double click Add new block to create a new Data Block.

2. Edit the Data Block settings.



a. Enter a name for the Data Block with the same format that was used with the Function Block but ending with _DB instead of _FB.

b. Set the type of block to Data block.

c. Set the Type setting to reference the Function Block.

d. Set the Data Block Number setting to Manual and enter a value for the number. It is suggested to set the Data Block number to the same as the value use for the Function Block.

2.4 TIA Portal 15 - Process Data Configuration

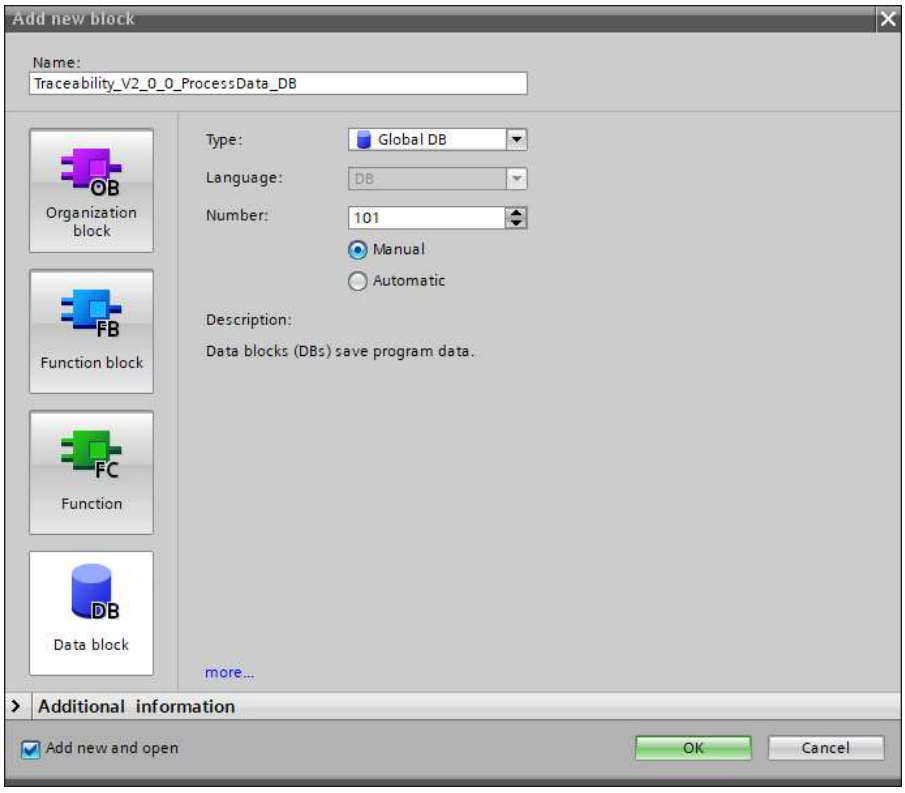
The Siemens traceability logic does not provide a predetermined set of tags that are used to store and read process data when sending results at the end of the machine cycle. Instead, the tags must be manually configured in the PLC and mapped to a database column in the application settings. The traceability application can read several types of PLC tags and converts them to string automatically when inserting to the database.

Note: If process data does not need to be read from the PLC and recorded in the database when sending part status results, skip to section 2.5.

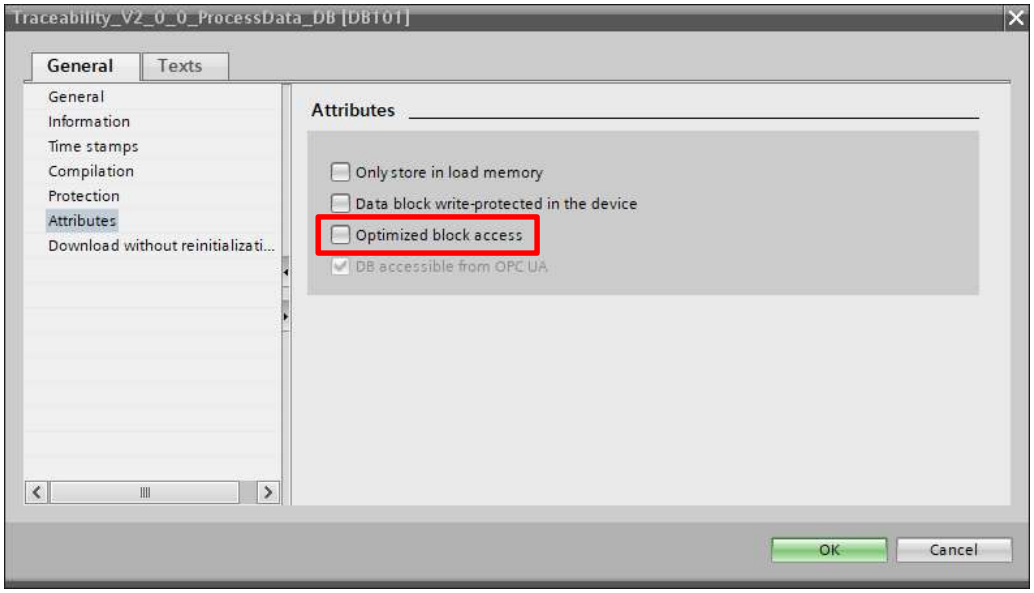
2.4.1 PLC Tag Configuration

It is recommended to add process data tags to a separate Data Block. This will make it easier to configure the tag addresses in the traceability application.

1. Create a new Data Block that will contain the process data.



2. Disable the option Optimized block access on the Attributes tab of the Data Block properties.



3. Create tags within the Data Block for each type of process data that will be captured at the end of the machine cycle. The following tag types are supported.

   - BOOL
   - BYTE
   - DINT
   - DWORD
   - INT
   - LINT (S7-1500 only)
   - LREAL (S7-1500 only)
   - LWORD (S7-1500 only)
   - REAL
   - SBYTE
   - STRING
   - ULINT
   - WORD

4. The following is an example process data tag configuration and will be referenced in section 2.4.3 when demonstrating how to configure the traceability application.

Traceability_V2_0_0_ProcessData_DB

	Name	Data type	Offset
1	Static		
2	Torque	Int	0.0
3	Angle	Real	2.0
4	Distance	DInt	6.0
5	BearingLot	String[20]	10.0

The tag address offset values will be used when referencing the process data tags in the traceability application configuration.

2.4.2 Tag Name Formats

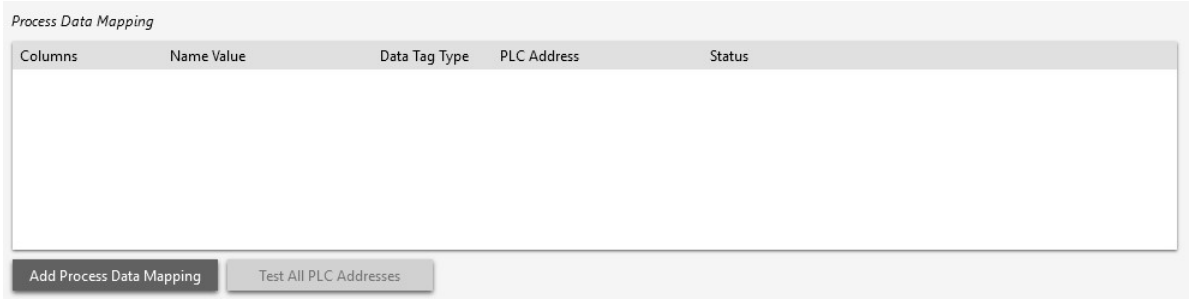
The traceability application references tags by the data block number that they reside within and the memory offset within the data block. The following table describes the tag name format that should be used for each type of PLC tag. The <mem address> portion of the tag name pertains the data block number. The <address> portion of the tag name indicates the memory offset within the data block of the tag. Using the example tags that were defined in the previous section, the INT Torque tag name would be DB101.DBW0 .

String tags must also have a "#" character and the string <length> appended to the tag name for the traceability application to read them, so the STRING BearingLot tag in the same example would be DB101.DBS10#20 .

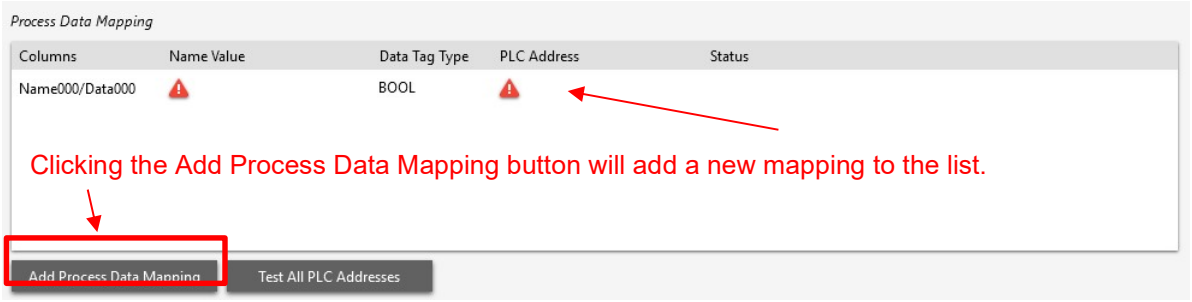
Tag Type	Tag Name Format
BOOL	DB.<mem address>.DBX<address>.<bit>
BYTE	DB.<mem address>.DBB<address>
SBYTE	DB.<mem address>.DBB<address>
INT	DB.<mem address>.DBW<address>
WORD	DB.<mem address>.DBW<address>
DINT	DB.<mem address>.DBD<address>
DWORD	DB.<mem address>.DBD<address>
REAL	DB.<mem address>.DBD<address>
LINT	DB.<mem address>.DBL<address>
ULINT	DB.<mem address>.DBL<address>
LREAL	DB.<mem address>.DBL<address>
LWORD	DB.<mem address>.DBL<address>
STRING	DB.<mem address>.DBS<address>#<length>

2.4.3 Traceability Application Settings Configuration

Open the device settings in the traceability application to display the process data mapping settings panel. Each process data tag shall be mapped to a corresponding database column using the mapping editor.

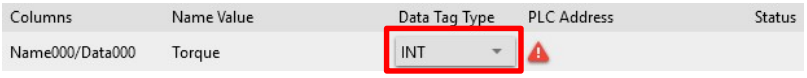


1. Click the Add Process Data Mapping button to add a new mapping definition. The NameXXX and DataXXX column pairs in the database are both configured for each mapping.

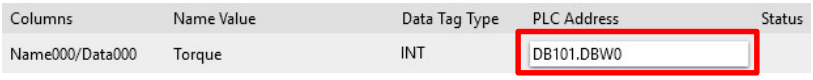


2. Enter a value for the Name Value which will be inserted into the NameXXX column. This value describes the process data that is inserted into the corresponding DataXXX column.

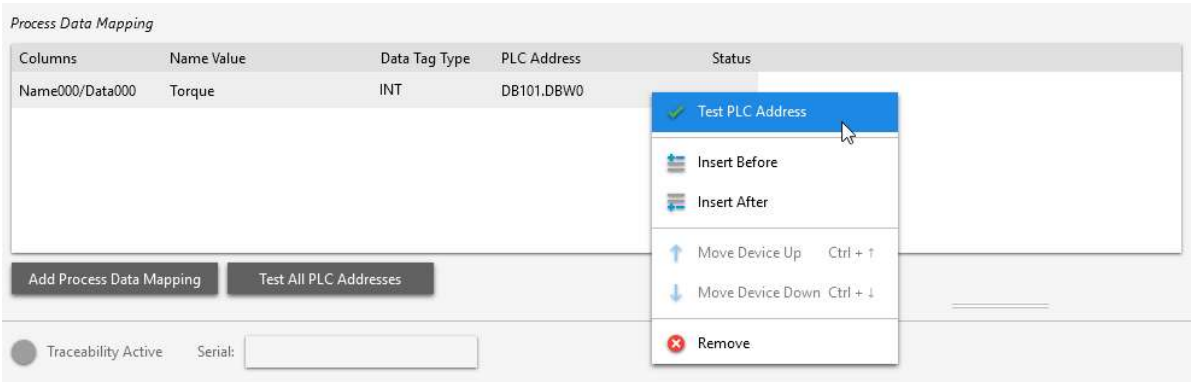
3. Select the type of tag that is being read from the PLC.



4. Enter the tag name using the format described in section 0.



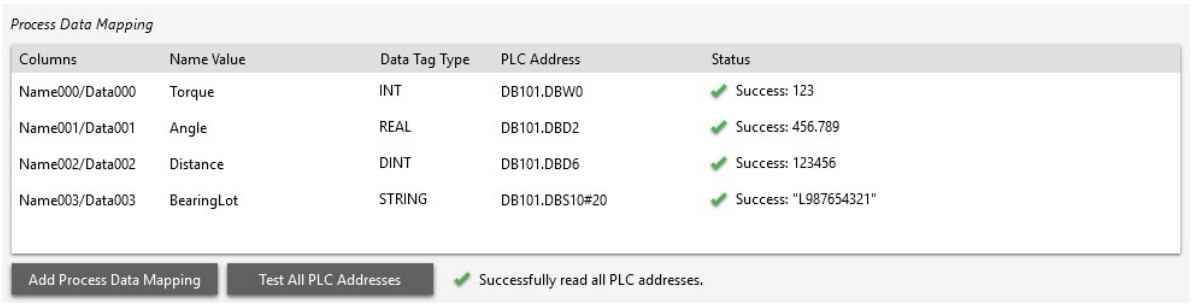
5. The tag can be tested by right clicking on the mapping and choosing Test PLC Address .



The Status column will show a green indicator with the tag value, or a red indicator with the error message.

The Test All PLC Addresses button can will attempt to read all PLC tags and display an overall status message.

6. The following example shows a successful test of all the example process data tags. When connecting to the PLC the software will automatically perform this full tag test and will not allow the connection to proceed if any tag configuration errors are present.

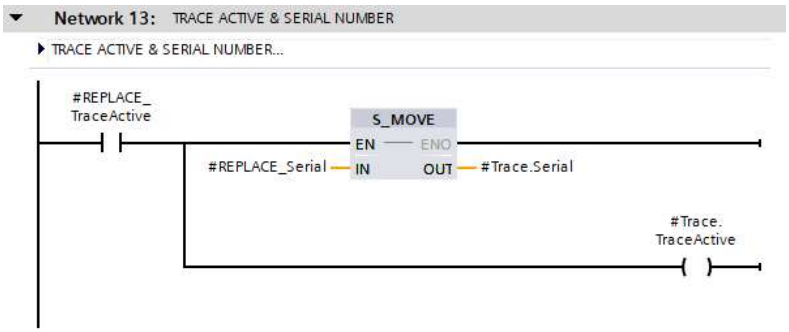


2.5 TIA Portal 15 - Logic Configuration

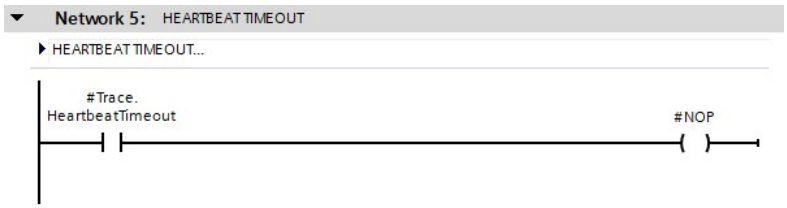
2.5.1 General Configuration

1. The scanned serial number shall be copied to the #Trace.Serial tag prior to turning on the #Trace.TraceActive coil instruction. The #Trace.TraceActive coil instruction enables the traceability functions. The logic that controls the #Trace.TraceActive coil instruction shall:
   a. Remain enabled throughout the entire part processing sequence and until all traceability operations are complete. Do not use a part present or sensor signal that may transition during the machine in cycle.
   b. Turn off between cycles.

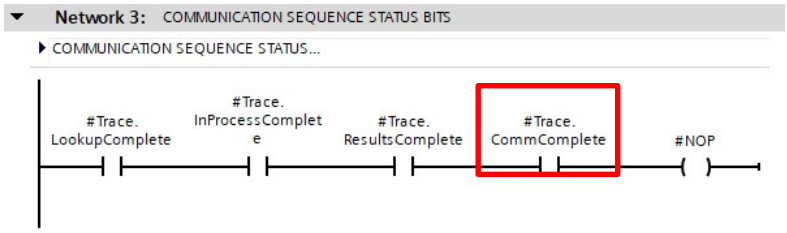
   Note: Disabling the #Trace.TraceActive tag will abort the Trace function requests and clears all Trace tag data.



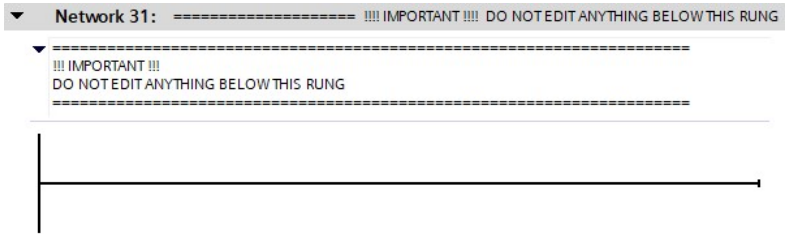
2. The #Trace.HeartbeatTimeout tag is used to signal a loss of communication with the Trace PC and shall be used to inhibit machine cycling. A loss of communication should prohibit the next cycle from initiating.



3. The #Trace.CommComplete tag is used to signal that all enabled traceability functions are complete and can be used as a sequence complete condition.



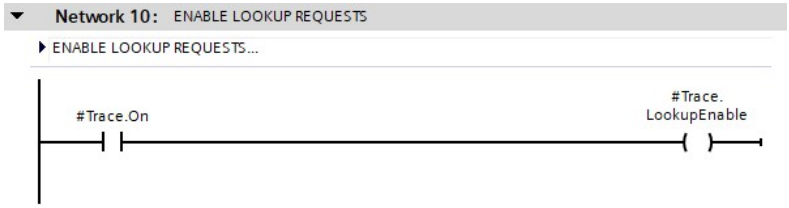
4. The logic shall not be edited below the phrase "DO NOT EDIT ANYTHING BELOW THIS RUNG."



2.5.2 Lookup Request (Permission to Run)

The Lookup Request is used to request information from the traceability application (SQL database) on a specific part serial number.

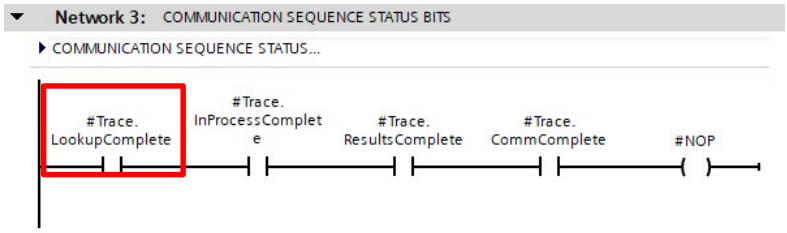
1. Enable the #Trace.LookupEnable coil instruction.



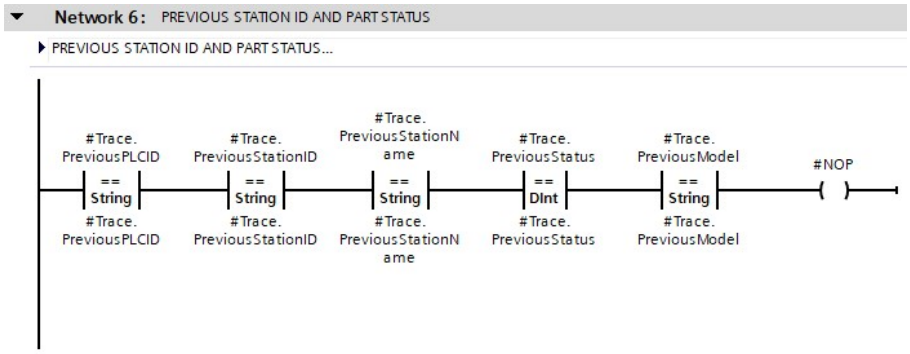
2. The Lookup Function is set by traceability application. Refer to section 2.1.55.
3. The #Trace.LookupSend coil initiates the Lookup Request with the Nexteer traceability application.

4. For Function (2) and (4) lookup requests the Station Lookup IDs are set by the traceability application. Reference section 2.1.54.

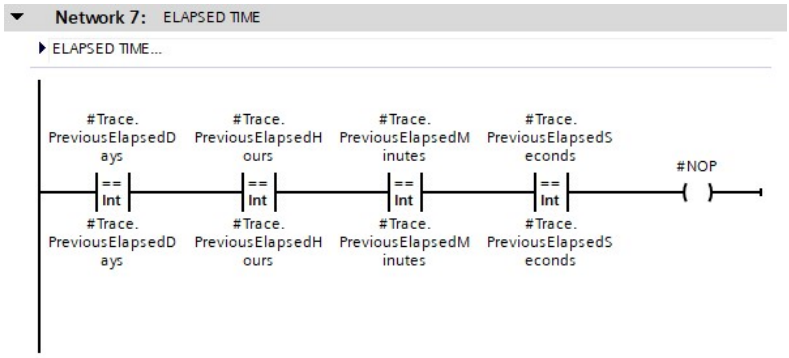
5. All data has been returned to the PLC when the #Trace.LookupComplete coil is turned on.



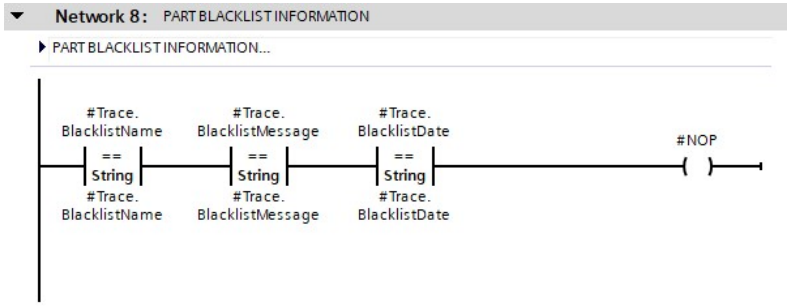
6. Previous station information.
   a. The #Trace.PreviousPLCID tag contains the machine identifier of the previous machine.
   b. The #Trace.PreviousStationID tag contains the station identifier of the previous station. This tag is typically used for comparison against of list of expected StationIDs to ensure that the part was processed at a valid station previously.
   c. The #Trace.PreviousStationName tag contains the human readable description of the previous station.
   d. The #Trace.PreviousStatus tag contains the part status result from the previous station. This tag is typically used for comparison against of list of expected statuses to ensure that the part can be run.
   e. The #Trace.PreviousModel tag contains the model that was configured when the part was run at the previous station.



7. The #Trace.PreviousElapsed(…) tags contain the calculated elapsed time from the time that the previous record was inserted into the database to the time that the lookup is performed. These values are typically used for part permissions when a part must be processed within a time window, or if a specified amount of time needs to pass before the part can be run.

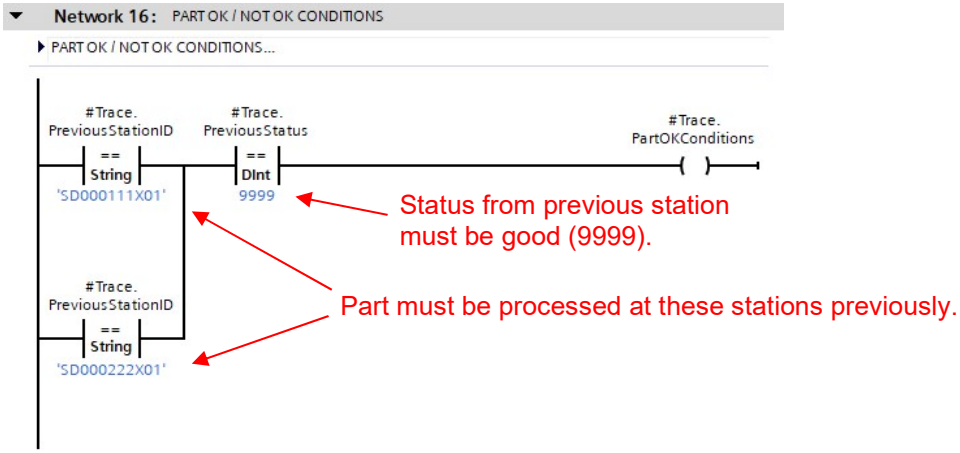


8. If the part is found to be blacklisted during a Lookup request, the #Trace.Blacklist(…) tags will contain detailed information about the blacklist.

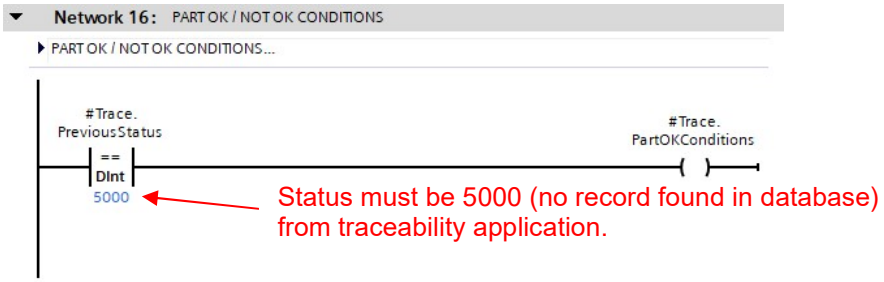


9. The logic shall be programmed to appropriately control the #Trace.PartOKConditions coil instruction for the application.

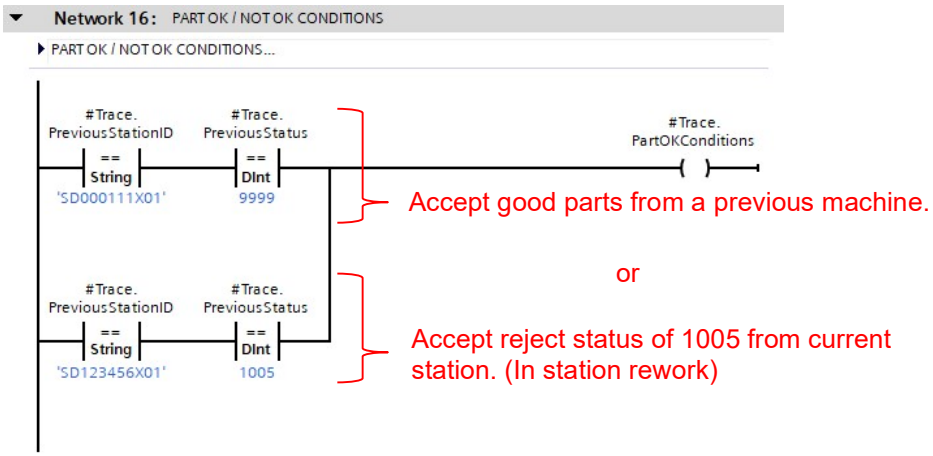
   a. Example: Expecting good parts from one of two previous stations.



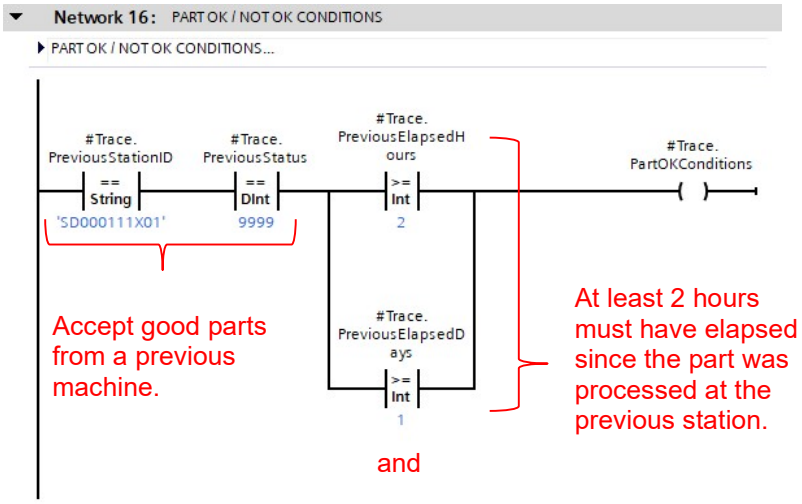
b. Example: Expecting no record found.



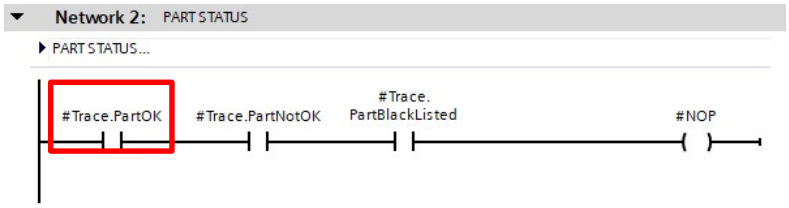
c. Example: Expecting good parts from previous station and rework of specific reject code in station.



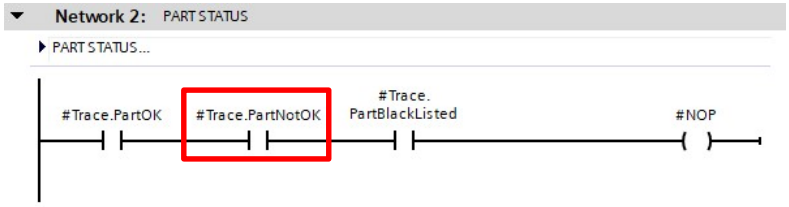
d. Example: Verification of elapsed time since part was processed at previous operation.



10. The #Trace.PartOK tag shall be used in the sequence routine to allow the sequence of the machine to continue processing the part and shall also be used to control the part status message display to indicate the part is OK to run.



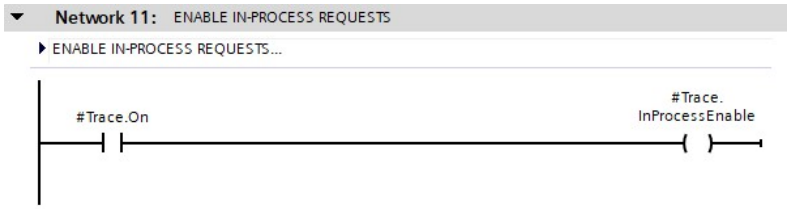
11. The #Trace.PartNotOK tag shall be used in the sequence routine to prevent processing the part and complete the sequence as needed and used to control a fault condition and the part status message display to indicate the part is not OK to run.



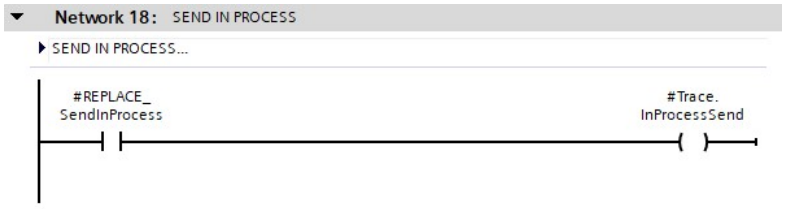
2.5.3 In Process Request (Indicate Part Started)

The In-Process Request sends a part status of 9000 to the SQL Database when enabled. This function is typically used to prohibit the reprocessing of parts if the cycle does not complete as expected, or to mark the part as consumed. It shall be enabled by a sequence step before the machine begins to alter the part.

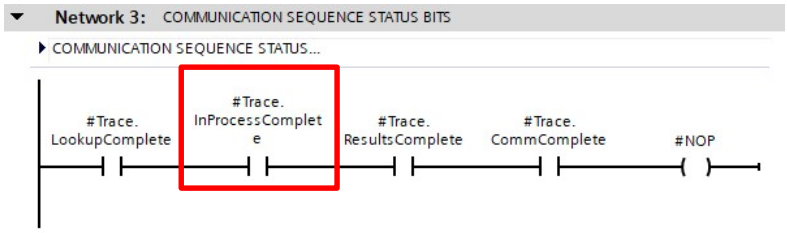
1. Enable the #Trace.InProcessEnable coil instruction.



2. The #Trace.InProcessSend coil instruction initiates the In-Process Request with the Nexteer traceability application.



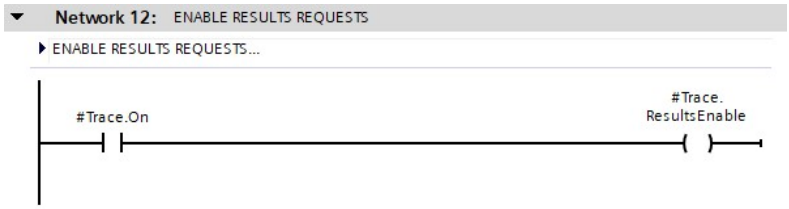
3. When the In-Process Request is complete, and a record has been inserted into the database the #Trace.InProcessComplete tag will be turned on.



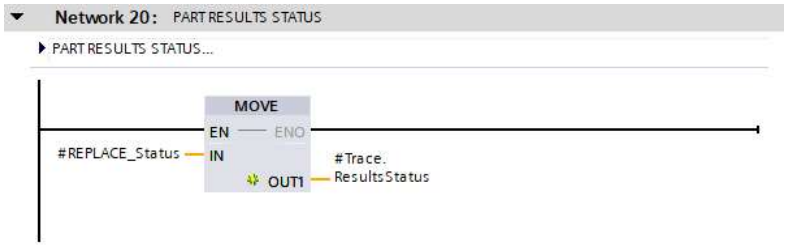
2.5.4 Results Request (Send Part Status and Process Data)

The Results Request is used to send the part status and process data to the traceability application (SQL database) on a specific part serial number typically when the part quality has been determined during the machine sequence.

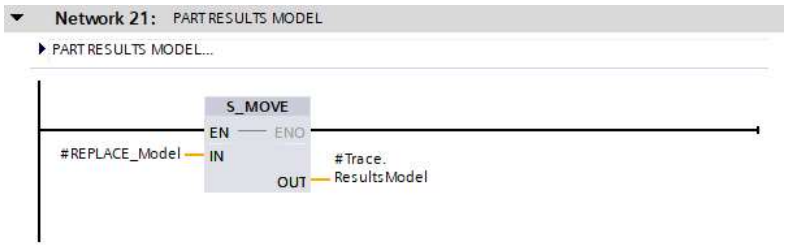
1. Enable the #Trace.ResultsEnable coil instruction.



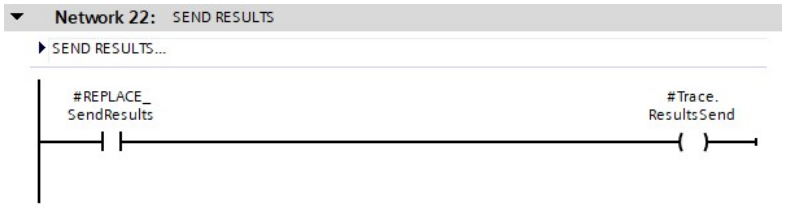
2. The Results Function is set by traceability application. Reference section 2.1.56.
3. The #Trace.ResultsStatus tag shall be loaded with the quality status of the processed part.



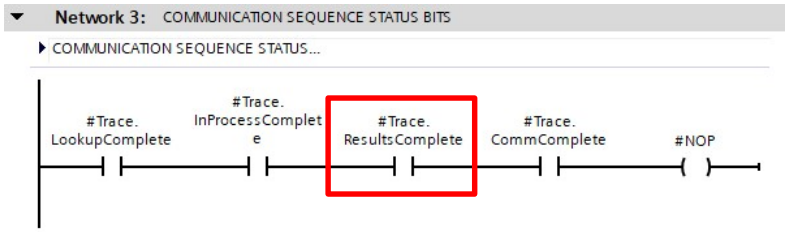
4. The current running model description shall be copied into the #Trace.ResultsModel tag. The Plant should provide a standardized list of model descriptions. These descriptions should be referenced on a manufacturing sequence chart. Model descriptions should be descriptive and be consistent from one station to another.



5. The #Trace.ResultsSend coil instruction initiates the Results Request with the Nexteer traceability application.



6. When the Results Request is complete, and a record has been inserted into the database the #Trace.ResultsComplete tag will be turned on.



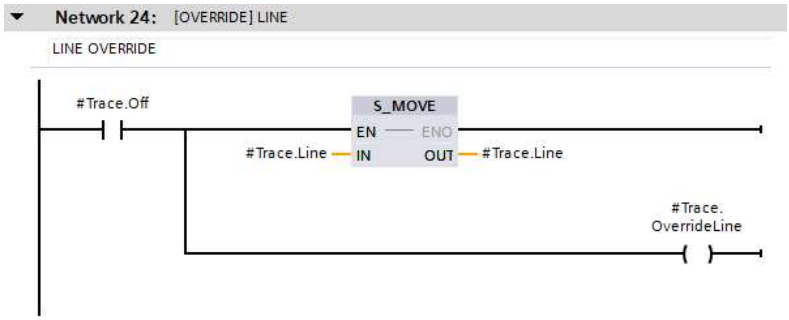
2.5.5 Overriding Application Settings

In some situations, it may be necessary for the PLC to override settings that are configured in the traceability application due to varying process requirements. For example, if multiple models that have different traceability requirements can be run on the machine, it may be necessary to change the StationID or Lookup Function depending on which model is currently set to run.

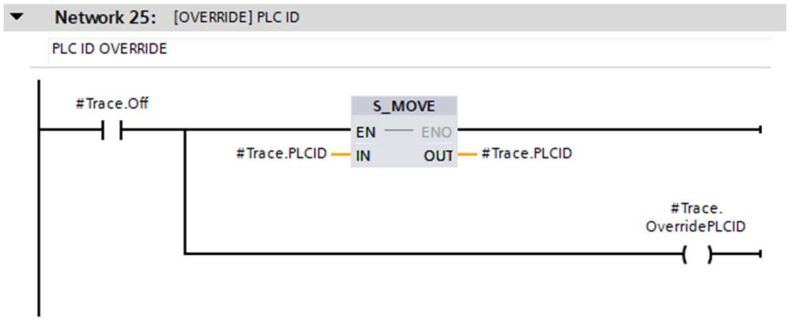
Note: Overriding traceability application settings is not a normal process function. Consult with Central Manufacturing IT before editing the following rungs.

1. The #Trace.Line setting can be overridden to change which “line” database to use for Lookup and Results requests if the traceability application is configured to use the multiple database system.

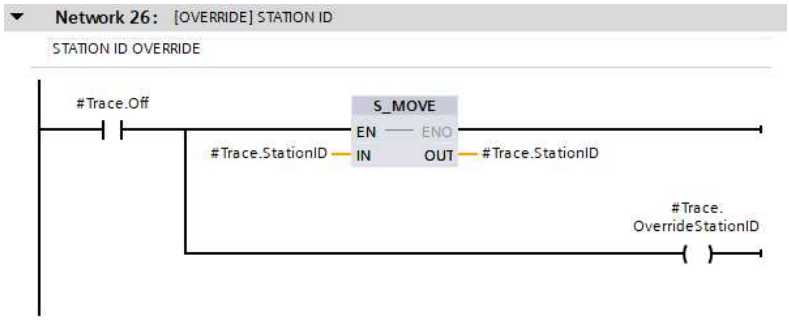
   Note: Not implemented (future functionality).



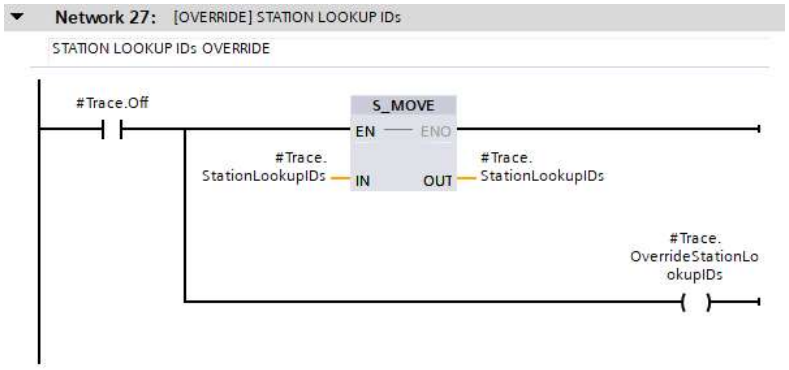
2. The #Trace.PLCID setting can be overridden to change the identity of the machine.



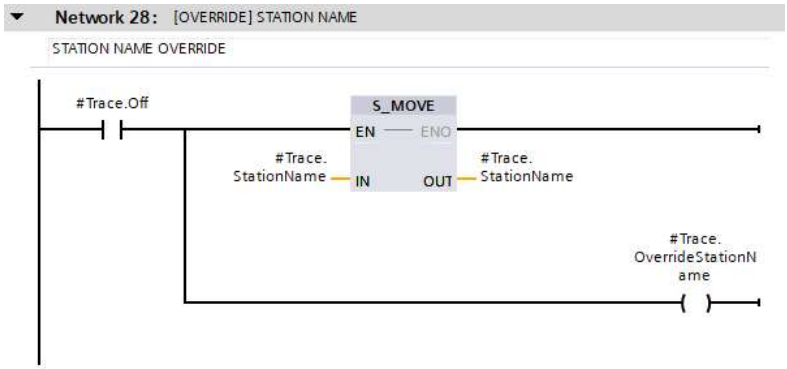
3. The #Trace.StationID setting can be overridden to change the identity of the station, which is typically used to affect permissions downstream, or if multiple models can be run on the machine that have different amounts of process data and need to be separated logically in the database.



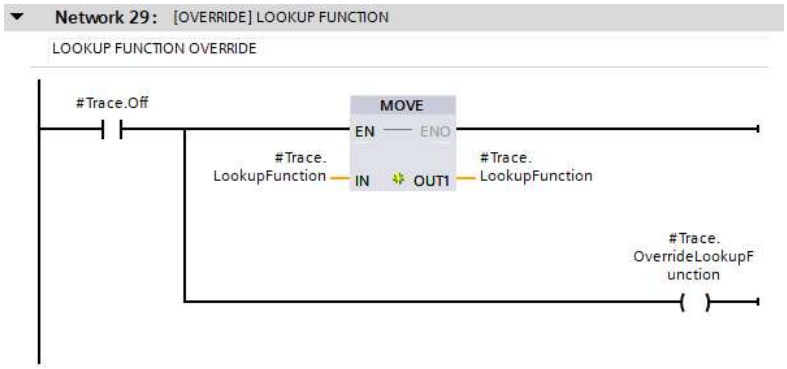
4. The #Trace.StationLookupIDs setting can be overridden to change which StationIDs to search for in the database when performing a function (2) or function (4) lookup.



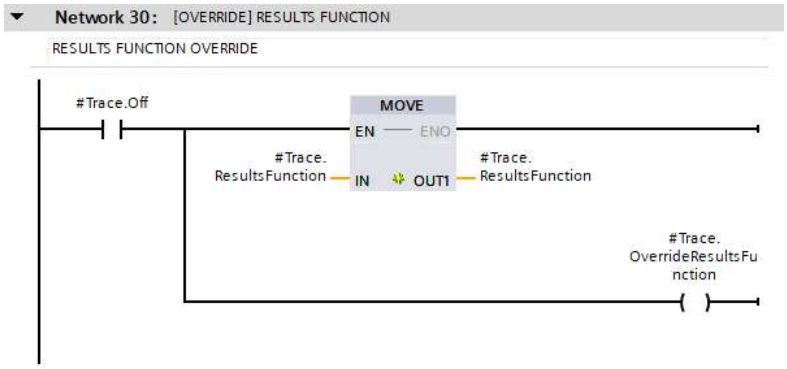
5. The #Trace.StationName setting can be overridden to change the human readable description of the station.



6. The #Trace.Lookup setting can be overridden to change which lookup function is used during a Lookup request.



7. The #Trace.Results setting can be overridden to change which results function is used during a Results request.



2.6 Simatic Step 7 - Logic Structure

The Siemens traceability logic is comprised of a LAD Function Block and a corresponding Data Block to expose the Function Block's tags to the traceability application. The Function Block contains a specific tag structure that must not be altered. See section 2.9 for a detailed tag structure reference. The logic networks are required to be modified to meet the requirements of the application. See section 2.10 for a detailed timing chart showing logic and traceability application communication.

Object name	Symbolic name	Created in language	Size in the work me...	Type
System data	---	---	---	SDB
OB1		LAD	72	Organization Block
FB100	Traceability_V2_0_0_FB	LAD	4056	Function Block
FC4	DELETE	STL	414	Function
FC10	EQ_STRNG	STL	152	Function
FC21	LEN	STL	76	Function
DB100	Traceability_V2_0_0_DB	DB	814	Instance data block ...
SFB4	TON	STL	---	System function block
SFC20	BLKMOV	STL	---	System function
SFC21	FILL	STL	---	System function

A few supporting library function blocks are required for the traceability logic to function. Functions FC4 (DELETE) , FC10 (EQ_STRNG) , FC21 (LEN) , SFB4 (TON) , SFC20 (BLKMOV) , and SFC21 (FILL) must be downloaded to the PLC prior to imported to the PLC before importing the traceability function block.

Object name	Symbolic name	Created in language	Size in the work me...	Type
System data	---	---	---	SDB
OB1		LAD	72	Organization Block
FB100	Traceability_V2_0_0_FB	LAD	4056	Function Block
FC4	DELETE	STL	414	Function
FC10	EQ_STRNG	STL	152	Function
FC21	LEN	STL	76	Function
DB100	Traceability_V2_0_0_DB	DB	814	Instance data block ...
SFB4	TON	STL	---	System function block
SFC20	BLKMOV	STL	---	System function
SFC21	FILL	STL	---	System function

2.6.1

The LAD function block Traceability_V2_0_0_FB contains the logic to perform traceability operations for a single part during the cycle. The function block can be copied from the template to the target programming by dragging & dropping. A unique copy of the function block is required for each part that requires traceability functions to be performed during the cycle.

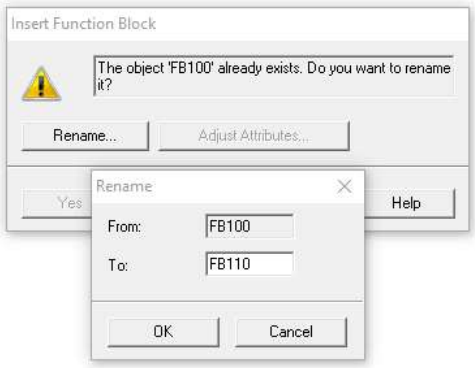
1. Copy the supporting function blocks to the program and download them to the PLC.

Object name	Symbolic name	Created in language	Size in the work me...	Type
System data	---	---	---	SDB
OB1			38	Organization Block
FC4	DELETE	STL	414	Function
FC10	EQ_STRNG	STL	152	Function
FC21	LEN	STL	76	Function
SFB4	TON	STL	---	System function block
SFC20	BLKMOV	STL	---	System function
SFC21	FILL	STL	---	System function

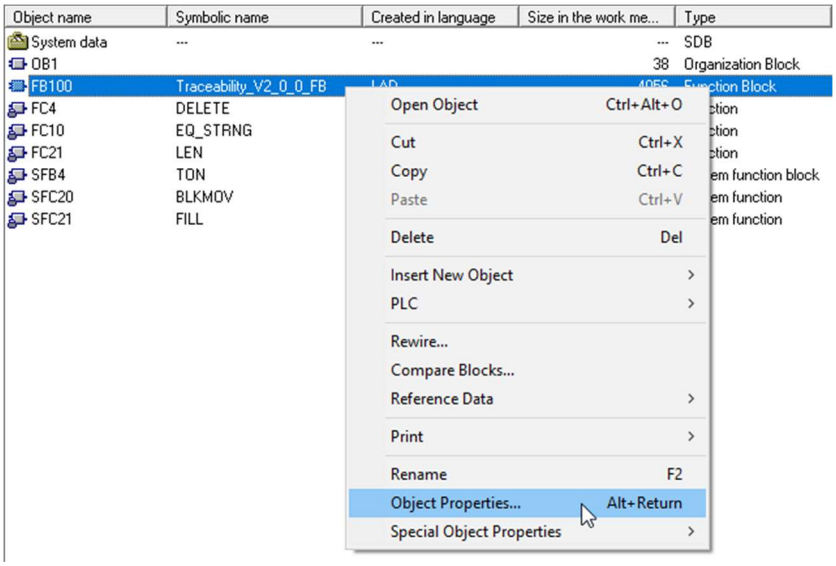
2. Copy the traceability Function Block into the program and download to the PLC.

Object name	Symbolic name	Created in language	Size in the work me...	Type
System data	---	---	---	SDB
OB1			38	Organization Block
FB100	Traceability_V2_0_0_FB	LAD	4056	Function Block
FC4	DELETE	STL	414	Function
FC10	EQ_STRNG	STL	152	Function
FC21	LEN	STL	76	Function
SFB4	TON	STL	---	System function block
SFC20	BLKMOV	STL	---	System function
SFC21	FILL	STL	---	System function

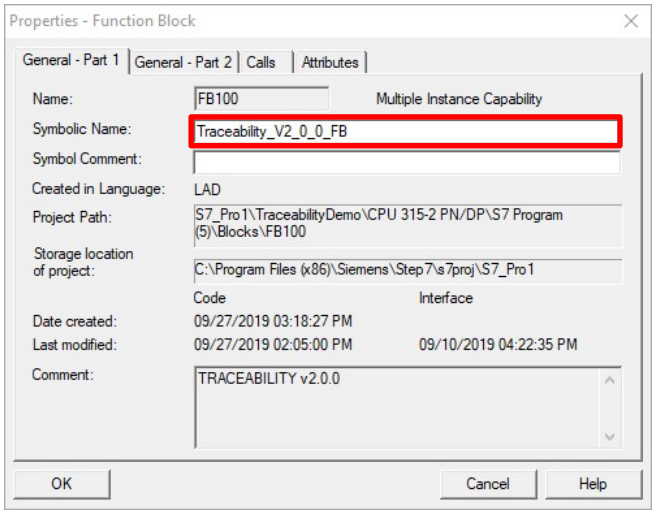
3. If multiple traceability Function Blocks are required by the application or a conflict occurs when copying, simply rename the block with a unique number.



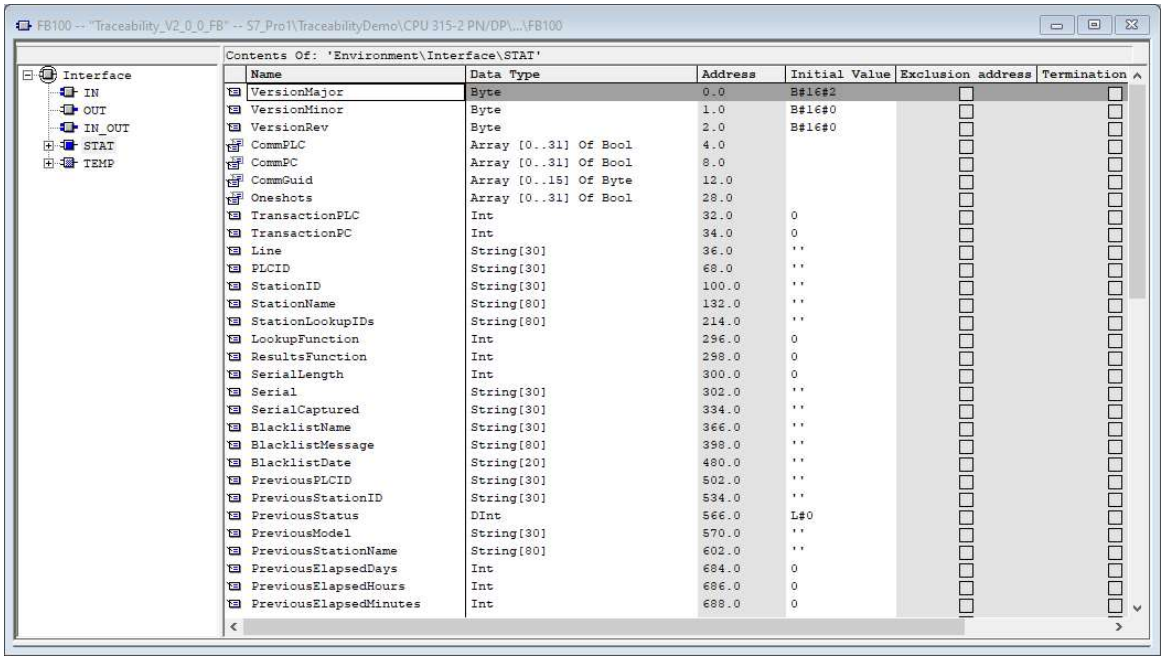
4. Right click on the imported function block and open Properties .



5. Configure the General settings of the function block. If more than one traceability function block is required to handle multiple parts during the machine cycle, it is suggested to give each function block a meaningful name such as Traceability_Housing_FB or Traceability_Pinion_FB to describe the function that is being provided by each.

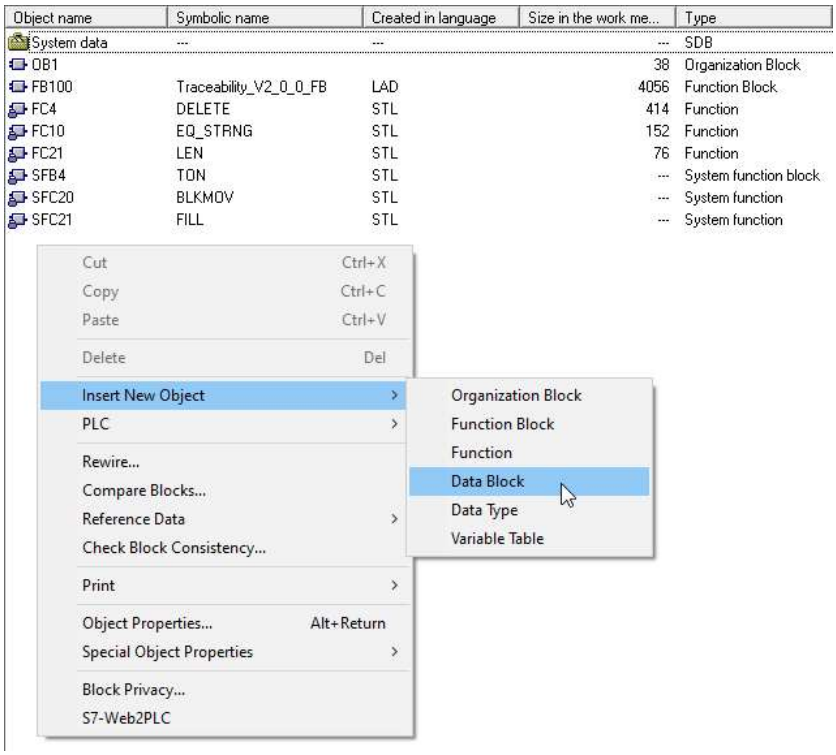


6. The Function Block contains a set of static tags that must not be altered.

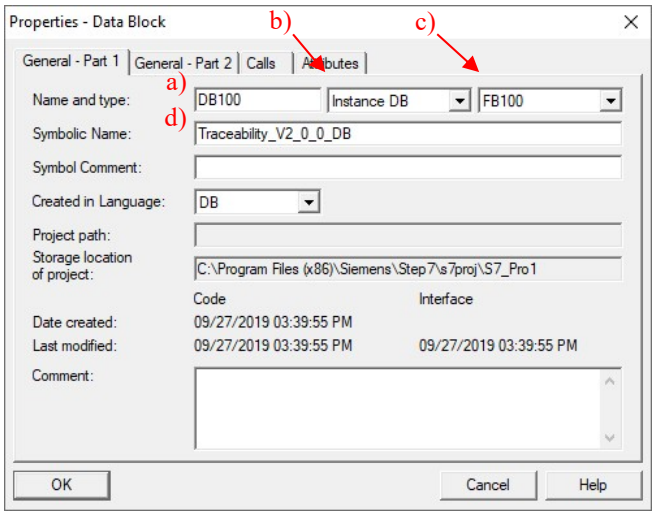


2.6.2 The traceability application cannot access the tags inside the Function block directly, so a Data Block must be created to expose the tags to external connections.

1. Right click and chose Insert New Object > Data Block using the context menu.



2. Edit the Data Block settings.



a. Set the Data Block number to the same value as the Function Block.

b. Set the type to Instance DB .

c. Set the instance reference to the Function Block.

d. Enter a name for the Data Block with the same format that was used with the Function Block but ending with _DB instead of _FB.

2.7 Simatic Step 7 - Process Data Configuration

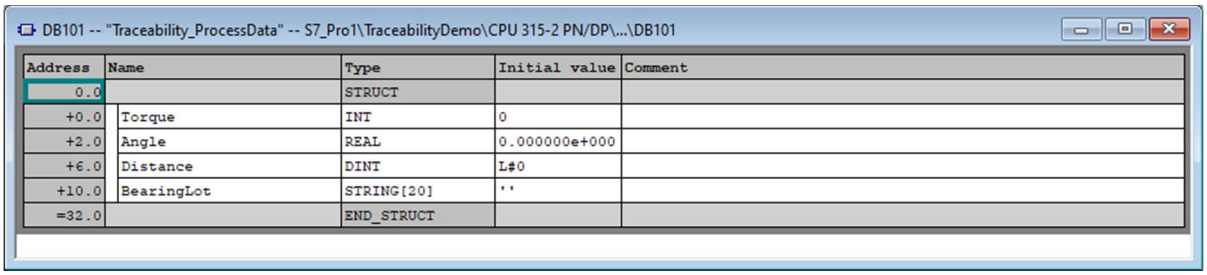
The Siemens traceability logic does not provide a predetermined set of tags that are used to store and read process data when sending results at the end of the machine cycle. Instead, the tags must be manually configured in the PLC and mapped to a database column in the application settings. The traceability application can read several types of PLC tags and converts them to string automatically when inserting to the database.

Note: If process data does not need to be read from the PLC and recorded in the database when sending part status results, skip to section 2.8.

2.7.1 PLC Tag Configuration

It is recommended to add process data tags to a separate Data Block. This will make it easier to configure the tag addresses in the traceability application.

1. Create a new Data Block that will contain the process data.

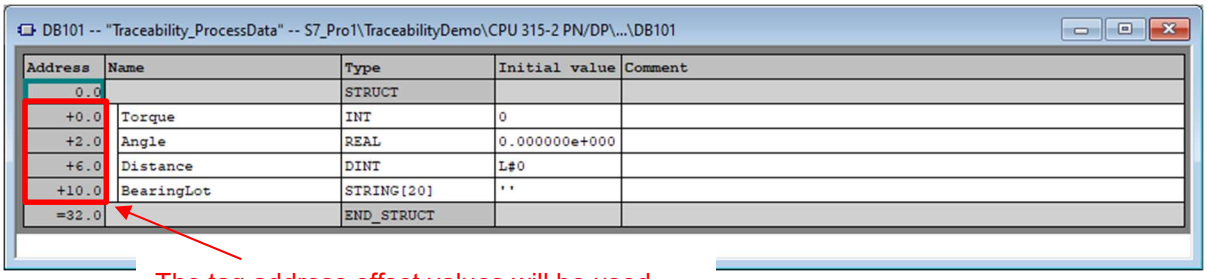


Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Torque	INT	0	
+2.0	Angle	REAL	0.000000e+000	
+6.0	Distance	DINT	L#0	
+10.0	BearingLot	STRING[20]	''	
=32.0		END_STRUCT		

2. Create tags within the Data Block for each type of process data that will be captured at the end of the machine cycle. The following tag types are supported.

- BOOL
- BYTE
- DINT
- DWORD
- INT
- LINT (S7-1500 only)
- LREAL (S7-1500 only)
- LWORD (S7-1500 only)
- REAL
- SBYTE
- STRING
- ULINT
- WORD

3. The following is an example process data tag configuration and will be referenced in section 2.7.3 when demonstrating how to configure the traceability application.



2.7.2 Tag Name Formats

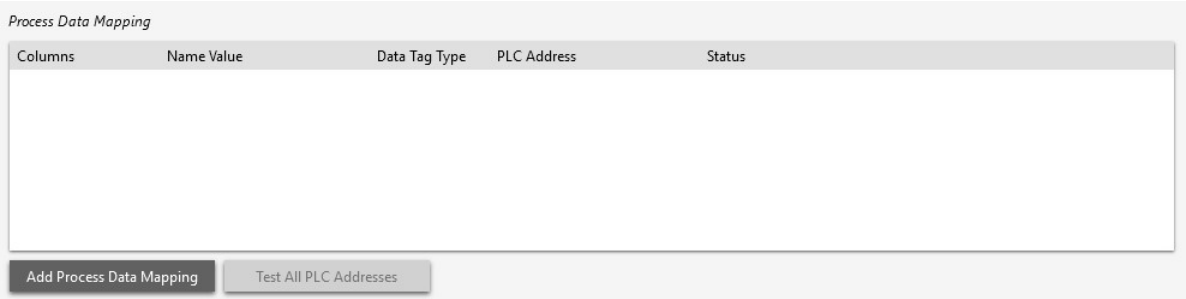
The traceability application references tags by the data block number that they reside within and the memory offset within the data block. The following table describes the tag name format that should be used for each type of PLC tag. The <mem address> portion of the tag name pertains the data block number. The <address> portion of the tag name indicates the memory offset within the data block of the tag. Using the example tags that were defined in the previous section, the INT Torque tag name would be DB101.DBW0.

String tags must also have a "#" character and the string <length> appended to the tag name for the traceability application to read them, so the STRING BearingLot tag in the same example would be DB101.DBS10#20.

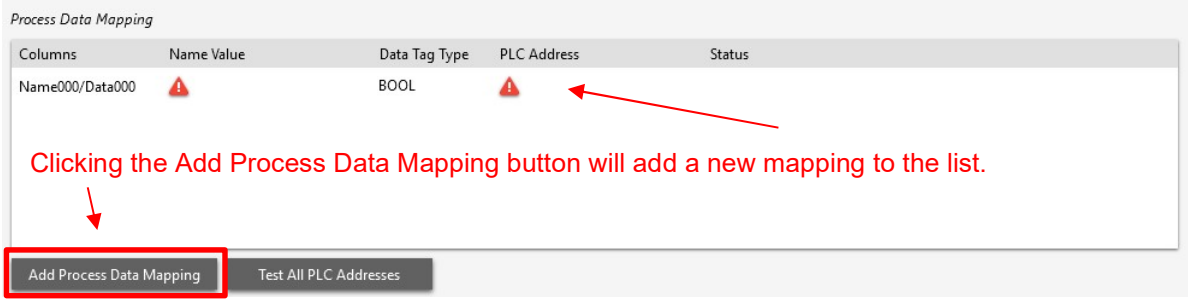
Tag Type	Tag Name Format
BOOL	DB.<mem address>.DBX<address>.<bit>
BYTE	DB.<mem address>.DBB<address>
SBYTE	DB.<mem address>.DBB<address>
INT	DB.<mem address>.DBW<address>
WORD	DB.<mem address>.DBW<address>
DINT	DB.<mem address>.DBD<address>
DWORD	DB.<mem address>.DBD<address>
REAL	DB.<mem address>.DBD<address>
LINT	DB.<mem address>.DBL<address>
ULINT	DB.<mem address>.DBL<address>
LREAL	DB.<mem address>.DBL<address>
LWORD	DB.<mem address>.DBL<address>
STRING	DB.<mem address>.DBS<address>#<length>

2.7.3 Traceability Application Settings Configuration

Open the device settings in the traceability application to display the process data mapping settings panel. Each process data tag shall be mapped to a corresponding database column using the mapping editor.

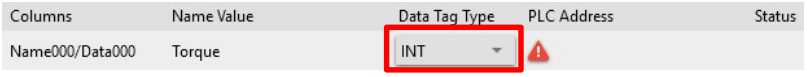


1. Click the Add Process Data Mapping button to add a new mapping definition. The NameXXX and DataXXX column pairs in the database are both configured for each mapping.

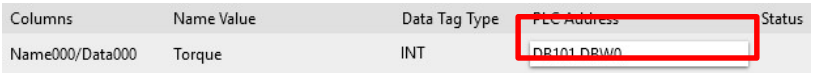


2. Enter a value for the Name Value which will be inserted into the NameXXX column. This value describes the process data that is inserted into the corresponding DataXXX column.

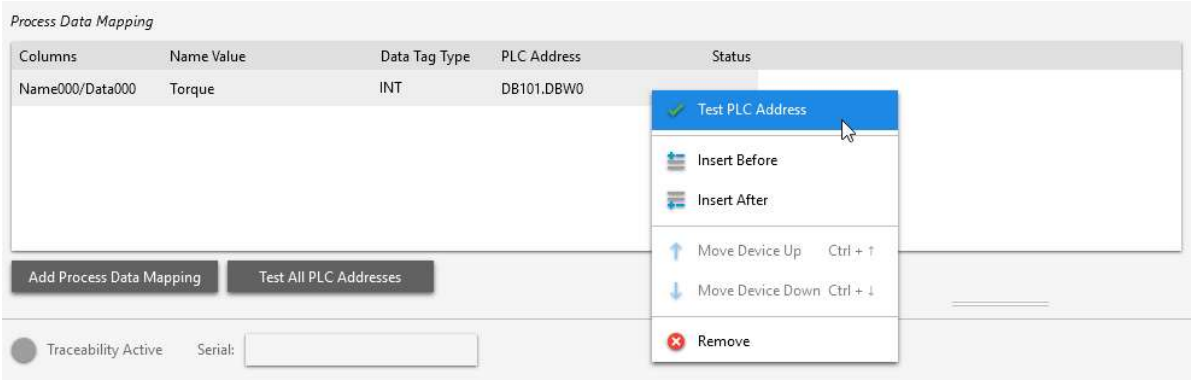
3. Select the type of tag that is being read from the PLC.



4. Enter the tag name using the format described in section 2.7.2.



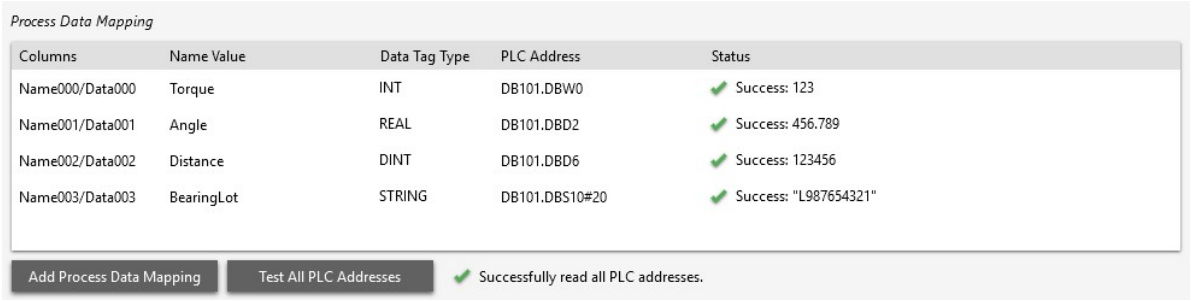
5. The tag can be tested by right clicking on the mapping and choosing Test PLC Address .



The Status column will show a green indicator with the tag value, or a red indicator with the error message.

The Test All PLC Addresses button can will attempt to read all PLC tags and display an overall status message.

6. The following example shows a successful test of all the example process data tags. When connecting to the PLC the software will automatically perform this full tag test and will not allow the connection to proceed if any tag configuration errors are present.



2.8 Simatic Step 7 - Logic Configuration

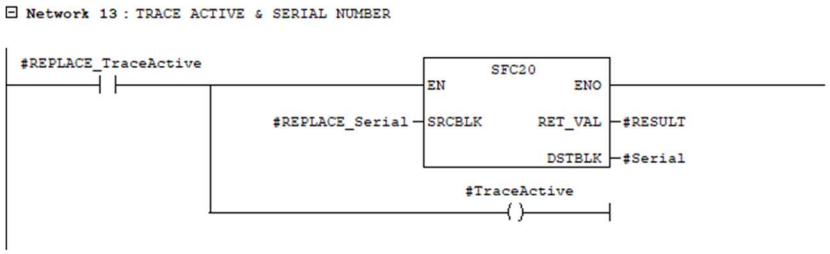
2.8.1 General Configuration

1. The scanned serial number shall be copied to the #Serial tag prior to turning on the #TraceActive coil instruction. The #TraceActive coil instruction enables the traceability functions. The logic that controls the #TraceActive coil instruction shall:

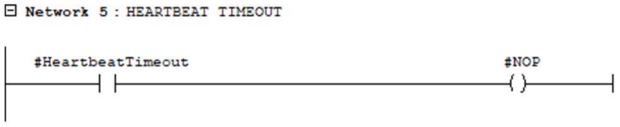
a. Remain enabled throughout the entire part processing sequence and until all traceability operations are complete. Do not use a part present or sensor signal that may transition during the machine in cycle.

b. Turn off between cycles.

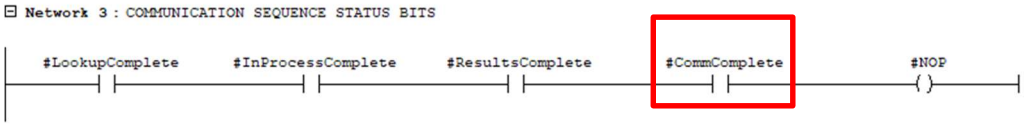
Note: Disabling the #TraceActive tag will abort the Trace function requests and clears all Trace tag data.



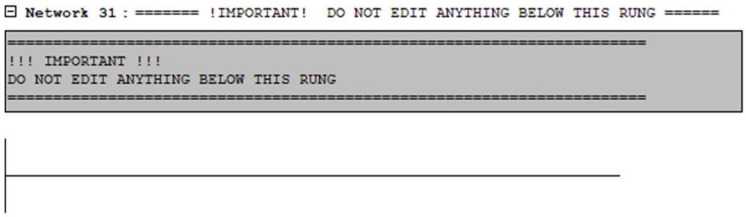
2. The #HeartbeatTimeout tag is used to signal a loss of communication with the Trace PC and shall be used to inhibit machine cycling. A loss of communication should prohibit the next cycle from initiating.



3. The #CommComplete tag is used to signal that all enabled traceability functions are complete and can be used as a sequence complete condition.



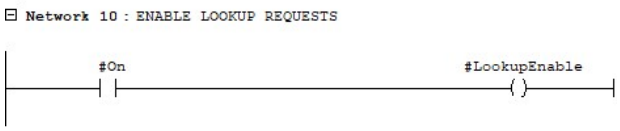
4. The logic shall not be edited below the phrase "DO NOT EDIT ANYTHING BELOW THIS RUNG."



2.8.2 Lookup Request (Permission to Run)

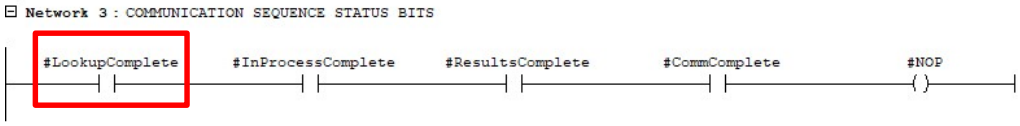
The Lookup Request is used to request information from the traceability application (SQL database) on a specific part serial number.

1. Enable the #LookupEnable coil instruction.



2. The Lookup Function is set by traceability application. Refer to section 2.1.55.
3. The #LookupSend coil instruction initiates the Lookup Request with the Nexteer traceability application.

4. For Function (2) and (4) lookup requests the Station Lookup IDs are set by the traceability application. Reference section 2.1.54.
5. All data has been returned to the PLC when the #LookupComplete coil is turned on.



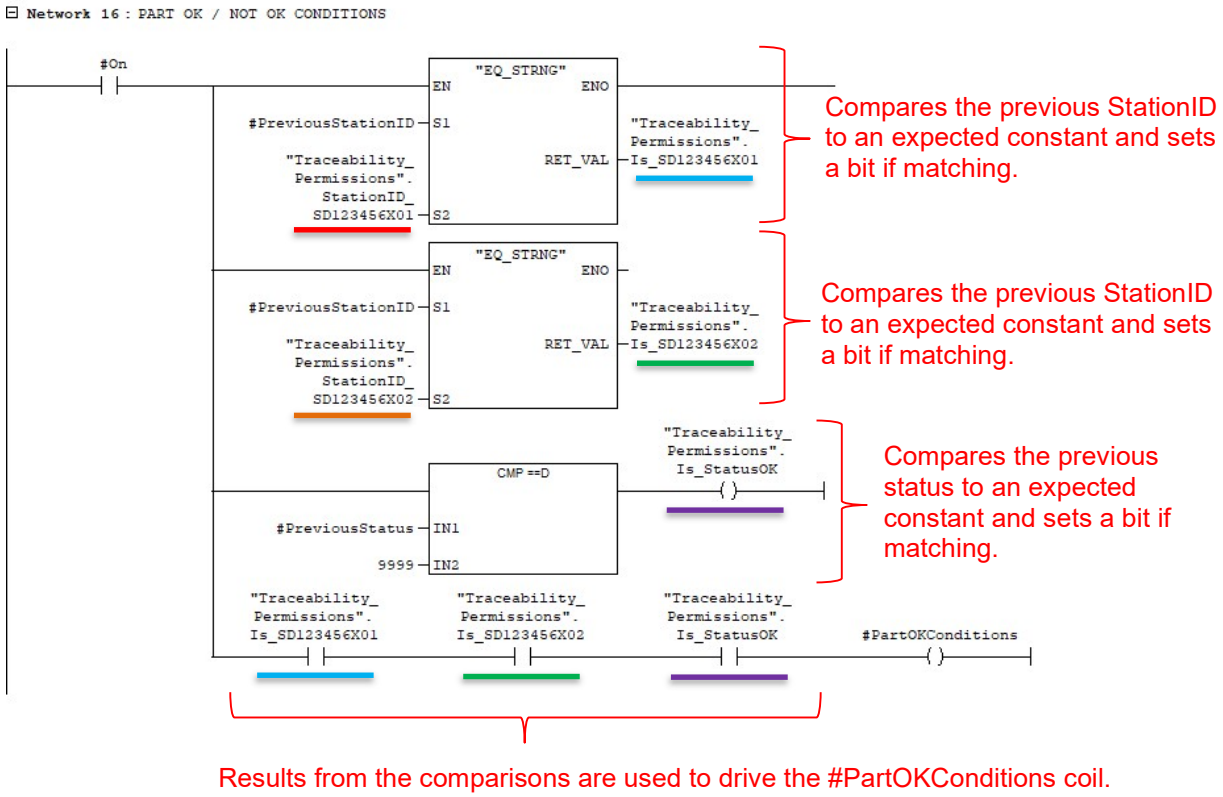
6. Previous station information. When the lookup request is complete, these tags will contain the data that was inserted to the database when the parts was run at the previous station. The previous station tags are intended to be compared to constants to determine the state of the #PartOKConditions bit as described below.
   a. The #PreviousPLCID tag contains the machine identifier of the previous machine.
   b. The #PreviousStationID tag contains the station identifier of the previous station. This tag is typically used for comparison against of list of expected StationIDs to ensure that the part was processed at a valid station previously.
   c. The #PreviousStationName tag contains the human readable description of the previous station.
   d. The #PreviousStatus tag contains the part status result from the previous station. This tag is typically used for comparison against of list of expected statuses to ensure that the part can be run.
   e. The #PreviousModel tag contains the model that was configured when the part was run at the previous station.
7. The #PreviousElapsed(…) tags contain the calculated elapsed time from the time that the previous record was inserted into the database to the time that the lookup is performed. These values are typically used for part permissions when a part must be processed within a time window, or if a specified amount of time needs to pass before the part can be run.

8. The logic shall be programmed to appropriately control the #PartOKConditions coil instruction for the application. The following is a simple example that shows how the logic may be configured to accept good parts (status 9999) from one of two previous upstream machines (StationID SD123456X01 or StationID SD123456X02).

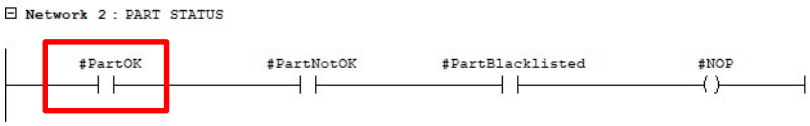
   a. It may be necessary to create an additional data block to store the constant data values used to compare against the previous station data.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	StationID_SD123456X01	STRING[30]	'SD123456X01'	Previous acceptable StationID.
+32.0	StationID_SD123456X02	STRING[30]	'SD123456X02'	Previous acceptable StationID.
+64.0	Is_SD123456X01	BOOL	FALSE	Boolean to set if previous StationID = "SD123456X01".
+64.1	Is_SD123456X02	BOOL	FALSE	Boolean to set if previous StationID = "SD123456X02".
+64.2	Is_StatusOK	BOOL	FALSE	Boolean to set if previous Status = 9999.
=66.0		END_STRUCT		

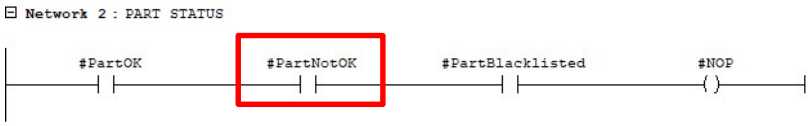
   b. The following is an example that shows how to compare the previous station data against the constants to control the #PartOKConditions coil.



9. The #PartOK tag shall be used in the sequence routine to allow the sequence of the machine to continue processing the part and shall also be used to control the part status message display to indicate the part is ok to run.



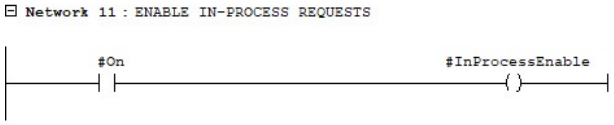
10. The #PartNotOK tag shall be used in the sequence routine to prevent processing the part and complete the sequence as needed and used to control a fault condition and the part status message display to indicate the part is not ok to run.



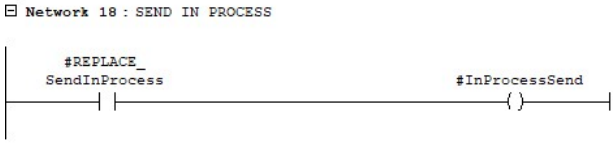
2.8.3 In Process Request (Indicate Part Started)

The In-Process Request sends a part status of 9000 to the SQL Database when enabled. This function is typically used to prohibit the reprocessing of parts if the cycle does not complete as expected, or to mark the part as consumed. It shall be enabled by a sequence step before the machine begins to alter the part.

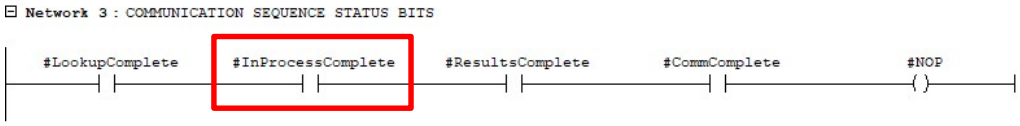
1. Enable the #InProcessEnable coil instruction.



2. The #InProcessSend coil instruction initiates the In-Process Request with the Nexteer traceability application.



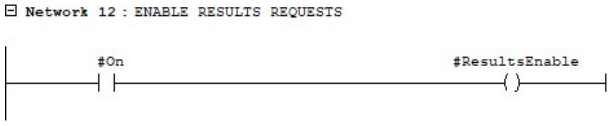
3. When the In-Process Request is complete, and a record has been inserted into the database the #InProcessComplete tag will be turned on.



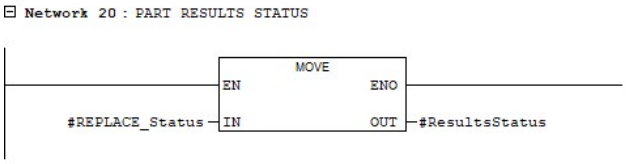
2.8.4 Results Request (Send Part Status and Process Data)

The Results Request is used to send the part status and process data to the traceability application (SQL database) on a specific part serial number typically when the part quality has been determined during the machine sequence.

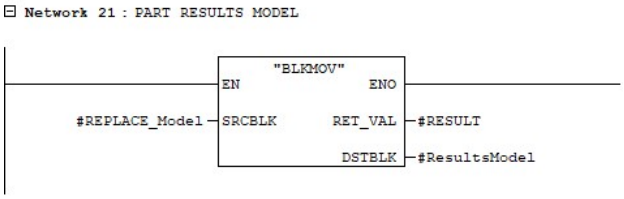
1. Enable the #ResultsEnable coil instruction.



2. The Results Function is set by traceability application. Reference section 2.1.56.
3. The #ResultsStatus tag shall be loaded with the quality status of the processed part.

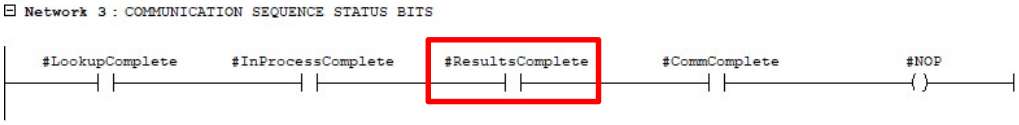


4. The current running model description shall be copied into the #ResultsModel tag. The Plant should provide a standardized list of model descriptions. These descriptions should be referenced on a manufacturing sequence chart. Model descriptions should be descriptive and be consistent from one station to another.



5. The #ResultsSend coil instruction initiates the Results Request with the Nexteer traceability application.

6. When the Results Request is complete, and a record has been inserted into the database the #ResultsComplete tag will be turned on.



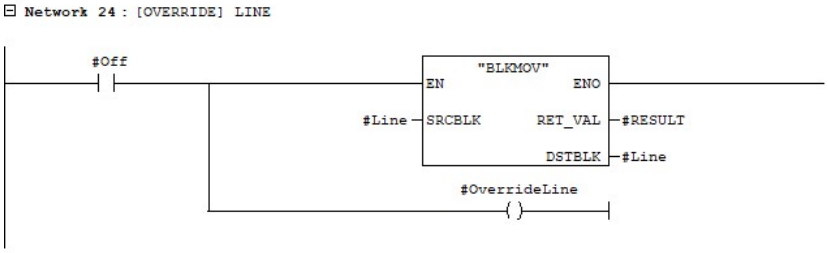
2.8.5 Overriding Application Settings

In some situations, it may be necessary for the PLC to override settings that are configured in the traceability application due to varying process requirements. For example, if multiple models that have different traceability requirements can be run on the machine, it may be necessary to change the StationID or Lookup Function depending on which model is currently set to run.

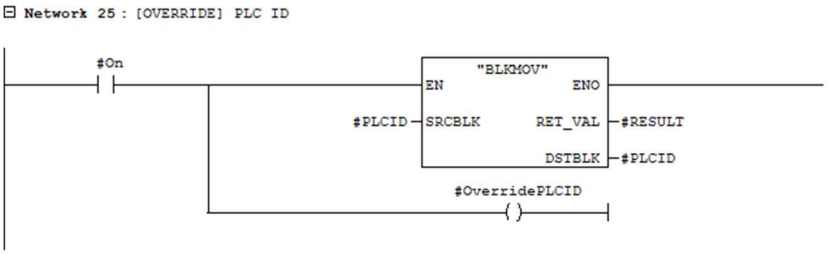
Note: Overriding traceability application settings is not a normal process function. Consult with Central Manufacturing IT before editing the following rungs.

1. The #Line setting can be overridden to change which "line" database to use for Lookup and Results requests if the traceability application is configured to use the multiple database system.

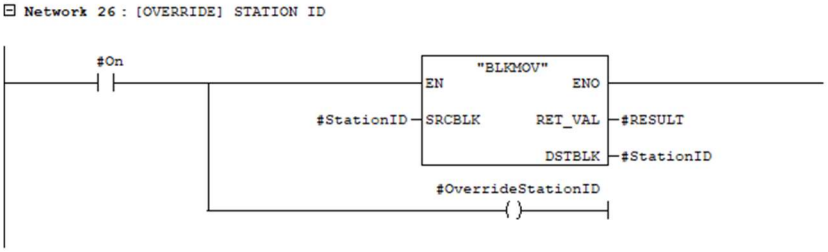
Note: Not implemented (future functionality).



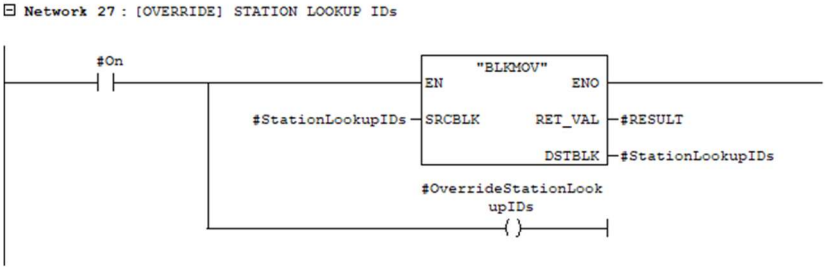
2. The #PLCID setting can be overridden to change the identity of the machine.



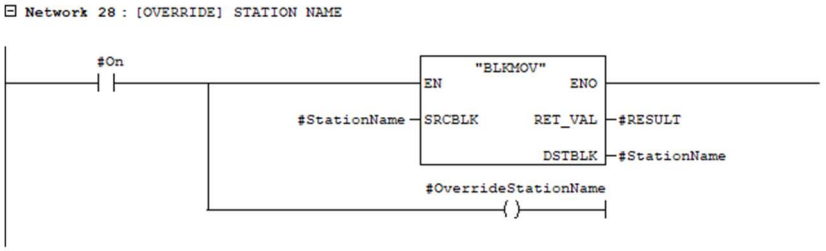
3. The #StationID setting can be overridden to change the identity of the station, which is typically used to affect permissions downstream, or if multiple models can be run on the machine that have different amounts of process data and need to be separated logically in the database.



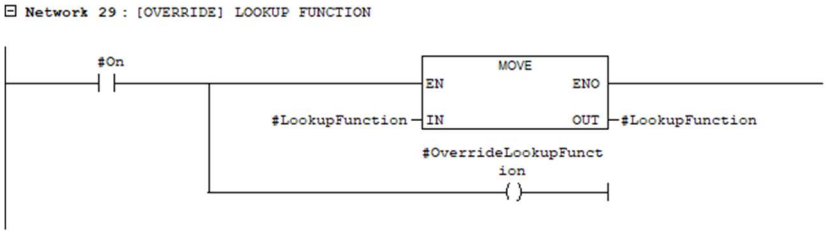
4. The #StationLookupIDs setting can be overridden to change which StationIDs to search for in the database when performing a function (2) or function (4) lookup.



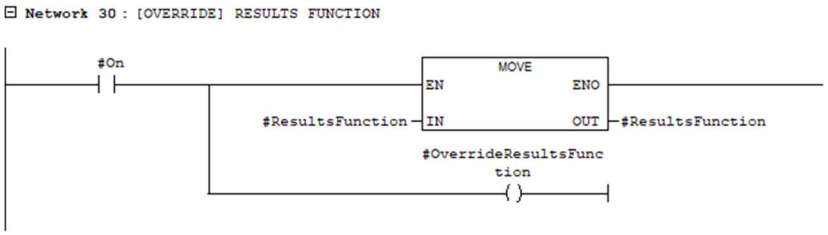
5. The #StationName setting can be overridden to change the human readable description of the station.



6. The #LookupFunction setting can be overridden to change which lookup function is used during a Lookup request.



7. The #ResultsFunction setting can be overridden to change which results function is used during a Results request.

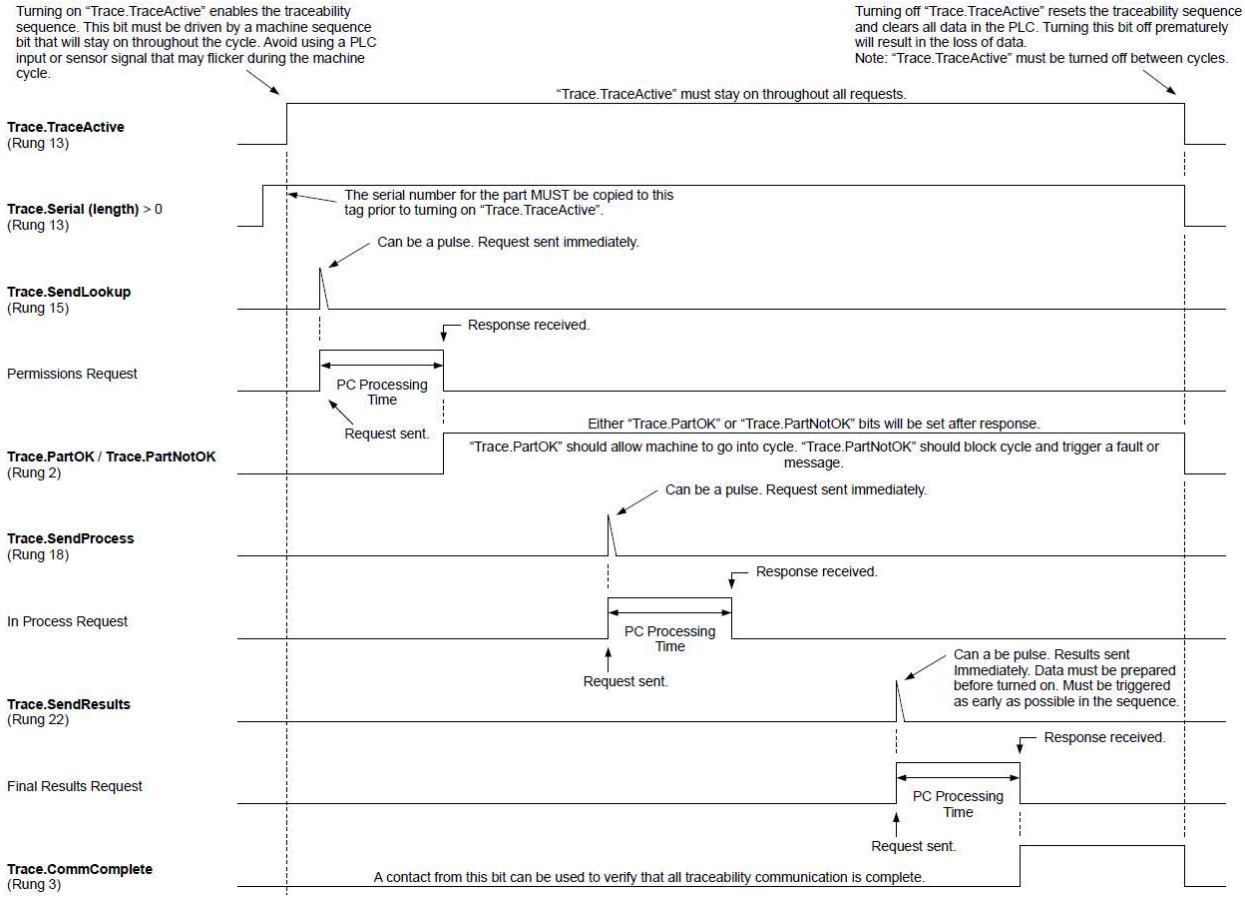


2.9 Traceability Tag Structure

The Siemens traceability logic requires the following tag structure to be present within the Function Block starting at byte (0) in the Static tags. Newer S7 PLCs programmed with TIA Portal can take advantage of a UDT to define the tag structure. The traceability system will not function correctly if the tag structure is altered in any way.

Tag Name	Data Type	Memory Offset
VersionMajor	Byte	0.0
VersionMinor	Byte	1.0
VersionRev	Byte	2.0
CommPLC	Array[0..31] of Bool	4.0
CommPC	Array[0..31] of Bool	8.0
CommGuid	Array[0..15] of Byte	12.0
OneShots	Array[0..31] of Bool	28.0
TransactionPLC	Int	32.0
TransactionPC	Int	34.0
Line	String[30]	36.0
PLCID	String[30]	68.0
StationID	String[30]	100.0
StationName	String[80]	132.0
StationLookupIDs	String[80]	214.0
LookupFunction	Int	296.0
ResultsFunction	Int	298.0
SerialLength	Int	300.0
Serial	String[30]	302.0
SerialCaptured	String[30]	334.0
BlacklistName	String[30]	366.0
BlacklistMessage	String[80]	398.0
BlacklistDate	String[20]	480.0
PreviousPLCID	String[30]	502.0
PreviousStationID	String[30]	534.0
PreviousStatus	Dint	566.0
PreviousModel	String[30]	570.0
PreviousStationName	String[80]	602.0
PreviousElapsedDays	Int	684.0
PreviousElapsedHours	Int	686.0
PreviousElapsedMinutes	Int	688.0
PreviousElapsedSeconds	Int	690.0
ResultsStatus	Dint	692.0
ResultsModel	String[30]	696.0
TraceActive	Bool	728.0
SerialReady	Bool	728.1
LookupEnable	Bool	728.2
LookupSend	Bool	728.3
LookupRequest	Bool	728.4
LookupComplete	Bool	728.5
LookupError	Bool	728.6
InProcessEnable	Bool	728.7
InProcessSend	Bool	729.0
InProcessComplete	Bool	729.1
InProcessError	Bool	729.2
ResultsEnable	Bool	729.3
ResultsSend	Bool	729.4
ResultsComplete	Bool	729.5
ResultsError	Bool	729.6
CommComplete	Bool	729.7
EvaluatePermissions	Bool	730.0
PartOKConditions	Bool	730.1
PartOK	Bool	730.2
PartNotOK	Bool	730.3
OverridePermissions	Bool	730.4
PCPartOK	Bool	730.5
PCPartNotOK	Bool	730.6
PartBlacklisted	Bool	730.7
HeartbeatTimeout	Bool	731.0
On	Bool	731.1
Off	Bool	731.2
DecisionInhibitPulse	Bool	731.3
OverrideLine	Bool	731.4
OverridePLCID	Bool	731.5
OverrideStationID	Bool	731.6
OverrideStationName	Bool	731.7
OverrideStationLookupIDs	Bool	732.0
OverrideLookupFunction	Bool	732.1
OverrideResultsFunction	Bool	732.2
ResetGuid	Bool	732.3
GuidTimer	Timer	734.0
HeartbeatTimer	Timer	Varies

2.10 Traceability Timing Chart



RECORD OF RECORDS

Revision No	Date	Section	Description
001	27AU21	All	Original Approval & Issue Date
002			
003			
004			
005			
006			
007			
008			
009			
010			
011			
012			
013			
014			
015			
016			
017			
018			
019			
020			