



Machine Controls Traceability Interface  
Studio 5000

Global Common

SD-1052

ISSUED	August 27, 2021
REVISED	January 9, 2025

© 2021 Nexteer Automotive

All rights reserved.

## Table of Contents

1	Scope and Purpose .....	3
2	Traceability Part Permission & Result Data Configuration.....	4
3	Central Comm Traceability Application Configuration .....	23
4	Database Part Validation Traceability Application Configuration.....	28
5	Test Cycle Application Technical Notes .....	45
A.	Traceability Timing Chart.....	47
B.	Traceability Field Guidelines .....	48

## 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.nexteerdatabase.com](http://www.nexteerdatabase.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 and to provide device configuration guidance for the Traceability application.
- 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.
- 1.2.5 This Document shall be used in conjunction with the Nexteer Traceability Input Document to configure and program the Traceability Program and PLC Logic

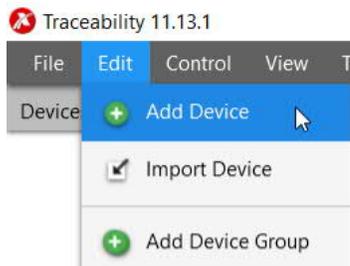
## 2 Traceability Part Permission & Result Data Configuration

The following configuration details show how to configure a device in the traceability application. Each device configured will have a connection to the PLC tags in a **R26\_Trace\_V2\_Station** routine. Each import of this trace routine into the PLC requires a corresponding device configured in the traceability application.

### 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 the StationID value that is referenced in the Nexteer Traceability Input document. For example: SD123456X01, SD123456X51, SD123456OP010.0, SD123456ST02, etc.
  - b. Set the device type to **Logix5000 Traceability v2.1**.
  - c. Click OK to finish adding the device connection.

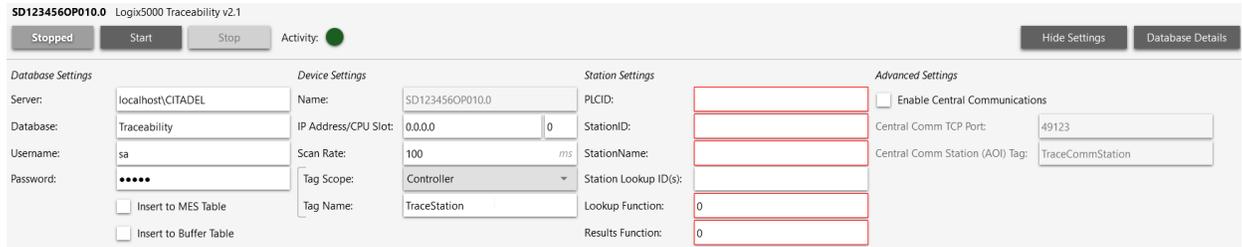


#### 2.1.2 Device settings panel.

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.



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

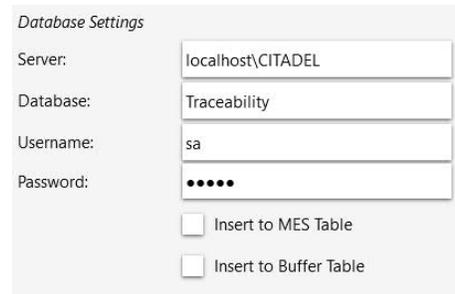


The screenshot shows the configuration interface for SD123456OP010.0. It includes a status bar with 'Stopped', 'Start', and 'Stop' buttons, and an 'Activity' indicator. The main settings are divided into four sections: Database Settings, Device Settings, Station Settings, and Advanced Settings. Database Settings includes fields for Server, Database, Username, and Password, with checkboxes for 'Insert to MES Table' and 'Insert to Buffer Table'. Device Settings includes Name, IP Address/CPU Slot, Scan Rate, Tag Scope, and Tag Name. Station Settings includes PLCID, StationID, StationName, Station Lookup ID(s), Lookup Function, and Results Function. Advanced Settings includes a checkbox for 'Enable Central Communications', 'Central Comm TCP Port', and 'Central Comm Station (AOI) Tag'.

### 2.1.3 Configure Database Settings

The default database configuration settings normally do not require modification.

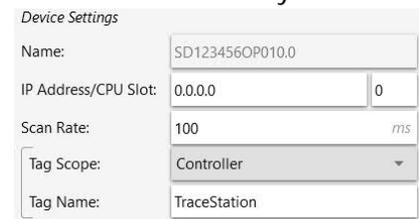
- Server:** The default Microsoft SQL server name is "localhost\CITADEL" which contains the standard Nexteer traceability database.
- Database:** The default database name is "Traceability".
- Username/Password:** This contains the credentials for the authorized database user. The default username is "sa" and password is "admin".
- 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.
- 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.



This close-up shows the Database Settings section. It contains input fields for Server (localhost\CITADEL), Database (Traceability), Username (sa), and Password (masked with dots). Below these fields are two checkboxes: 'Insert to MES Table' and 'Insert to Buffer Table', both of which are currently unchecked.

### 2.1.4 Configure Device Settings

- IP Address / CPU Slot:** Enter the IP Address of the PLC and slot number of the CPU within the PLC rack.
- Scan Rate:** The default scan rate of how often the software polls the machine for new data is "100" milliseconds. This should not require modification.
- Tag Scope / Program Name / Tag Name:** The default tag name is "TraceStation". This is a user defined tag in the `R26_Trace_V2_Station` routine containing the tag structure needed to interface with the traceability application. When multiple `R26_Trace_V2_Station` routines exist in the same program, this tag name shall be unique such as "OPxxx\_" or "STxxx " + Name of the Component specified in the Traceability Input Document. The following 3 settings are used to configure a reference to the tag.



This close-up shows the Device Settings section. It includes input fields for Name (SD123456OP010.0), IP Address/CPU Slot (0.0.0.0 and 0), Scan Rate (100 ms), Tag Scope (Controller), and Tag Name (TraceStation).

- a. **Tag Scope:** Select the scope of the PLC tag.
- b. **Program Name (optional):** If Tag Scope is set to "Program", this setting will be displayed. Enter the name of the program within the PLC where the tag resides.
- c. **Tag Name:** Enter the name of the PLC tag.

Example (Controller Scope):

Tag Scope:	Controller
Tag Name:	TraceStation

Example (Program Scope):

Tag Scope:	Program
Program Name:	OP10
Tag Name:	TraceStation

### 2.1.5 Configure Station Settings

1. **PLCID:** Enter the machines SD number for the PLC ID. For example: SD123456.
2. **StationID:** Enter the StationID value that is referenced in the Nexteer Traceability Input document. For example: SD123456X01, SD123456X51, SD123456OP010.0, SD123456ST02, 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.

*Station Settings*

PLCID:

StationID:

StationName:

Station Lookup ID(s):

Lookup Function:

Results Function:

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. (Used for 'Marriage' of child to a parent part)
  - c. Function (20): Used for collecting a large amount of data that is intended to be sent directly to the Nextrace reporting system. A full data record is inserted into the **dbo.\_nextrace\_data** table with the Name & Data values represented as separate comma separated strings in separate **Name** and **Data** columns. A small record is also insert into the **dbo.\_status** table that contains the header information to use for permissions if needed.

**Note:** It is recommended to use function (22) instead of function (20) for new applications.

- d. Function (21): Used for collecting a large amount of data that is intended to be sent directly to the Nextrace reporting system. A full data record is inserted into the **dbo.\_nextrace\_data** table with the Name & Data represented as separate comma separated strings in separate **Name** and **Data** columns. No records are inserted into the **dbo.\_status** table.

**Note:** It is recommended to use function (23) instead of function (21) for new applications.

- e. Function (22): Used for collecting a large amount of data that is intended to be sent directly to the Nextrace reporting system. A full data record is inserted into the **dbo.\_nextrace\_data** table with the Name & Data values represented as single interlaced comma separated strings in a single **NameData** column. A small record is also insert into the **dbo.\_status** table that contains the header information to use for permissions if needed.
- f. Function (23): Used for collecting a large amount of data that is intended to be sent directly to the Nextrace reporting system. A full data record is inserted into the **dbo.\_nextrace\_data** table with the Name & Data values represented as single interlaced comma separated strings in a single **NameData** column. No records are inserted into the **dbo.\_status** table.

### 2.1.6 Advanced Settings

1. **Enable Central Communications:** For large assembly lines with 20 or more stations this option can be enabled to conserve class 3 communication bandwidth of the PLC Controller. When using this you will need one **R26a\_CentralComm** rung that shall be imported into each **R26\_Trace\_V2\_Station** PLC routine.



Advanced Settings

Enable Central Communications

Central Comm TCP Port: 49123

Central Comm Station (AOI) Tag: TraceComm\_OP010

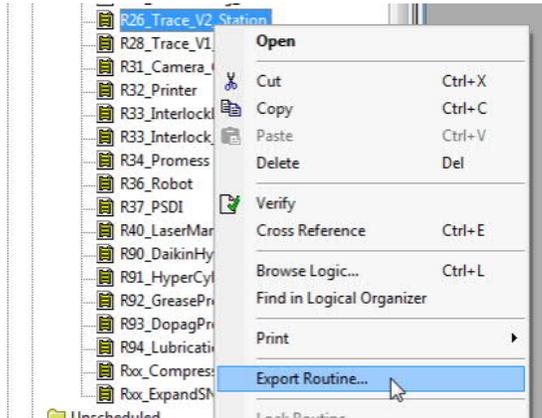
2. **Central Comm TCP Port:** This port number shall be unique from any other port numbers used in the same traceability PC. Port number range can be from 49100 to 49999.
3. **Central Comm Station (AOI) Tag:** This Tag name shall match the UDT tag name in the PLC for the **R26a\_CentralComm** rung that was imported into each **R26\_Trace\_V2\_Station** PLC routine.

### 2.2 Logic Configuration (R26\_Trace\_V2\_Station)

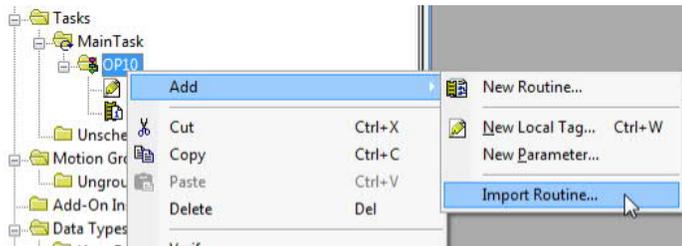
This routine is used for traceability part permissions and data collection for each configured device in the traceability application. A uniquely named routine shall be configured for each device defined in the traceability application. The logic in this routine shall modified to meet the requirements of the application. See Annex A for a detailed timing chart showing logic routine and traceability application communication.

## 2.2.1 Importing the Routine

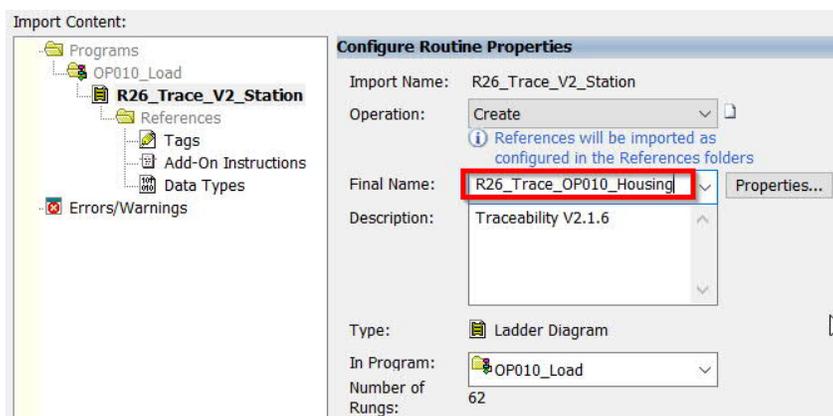
1. Export the **R26\_Trace\_V2\_Station** from the Nexteer PLC library program by right clicking on the routine and clicking **Export Routine...**. Save the file to a location on your hard drive.



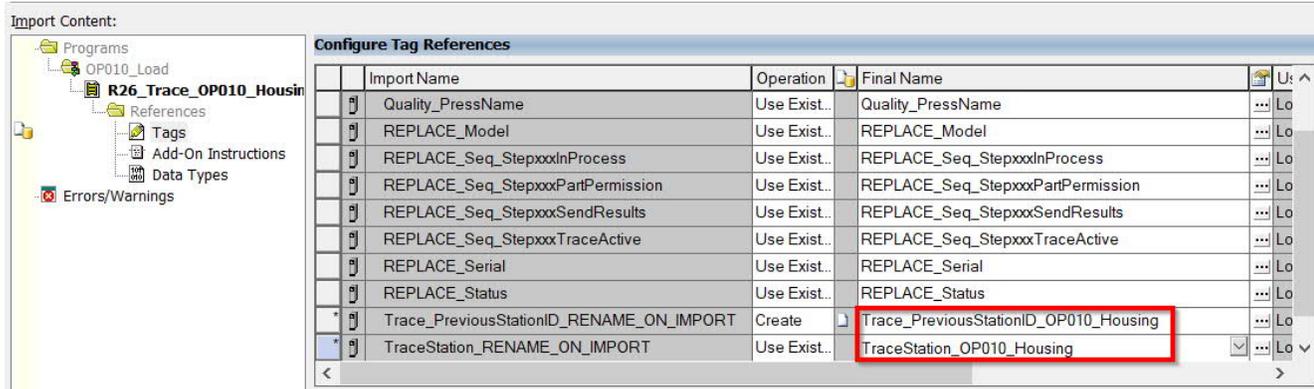
2. Right click on the program that **R26\_Trace\_V2\_Station** needs to be imported into and click **Add > Import Routine** in the context menu. Browse to the location that exported routine was saved to and click **Open**.



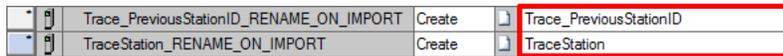
3. The **Import Configuration** dialog will appear. Ensure that **Operation** is set to **Create** by entering a unique value for the **Final Name** setting. Typically, the name should be "R26\_Trace\_" + ("OPXXX\_" or "STxxx\_") + Name of the Component specified in the Traceability Input Document.



- Click on the **Tags** option in the **Import Content** tree. There are 2 tags that need to be renamed prior to importing the routine. These tags contain the value **RENAME\_ON\_IMPORT** in their tag name, which can be removed completely or replaced with a value that ensures that the tag is unique to the station.



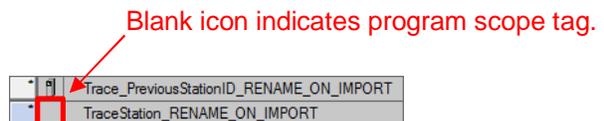
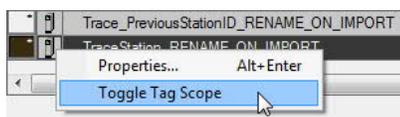
- For a single station machine, the **RENAME\_ON\_IMPORT** portion of the tag name should simply be removed from the tag names.



- For a machine that requires multiple **R26\_Trace\_V2\_Station** routines to be imported, the **RENAME\_ON\_IMPORT** portion should be replaced with station specific information to ensure that the tag names are unique for each routine. Typically, the **RENAME\_ON\_IMPORT** portion of the name should be replaced with ("OPxxx\_" or "STxxx\_") + Name of the Component specified in the Traceability Input Document.



- The **TraceStation** tag is also allowed to be Controller or Program scope. The scope of the tag can be changed by right clicking the icon and choosing **Toggle Tag Scope** in the context menu.



## 2.2.2 General Configuration

- The current running model description shall be copied into the **TraceStation.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.



2. The scanned serial number shall be copied to the **TraceStation.Serial** tag prior to turning on the **TraceStation.TraceActive** OTE instruction. The **TraceStation.TraceActive** OTE instruction enables the traceability functions. The logic that controls the **TraceStation.TraceActive** OTE instruction shall:

- a. Remain **on** 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.

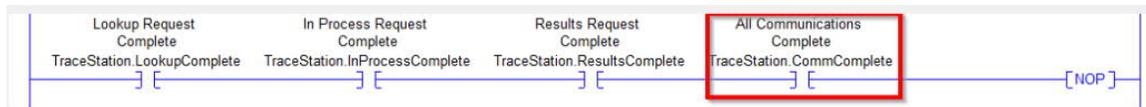
**Note:** Disabling the **TraceStation.TraceActive** tag will abort the Trace function requests and clears all **TraceStation** tag data.



3. The **TraceStation.HeartbeatTimeout** tag is used to signal a loss of communication with the Trace PC and shall be used in the **R08c\_Fault\_CycleStop** routine to inhibit machine cycling. A loss of communication should prohibit the next cycle from initiating.



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



5. The **R26\_Trace\_V2\_Station** shall not be edited below the phrase "DO NOT EDIT ANYTHING BELOW THIS RUNG"



### 2.2.3 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.

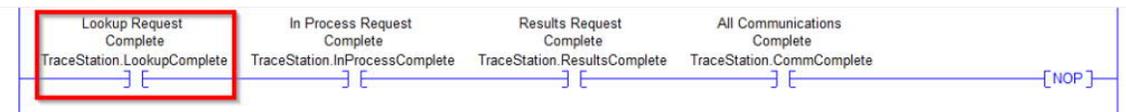
1. Enable the **TraceStation.EnableLookup** OTE instruction.



2. The **Lookup Function** is set by Traceability application. Refer to section [2.1.55](#).
3. The **TraceStation.SendLookup** OTE instruction initiates the Lookup Request with the Nexteer Traceability Application.

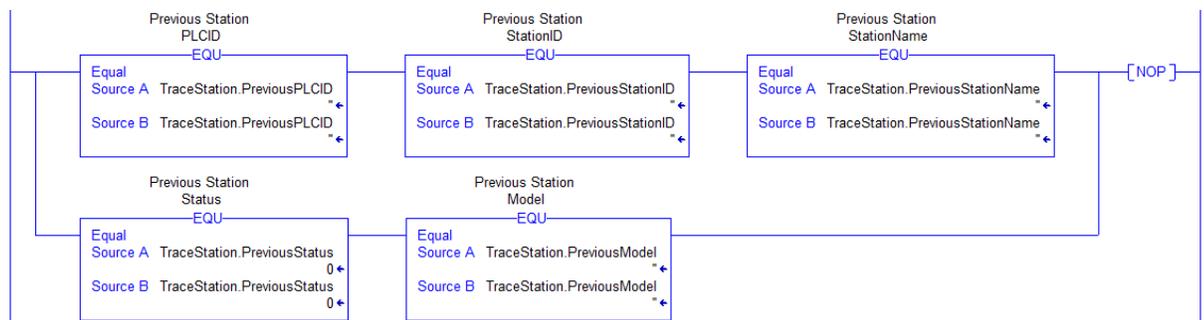


4. For Function (2) and (4) lookup requests the **Station Lookup IDs** are set by the Traceability application. Reference section [2.1.54](#).
5. All data has been returned to the PLC when the **TraceStation.LookupComplete** OTE is turned on.

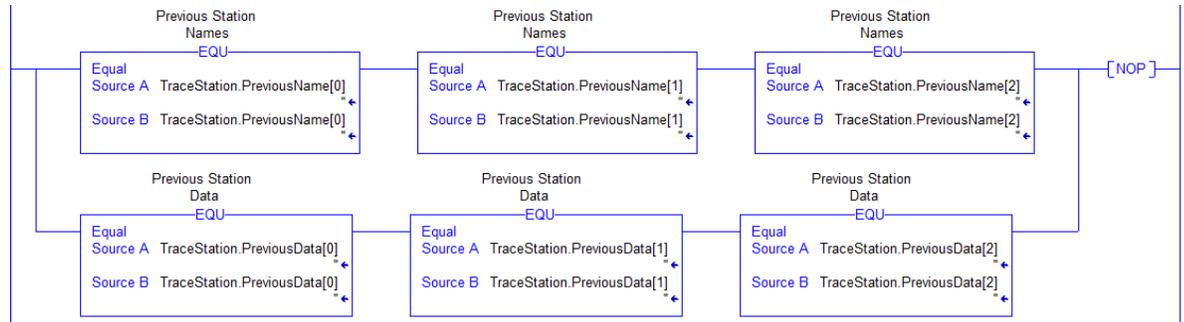


6. Previous station information.

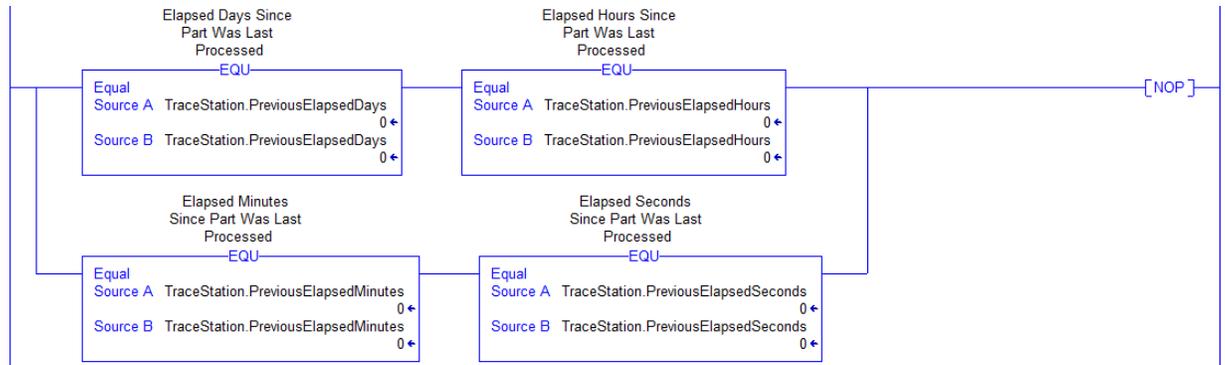
- a. The **TraceStation.PreviousPLCID** tag contains the machine identifier of the previous machine.
- b. The **TraceStation.PreviousStationID** tag contains the station identifier of the previous station. This tag is typically used for comparison against of list of expected StationIDs to ensure that the part was processed at a valid station previously.
- c. The **TraceStation.PreviousStationName** tag contains the human readable description of the previous station.
- d. The **TraceStation.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 **TraceStation.PreviousModel** tag contains the model that was configured when the part was run at the previous station.



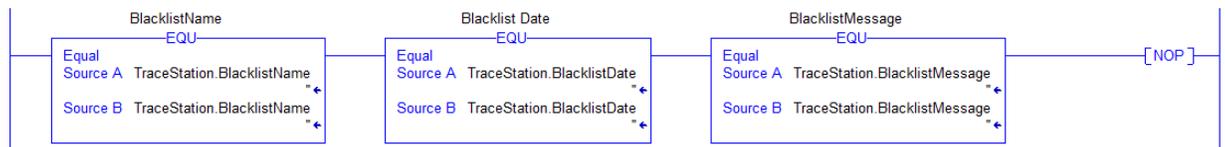
7. Previous station data. These tags are only populated during a function (2), (3), or (4) Lookup request.
  - a. The `TraceStation.PreviousName[X]` tag array contains the descriptions of the process data that was collected at the previous station.
  - b. The `TraceStation.PreviousData[X]` tag array contains the process data values that were collected at the previous station.



8. The `TraceStation.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.



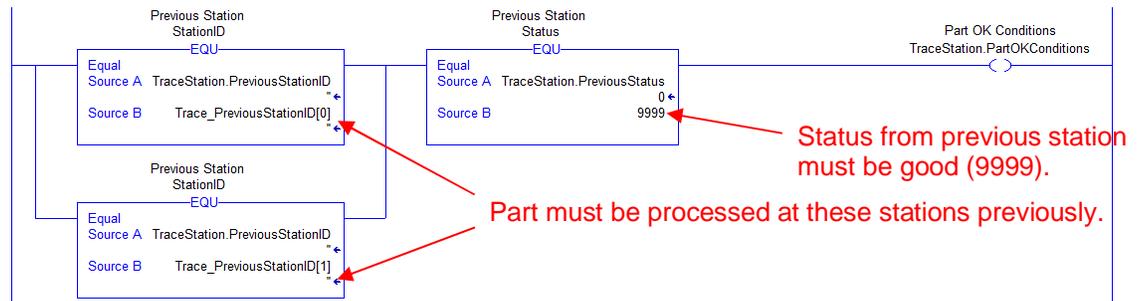
9. If the part is found to be blacklisted during a Lookup request, the `TraceStation.Blacklist(...)` tags will contain detailed information about the blacklist.



10. The logic shall be programmed to appropriately control the `TraceStation.PartOKConditions` OTE instruction for the application.

**Note:** The information from the Nexteer Traceability Input document shall be used for programming this section.

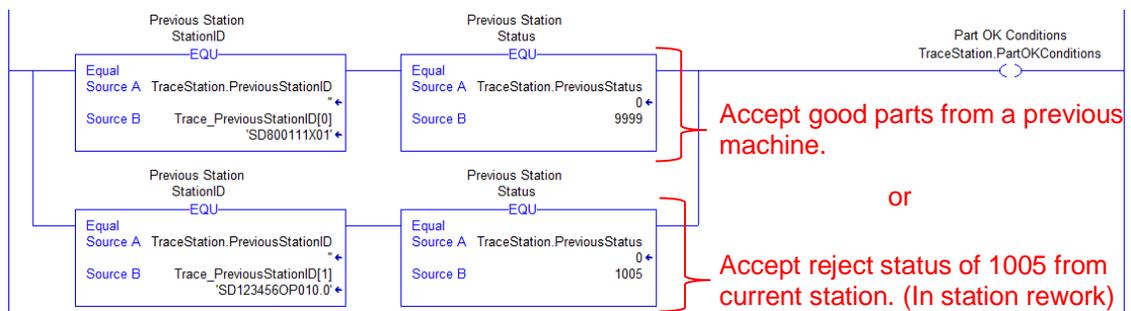
- a. **Example:** Expecting good parts from one of two previous stations.



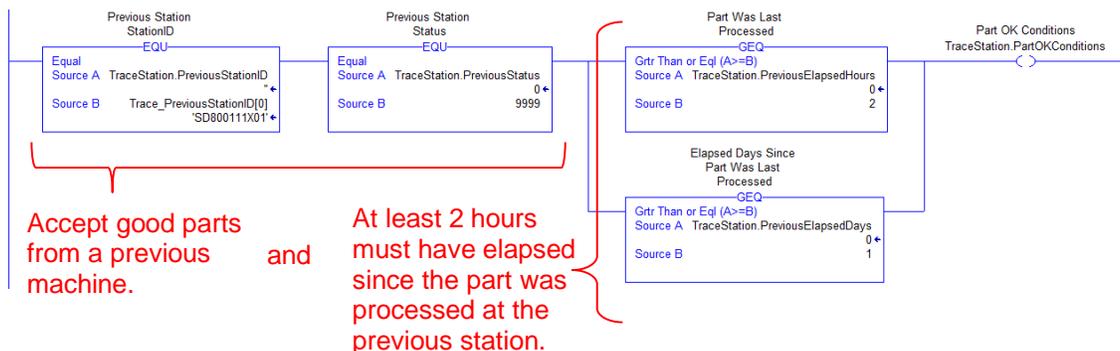
- b. **Example:** Expecting no record found.



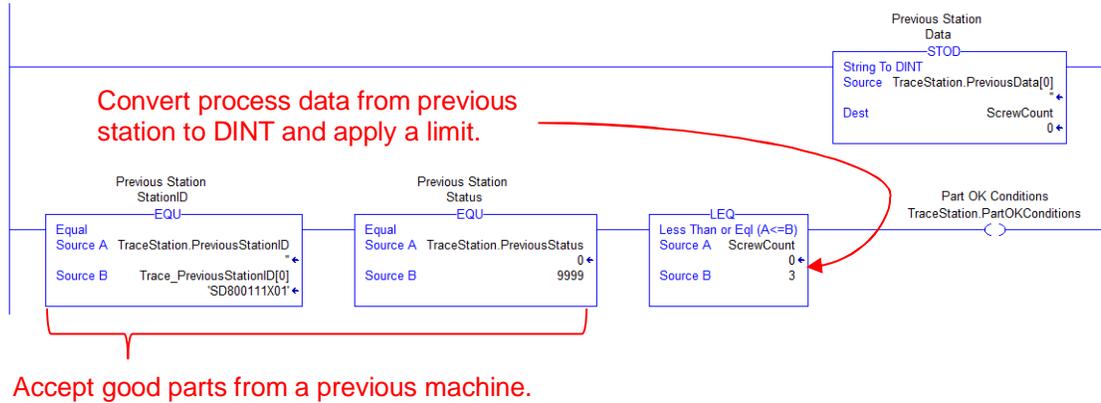
- c. **Example:** Expecting good parts from previous station and rework of specific reject code in station.



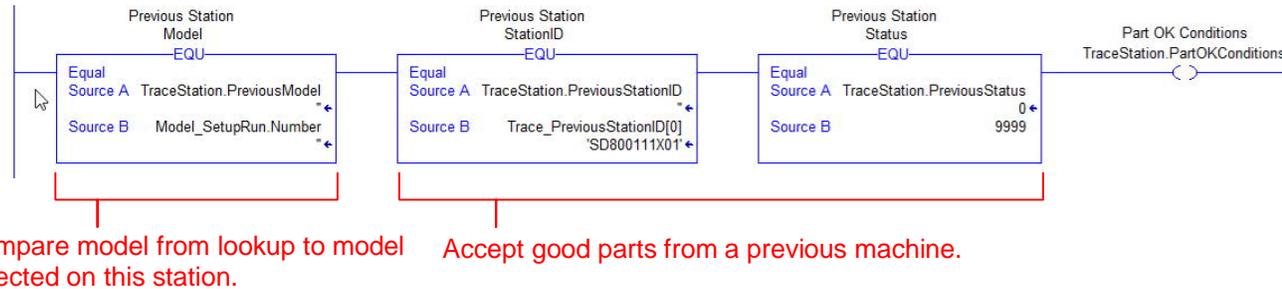
- d. **Example:** Verification of elapsed time since part was processed at previous operation.



e. Example: Verification of previous data.



f. Example: Verification of model.



11. The **TraceStation.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.



12. The **TraceStation.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.



### 2.2.4 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 so it cannot be reprocessed in the machine. It shall be enabled by a sequence step before the machine begins to alter the part.

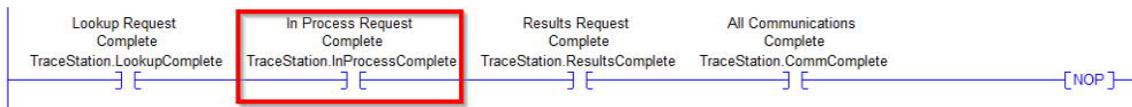
1. Enable the **TraceStation.EnableInProcess** OTE instruction.



2. The **TraceStation.SendInProcess** OTE instruction initiates the In Process Request with the Nexteer Traceability Application.



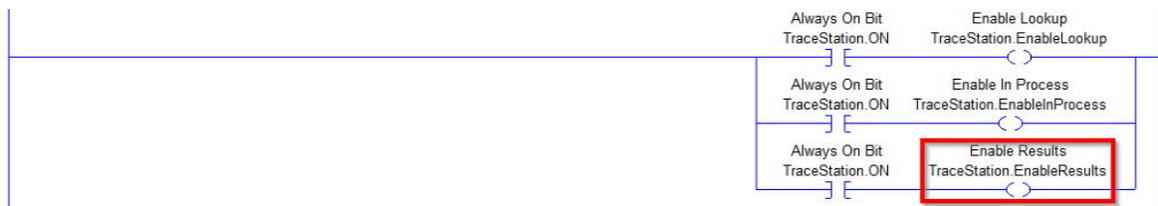
3. When the In Process Request is complete and a record has been inserted into the database the **TraceStation.InProcessComplete** tag will be turned on.



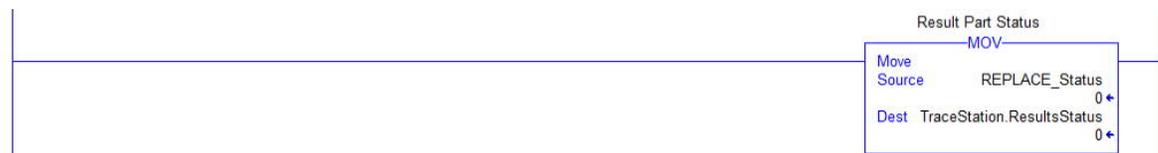
### 2.2.5 Results Request (Send Part Status and Process Data)

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 **TraceStation.EnableResults** OTE instruction.

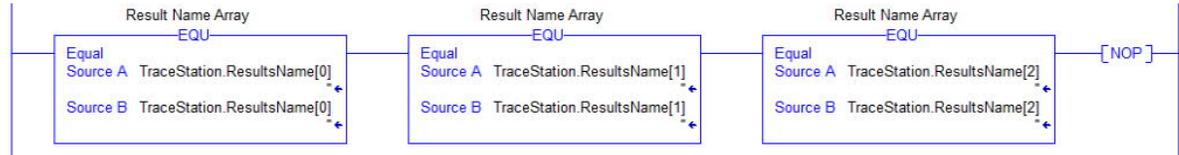


2. The **Results Function** is set by Traceability application. Reference section 2.1.56.
3. The **TraceStation.ResultsStatus** tag shall be loaded with the quality status of the processed part.



4. The **TraceStation.ResultsName[X]** array shall be updated to reflect the appropriate descriptions for each process data value, including units, that is to be collected at the end of the machine cycle. A maximum number of 150 Name values can be configured.

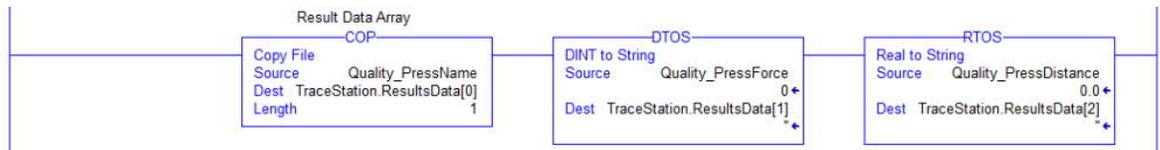
**Note:** The information from the Nexteer Traceability Input document shall be used for programming this section.



5. The **TraceStation.ResultsData[X]** array shall be updated to prepare the process data values that need to be collected at the end of the cycle. A maximum number of 150 Data values can be configured.

- a. **STRING** type data can be copied directly the array with a **COP** instruction.
- b. **DINT** type data must be converted to **STRING** type with a **DTOS** instruction prior to copying the array.
- c. **REAL** type data must be converted to **STRING** type with an **RTOS** instruction prior to copying to the array.

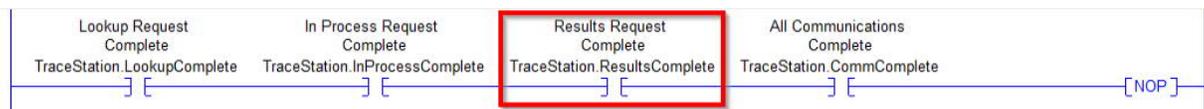
**Note:** Using a **COP** instruction with **DINT** or **REAL** types will cause data corruption and may lock up communication with the Traceability application.



6. The **TraceStation.SendResults** OTE instruction initiates the Results Request with the Nexteer Traceability Application.



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



### 2.2.6 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, multiple models that have different traceability requirements., it may be necessary to change the **StationID** or **LookupFunction** depending on which model is currently set to run.

**Note:** Overriding Traceability application settings is not a normal process function. Nexteer Central Manufacturing IT shall be consulted before editing the following rungs.

1. The **TraceStation.Line** setting can be overridden to change the line or program that the part being run is associated with.



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



3. The **TraceStation.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 **TraceStation.StationName** setting can be overridden to change the human readable description of the station.



5. The `TraceStation.StationLookupIDs` setting can be overridden to change which station IDs are used during a Lookup request.



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



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



### 2.3 HMI Status Display and Diagnostics

1. **Automatic Screen:** Multi-state indicators and string displays shall be configured in the Automatic Screen of the HMI application to display the traceability status to the operator. These objects interface directly with the **R26\_Trace\_V2\_Station** routine tags.

AUTOMATIC CYCLE SCREEN	USER LOGOUT "s...s" Timeout = ## min		DIRECTORY SCREEN	s...s s...s SELECTED
MACHINE STATUS:	READY TO CYCLE			SELECT AUTO MODE
PART STATUS:	READY TO CYCLE			SELECT MANUAL MODE
LOT TRACKING:	READY TO CYCLE			
CYCLE TIME:	###.## SECONDS	SERIAL:	ssssssssssssssssssssss	
PART-TO-PART TIME:	###.## SECONDS	PART STATUS:	PERMISSIONS	
		PLC ID:	ssssssssssssssssssss	
		STATUS:	ssssssssssssssssssss	
OPERATOR PROMPT:	LOAD PART INTO FIXTURE			RETURN ALL



### 3 Central Comm Traceability Application Configuration

For large assembly lines that have 25 or more **R26\_Trace\_V2\_Station** routines, it may be necessary to implement a special central communication plugin to significantly reduce class 3 communication between the PC and PLC. A single **Logix5000 Traceability v2.1 Central Comm** device can be configured to take over sending/receiving the **CommPLC & CommPC** tag values on behalf of all the **R26\_Trace\_V2\_Station** routines. The communication is made more efficient by grouping the **CommPLC & CommPC** tags from all the traceability routines into a pair of arrays that are read and written by the central communication device each scan. These values are then made accessible to each **Logix5000 Traceability v2.1** device in the traceability software through an internal TCP communication protocol.

Each **Logix5000 Traceability v2.1** device must then be configured to communicate with the central communication device through a common TCP port, and will no longer read or write their respective **CommPLC & CommPC** tags in the PLC. However, when a request such as lookup or results is triggered, the **Logix5000 Traceability v2.1** device will directly read and write the required tags within the routines to perform the necessary functions.

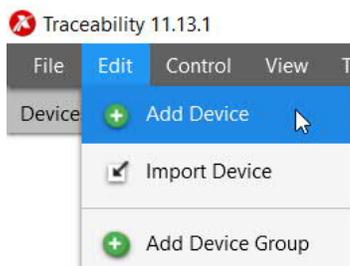
A single **R26a\_CentralComm** rung must be added to each **R26\_Trace\_V2\_Station** routine which contains an AOI that handles translating the **CommPLC & CommPC** tags from the routine into the array that the **Logix5000 Traceability v2.1 Central Comm** device can access.

This section explains how to configure a central communication device in the traceability application and the logic within the PLC.

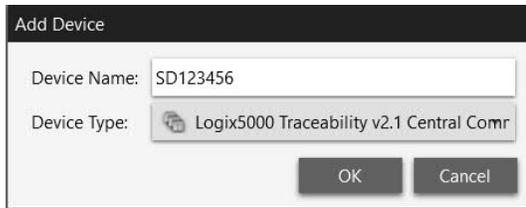
#### 3.1 Traceability Application Configuration

##### 3.1.1 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 the **Device Name** which is typically the machine SD number. For example: SD123456.
  - b. Set the device type to **Logix5000 Traceability v2.1 Central Comm.**
  - c. Click OK to finish adding the device connection.

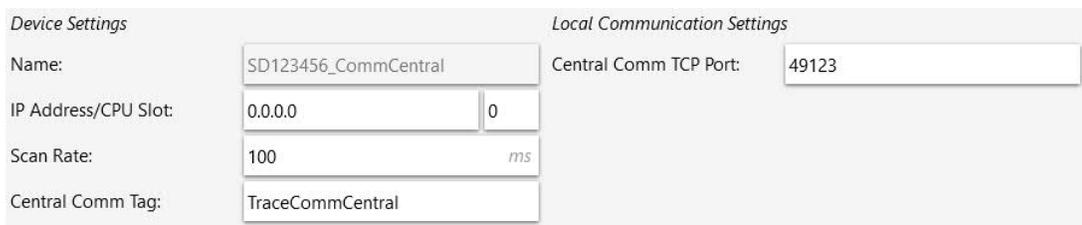


### 3.1.2 Device settings panel.

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.

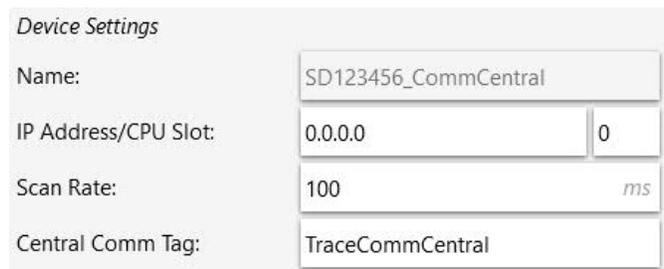


The following panel will be displayed. It is used to configure the PLC connection settings.



### 3.1.3 Configure Device Settings

1. **Name:** Typically set to the SD number of the machine + “\_CommCentral”.
2. **IP Address:** Enter the IP Address of the PLC.
3. **Scan Rate:** The default scan rate of how often the software polls the machine for new data is “100” milliseconds. This should not require modification.



4. **Central Comm Tag:** The default tag name is "TraceCommCentral". This is a user defined tag in the **R26a\_CentralComm** routine containing the tag structure needed to interface with the traceability application.

### 3.1.4 Local Communication Settings

1. **Central Comm TCP Port:** This port number should be unique from any other TCP number used in the same Plant. Port number range can be from 49100 to 49999. This port number must match the port number setup in all of the **Advanced Settings** for all of the **Logix5000 Traceability v2.1** devices.

Local Communication Settings

Central Comm TCP Port:

### 3.1.5 Advanced Settings in Logix5000 Traceability v2.1 Devices

1. **Enable Central Communications:** This must be checked for Central Comms to function.
2. **Central Comm TCP Port:** This port number should be unique from any other TCP number use in the same Plant. Port number range can be from 49100 to 49999 and must match the port number setup in Central Comm Device Settings.
3. **Central Comm Station (AOI) Tag:** This Tag name should match the UDT tag name in the PLC for the **R26a\_CentralComm** routine rung.

Advanced Settings

Enable Central Communications

Central Comm TCP Port:

Central Comm Station (AOI) Tag:

### 3.2 Logic Configuration (R26a\_CentralComm)

1. Determine the version of **R26\_Trace\_V2\_Station** routine that is being used in your program by selecting one of the **R26\_Trace\_V2\_Station** routines in your program and checking the version from the comment text on the first rung.

**Note:** Currently, the **R26a\_CentralComm** routine only supports v2.1.5 and v2.1.6 **R26\_Trace\_V2\_Station** logic routines.

```

=====
TRACE_V2_STATION ROUTINE
Version: 2.1.6

For use with Traceability Application v11.0.0 or newer, device type: Logix5000 V2.1
Traceability Application v11.12.0 or newer is required to utilize the Central Communication feature.

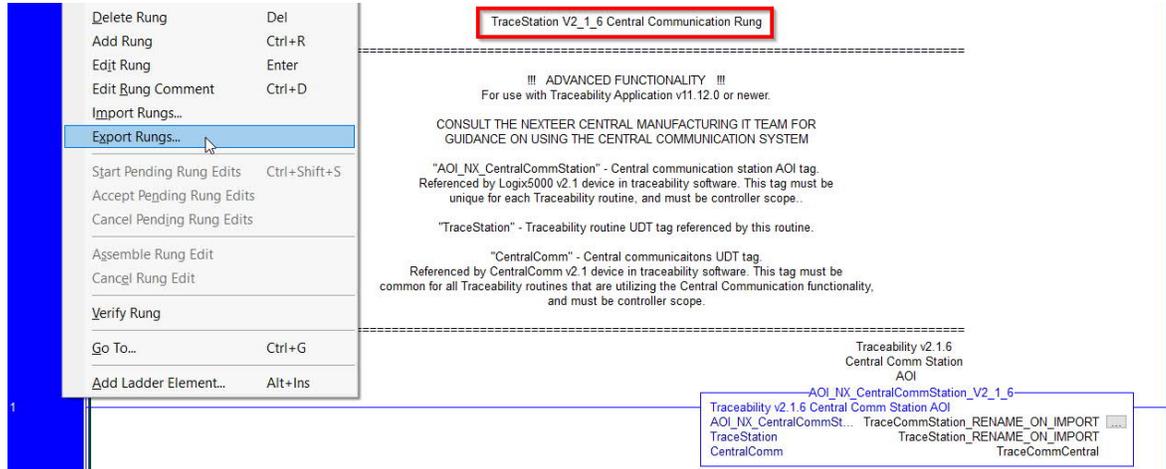
R26_Trace_V2_Station routine(s) communicate to the PC through PLC requests (both getting & sending data)
This routine must be included in all machines that require any traceability function.
This routine is required for each component with traceability requirements.
(This means multiple R26_Trace_V2_Station routines may be required in a single program).

Exporting this routine from the library and importing into the desired program will automatically populate the required UDTs.
Exception: The "TraceStation_RENAME_ON_IMPORT" tag must be renamed with each import.
(Renaming of this tag should be associated with the component, for example: TraceStation_Motor)

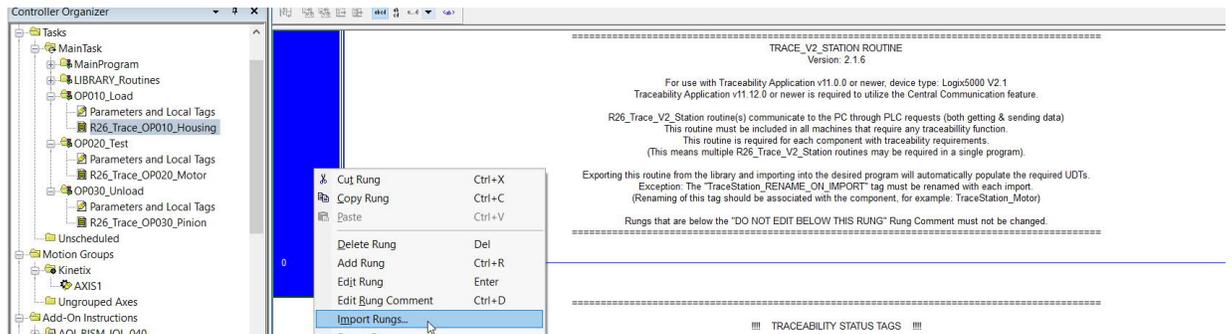
Rungs that are below the "DO NOT EDIT BELOW THIS RUNG" Rung Comment must not be changed.
=====

```

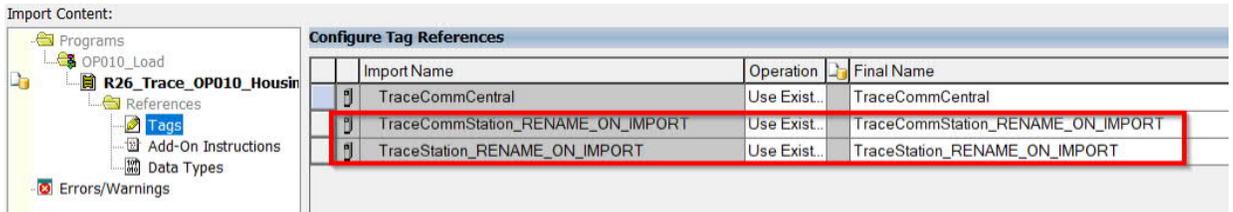
- From the **R26a\_CentralComm** routine library select and export the rung that matches the version of the **R26a\_CentralComm** routine your program is using. Save the file to a location on your hard drive.



- From the first **R26\_Trace\_V2\_Station** routine select the first rung and import the rung that is saved to your hard drive.

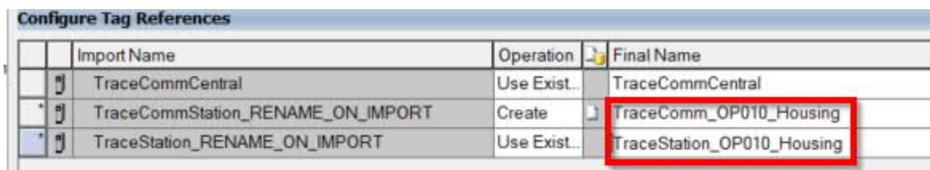


4. Click on the **Tags** option in the **Import Content** tree. There are 2 tags that need to be renamed prior to importing the rung. These tags contain the value **RENAME\_ON\_IMPORT** in their tag name which can be replaced with a value that ensures that the tag is unique to the station.

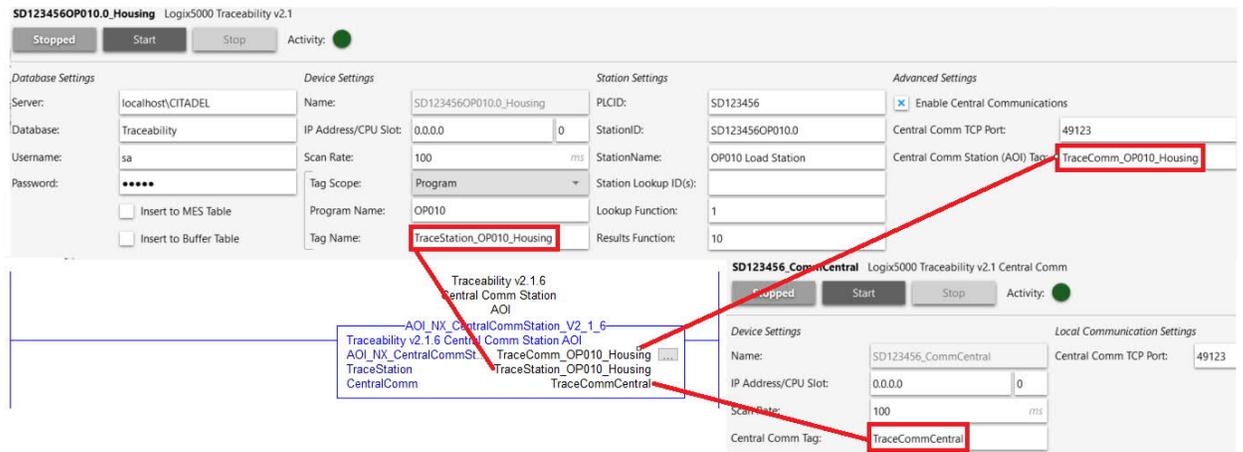


- a. The **RENAME\_ON\_IMPORT** portion should be replaced with station specific information to ensure that the tag names are unique for each routine. Typically, the **RENAME\_ON\_IMPORT** portion of the name should be replaced with 'OPxxx\_ or STxxx\_' + Name of the Component from the Traceability Input Document.

**Note:** Tag must be set as controller scope.



5. For the Central Comm to properly function, the Tag names in the PLC rung for Central Comms must match the tag names defined in the Traceability Comm Central device and the Station Traceability device as shown below.



6. Repeat steps 3 through 5 for the remaining **R26\_Trace\_V2\_Station** routines.

## 4 Database Part Validation Traceability Application Configuration

Note: Requires Traceability Application Version 12.1.14 or newer.

The **Database Part Validator v2.0** shall be used to verify every component has passed all operations of the Assembly process before leaving the last operation, or when it is married to a Master component.

The **Database Part Validator v2.0** will write a record to Traceability once the result of the test is complete. For a Passed status the record will have a '9999' status and for a failed status the record will have a status of '3002' and will get added to the Blacklist Table.

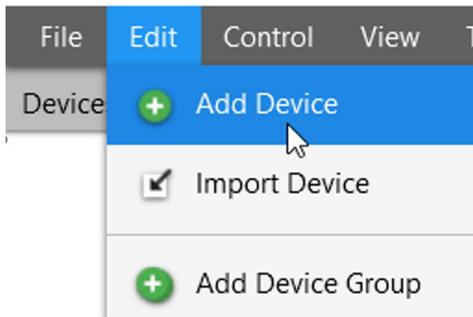
Each **Logix5000 Database Part Validator v2.0** device must then be configured to communicate with the R27\_Trace\_V2\_PartValidator Logic routine.

This section explains how to configure a Database Part Validator device in the traceability application and the logic within the PLC.

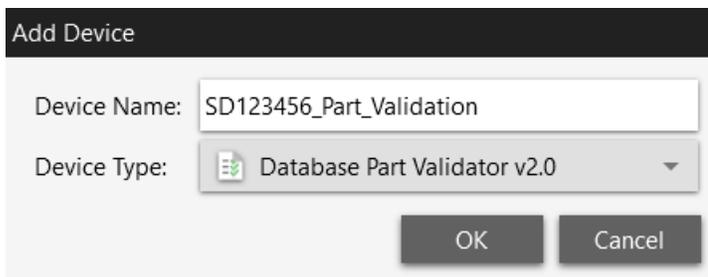
### 4.1 Traceability Application Configuration

#### 4.1.1 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.

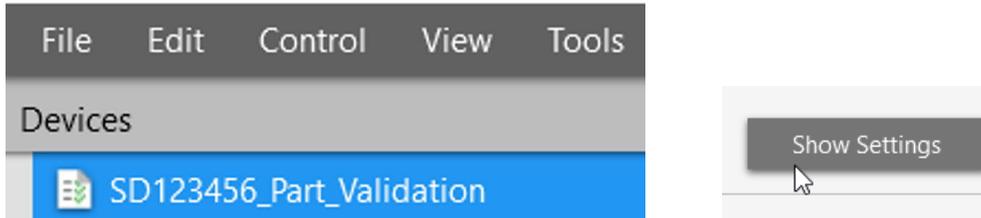


2. A dialog that is used to configure the new device connection will appear.
  - a. Enter the **Device Name** which is typically the machine SD number and 'Part\_Validation'. For example: SD123456\_Part\_Validation.
  - b. Set the device type to **Database Part Validator v2.0**
  - c. Click OK to finish adding the device connection.



#### 4.1.2 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.

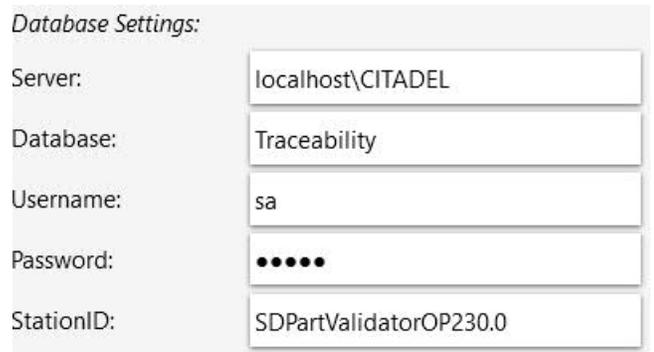


The following panel will be displayed. It is used to configure the Database Settings for the Record and the PLC connection settings.

<i>Database Settings:</i>	<i>Device Settings:</i>
Server: <input style="width: 100%;" type="text" value="localhost\CITADEL"/>	Name: <input style="width: 100%;" type="text" value="SD123456_Part_Validation"/>
Database: <input style="width: 100%;" type="text" value="Traceability"/>	IP Address: <input style="width: 100%;" type="text" value="10.199.110.93"/>
Username: <input style="width: 100%;" type="text" value="sa"/>	Tag Name: <input style="width: 100%;" type="text" value="TracePartValidator"/>
Password: <input style="width: 100%;" type="password" value="•••••"/>	Scan Rate: <input style="width: 100%;" type="text" value="100"/>
StationID: <input style="width: 100%;" type="text" value="SDPartValidatorOP230.0"/>	

#### 4.1.3 Configure Database Settings for the Part Validation Record

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. **StationID:** This contains the StationID of the record that the plugin will write to the Traceability\_status table.



#### 4.1.4 Configure Device Settings

1. **Name:** Typically set to the SD number of the machine + “\_Part\_Validation”.

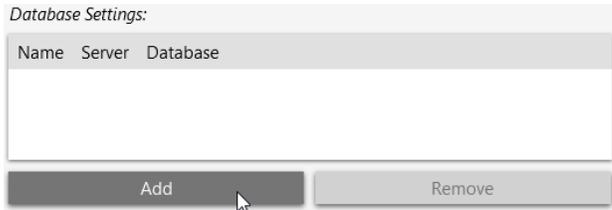
*Device Settings:*

Name:	SD123456_Part_Validation
IP Address:	0.0.0.0
Tag Name:	PartValidator
Scan Rate:	100

2. **IP Address:** Enter the IP Address of the PLC.
3. **Tag Name:** The default tag name is “PartValidator”. This is a user defined tag in the **R27\_Trace\_V2\_PartValidator** Logic routine containing the tag structure needed to interface with the traceability application.
4. **Scan Rate:** The default scan rate of how often the software polls the machine for new data is “100” milliseconds. This should not require modification.

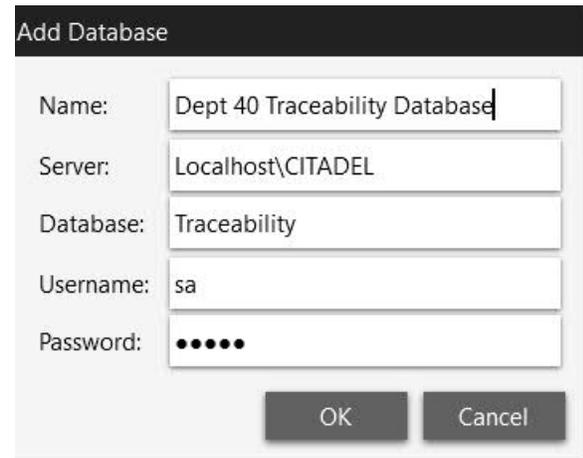
#### 4.1.5 Configure Database Settings

1. Select the Add button:

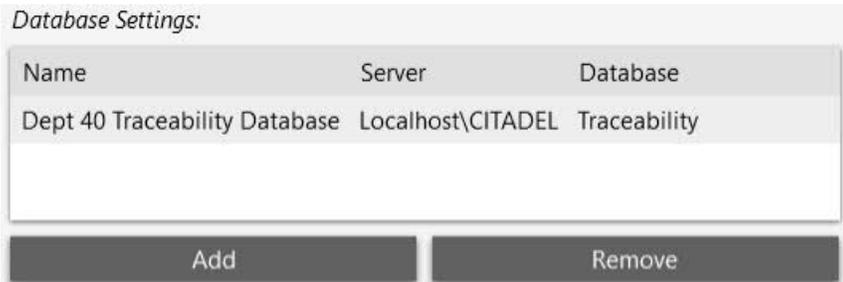


#### 4.1.6 Configure Database Settings

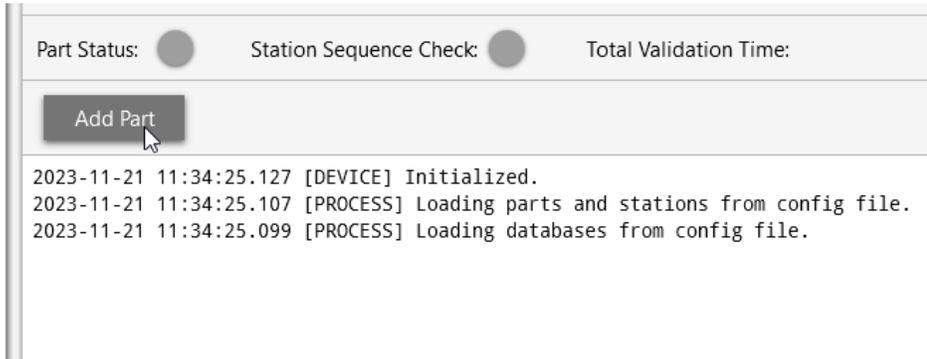
1. **Name:** The name of the database you are connecting too.
2. **Server:** The default Microsoft SQL server name is "localhost\CITADEL" which contains the standard Nexteer traceability database.
3. **Database:** The default database name is "Traceability".
4. **Username/Password:** This contains the credentials for the authorized database user. The default username is "sa" and password is "admin".



5. Add more database connections if the Parts records span more than one Database.



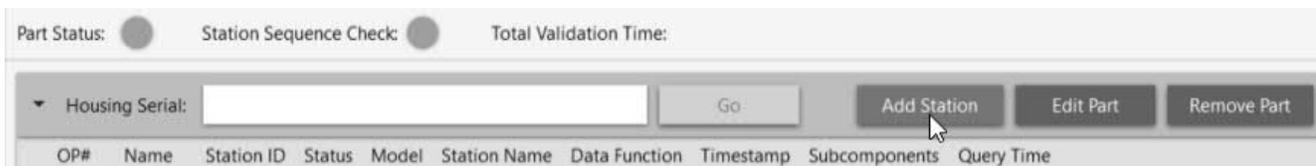
4.1.7 Select the 'Add Part' to add a component to the configuration.



4.1.8 Type in in Part name such as 'Housing' and select 'OK'



4.1.9 Once part is input select 'Add Station'



4.1.10 Input Station information

1. **Name:** The name of the component.
2. **OP# :** Operation number of the Station to verify component has passed
3. **Query Function :** Lookup function that will be used to return the record from the Database. Note: Function 2 is the default lookup function. If you require a special lookup function that is defined in the [dbo].[\_version2\_lookup\_custom] stored procedure you can use functions 100 thru 199.
4. **Accepted Stations:** Enter Station IDs that this station writes to the Database when the cycle is completed.  
  
Note: For parallel operations comma separate the Station IDs ex 'SD123456OP05A.0, SD123456OP05B.0'
5. **Accepted Statuses:** Enter Part Status that are accepted at this station. For more than one part status that is acceptable, comma separate the Statuses. Ex '9999,9500'
6. **Database:** Select Database where the record will exist.
7. **Part has Subcomponents checkbox:** Select this if you are adding Subcomponents to validate at this station.
8. **Subcomponents Serial Column(s):** If the Part Has Subcomponents box is checked, add the Data column in this record where the Subcomponent serial number exists.

Select 'OK' to finish this station configuration.

**Edit Station**

Name: Housing

OP#: OP010

Query Function: 2

Accepted Stations: SD123456OP010.0

Accepted Statuses: 9999

Database: Dept 40 Traceability Database

Part Has Subcomponent(s)

Subcomponent Serial Column(s):

OK Cancel

4.1.11 Repeat Steps 4.1.8 and 4.1.9 until all desired stations are configured for the component.

The screenshot shows two data tables in the interface. The top table is for 'Housing' components and the bottom table is for 'Motor' components. Both tables have columns for OP#, Name, Station ID, Status, Model, Station Name, Data Function, Timestamp, Subcomponents, and Query Time.

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP010	Housing								
OP020	Housing								
OP030	Housing								
OP040	Housing							Motor [Data000]	
OP050	Housing								

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP035	Motor								

4.1.12 For Stations that introduce Child components such as OP40 select the 'Part Has Subcomponents(s)' checkbox and enter the Data column where the serial number exists and also populate the Subcomponent Serial Column(s) data field. This data field is where the subcomponent serial number will be located.

The 'Edit Station' dialog box contains the following fields and controls:

- Name: Housing
- OP#: OP040
- Accepted Stations: SD123456OP040.0
- Accepted Statuses: 9999
- Database: Dept 40 Traceability Database
- Part Has Subcomponent(s)
- Subcomponent Serial Column(s): Data000
- Buttons: OK, Cancel

4.1.13 For Child components being introduced at any station, a record with a status of 9000 should exist in the Database as shown below.

**Edit Station**

Name:

OP#:

Accepted Stations:

Accepted Statuses:

Database:

Part Has Subcomponent(s)

Subcomponent Serial Column(s):

## 4.2 Traceability Application Testing

Note: The Traceability Validation Plugin must be in the ‘Stopped’ state to perform this test.

4.2.1 For testing that components have passed all the required operations, scan or enter the serial number of a component that has been through the entire define process sequence and select ‘GO’

Note: When using the testing option, no record will be saved to the Traceability database and if the serial number fails the check, a Blacklist record will not be written to the Blacklist table.

Part Status:  Station Sequence Check:  Total Validation Time:

---

Housing Serial:

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Qu
OP010	Housing								
OP020	Housing								
OP030	Housing								
OP040	Housing							Motor [Data000]	
OP050	Housing								

4.2.2 In this case since no records exist in the database for this serial number, the application will display the Query times and will also display errors in the application log window.

Housing Serial:

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP010	Housing								392ms
OP020	Housing								392ms
OP030	Housing								388ms
OP040	Housing							Motor [Data000]	387ms
OP050	Housing								387ms

Motor Serial:

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP040	Motor								

```

2024-02-15 10:54:43.762 [PROCESS] Record for StationID:SD1234560P50.0,Status:9999 not found
2024-02-15 10:54:43.760 [PROCESS] Record for StationID:SD1234560P040.0,Status:9999 not found
2024-02-15 10:54:43.759 [PROCESS] Record for StationID:SD1234560P020.0,Status:9999 not found
2024-02-15 10:54:43.758 [PROCESS] Record for StationID:SD1234560P30A.0,SD1234560P30B.0,Status:9999,9999 not found
2024-02-15 10:54:43.750 [PROCESS] Record not found for serial number 12345678 at station Housing.
2024-02-15 10:54:43.750 [PROCESS] Record not found for serial number 12345678 at station Housing.
2024-02-15 10:54:43.750 [PROCESS] Record not found for serial number 12345678 at station Housing.
2024-02-15 10:54:43.750 [PROCESS] Record not found for serial number 12345678 at station Housing.
  
```

4.2.3 Using the database lookup tool you can see that no records exist for this serial number.

Database Connection: Server: localhost\CITADEL Database: Traceability Username: sa Password: \*\*\*\*\*

Table Settings: Table: dbo\_status Sort Column: Timestamp Sort Order: Descending Record Limit: 1000

Column Filters: AND Serial = 12345678

Get Records (F5) No Refresh 0 rows returned.

ID	Timestamp	PLCID	StationID	DataFunction	Serial	Status	Model	StationName	Name000	Data000	Name001	Data001	Name002	Data002	Name003	Data003	Name004

4.2.4 In the case below, all the record with the correct Station IDs and Statuses have been found in the database. The application will show a green bar to the left when the criteria has been met for each station. The green ‘Station Sequence Check’ light will tun on when test is completed. The ‘Part Status’ light will turn green when the test is completed, and all criteria have been met when running a cycle connected to the PLC.

Part Status:  Station Sequence Check:  Total Validation Time:

Housing Serial: 12345678

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP010	Housing	SD123456OP010.0	9999	Test Model	Test Station	10	2024-02-15 11:00:18.737		3ms
OP020	Housing	SD123456OP020.0	9999	Test Model	Test Station	10	2024-02-15 11:00:27.447		46ms
OP030	Housing	SD123456OP030B.0	9999	Test Model	Test Station	10	2024-02-15 11:00:55.747		3ms
OP040	Housing	SD123456OP040.0	9999	Test Model	Test Station	10	2024-02-15 11:02:10.287	Motor [Data000]	3ms
OP050	Housing	SD123456OP050.0	9999	Test Model	Test Station	10	2024-02-15 11:02:23.047		46ms

Motor Serial: 23456789

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP040	Motor	SD123456OP040.1	9000	Motor	Test Station	10	2024-02-15 11:09:48.567		42ms

4.2.5 Note the lookup tool results for this serial number below showing that all criteria have been met for all the stations.

ID	Timestamp	PLCID	StationID	DataFunction	Serial	Status	Model	StationName	Name000	Data000
7049	2024-02-15 11:02:23.047	DataGenerator	SD123456OP050.0	10	12345678	9999	Test Model	Test Station		
7050	2024-02-15 11:02:11.567	DataGenerator	SD123456OP040.1	10	23456789	9000	Motor	Test Station		
7048	2024-02-15 11:02:10.287	DataGenerator	SD123456OP040.0	10	12345678	9999	Test Model	Test Station	Motor Serial	23456789
7047	2024-02-15 11:00:55.747	DataGenerator	SD123456OP030B.0	10	12345678	9999	Test Model	Test Station		
7046	2024-02-15 11:00:27.447	DataGenerator	SD123456OP020.0	10	12345678	9999	Test Model	Test Station		
7045	2024-02-15 11:00:18.737	DataGenerator	SD123456OP010.0	10	12345678	9999	Test Model	Test Station		

4.2.6 In the case below Station 30 timestamp is before Station 20 Timestamp so the red ‘Sequence Check’ light will come on and the log will display which stations are out of order. Since Station 20 is out of order all subsequent station will be out of order as shown in the Log display.

Part Status:  Station Sequence Check:  Total Validation Time:

Housing Serial: 12345678

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP010	Housing	SD123456OP010.0	9999	Test Model	Test Station	10	2024-02-15 11:00:18.737		193ms
OP020	Housing	SD123456OP020.0	9999	Test Model	Test Station	10	2024-02-15 11:00:27.447		197ms
OP030	Housing	SD123456OP030B.0	9999	Test Model	Test Station	10	2024-02-15 11:00:00.747		196ms
OP040	Housing	SD123456OP040.0	9999	Test Model	Test Station	10	2024-02-15 11:02:10.287	Motor [Data000]	197ms
OP050	Housing	SD123456OP050.0	9999	Test Model	Test Station	10	2024-02-15 11:02:23.047		198ms

Motor Serial: 23456789

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP040	Motor	SD123456OP040.1	9000	Motor	Test Station	10	2024-02-15 11:02:11.567		4ms

```

2024-02-19 11:16:27.196 [PROCESS] Sequence fail for StationID:SD123456OP040.0,Status:9999, Sequence: 3, Timestamp: 2/15/2024 11:02:10 AM
2024-02-19 11:16:27.196 [PROCESS] Sequence fail for StationID:SD123456OP040.1,Status:9000, Sequence: 0, Timestamp: 2/15/2024 11:02:11 AM
2024-02-19 11:16:27.193 [PROCESS] Sequence fail for StationID:SD123456OP050.0,Status:9999, Sequence: 4, Timestamp: 2/15/2024 11:02:23 AM
2024-02-19 11:16:27.192 [PROCESS] Sequence fail for StationID:SD123456OP020.0,Status:9999, Sequence: 1, Timestamp: 2/15/2024 11:00:27 AM
2024-02-19 11:16:27.191 [PROCESS] Sequence fail for StationID:SD123456OP030B.0,Status:9999, Sequence: 2, Timestamp: 2/15/2024 11:00:00 AM
2024-02-19 11:09:46.507 [PROCESS] Record for StationID:SD123456OP040.1,Status:9000 not found
  
```

4.2.7 To Start the Plugin press the Start button. A green running light will appear, and the green activity light will flash. In the bottom status window, you will see that the process has started.

SD123456\_Part Validation Database Part Validator v2.0

Activity:  Scan Rate: 107ms

Part Status:  Station Sequence Check:  Total Validation Time:

Housing Serial:

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP010	Housing								
OP020	Housing								
OP030	Housing								
OP040	Housing							Motor [Data000]	
OP050	Housing								

Motor Serial:

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP040	Motor								

```

2024-02-19 11:32:05.880 [PROCESS] Process started.
2024-02-19 11:32:05.866 [PROCESS] Device Serial: 60AA2B57
2024-02-19 11:32:05.866 [PROCESS] Device Firmware: 32.012
2024-02-19 11:32:05.866 [PROCESS] Device Model: 5069-L306ER/A
2024-02-19 11:32:05.710 [PROCESS] Connection successful.
2024-02-19 11:32:05.672 [PROCESS] Connecting to device: 10.199.110.91
  
```

4.2.8 When the PLC has initiated a component check you will see Part status green light and Station Sequence check light as green indicating that the Component has passed all station checks and the records are in the correct sequence.

**SD123456\_Part\_Validation** Database Part Validator v2.0

Running Start Stop Activity: Scan Rate: 107ms

Part Status: Station Sequence Check: Total Validation Time: 67ms

Housing Serial: 12345678 Go

OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP010	Housing	SD123456OP010.0	9999	Test Model	Test Station	10	2024-02-15 11:00:18.737		22ms
OP020	Housing	SD123456OP020.0	9999	Test Model	Test Station	10	2024-02-15 11:00:27.447		14ms
OP030	Housing	SD123456OP030B.0	9999	Test Model	Test Station	10	2024-02-15 11:00:50.747		21ms
OP040	Housing	SD123456OP040.0	9999	Test Model	Test Station	10	2024-02-15 11:02:10.287	Motor [Data000]	21ms
OP050	Housing	SD123456OP050.0	9999	Test Model	Test Station	10	2024-02-15 11:02:23.047		26ms

Motor Serial: 23456789 Go

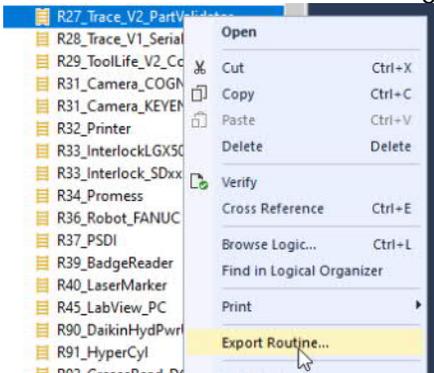
OP#	Name	Station ID	Status	Model	Station Name	Data Function	Timestamp	Subcomponents	Query Time
OP040	Motor	SD123456OP040.1	9000	Motor	Test Station	10	2024-02-15 11:02:11.567		16ms

2024-02-19 11:37:22.378 [PROCESS] Validation for serial number 12345678 passed in 67ms.

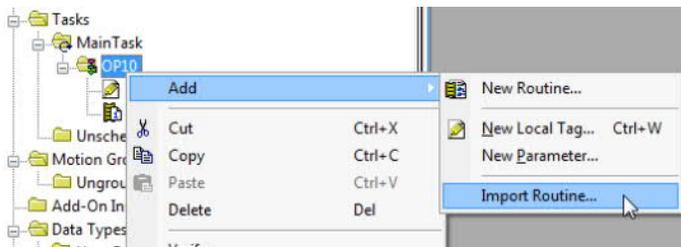
### 4.3 Logic Configuration (R27\_Trace\_V2\_PartValidator)

This routine is used for traceability part validation. A uniquely named routine shall be configured for each device defined in the traceability application. The logic in this routine shall be modified to meet the requirements of the application.

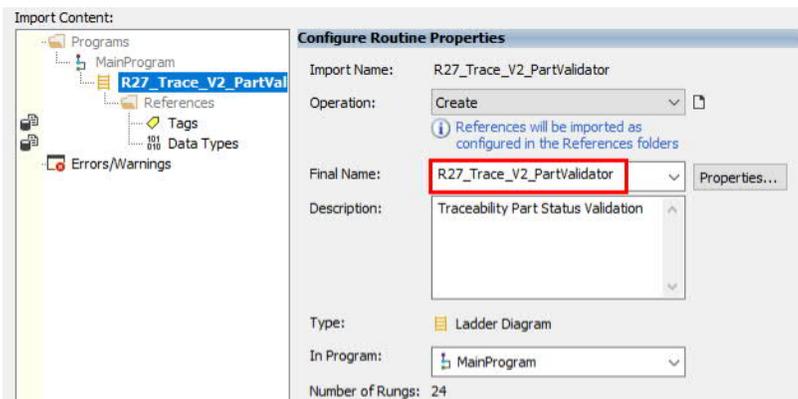
#### 4.3.1 Export the R27\_Trace\_V2\_PartValidator from the Nexteer PLC library program by right clicking on the routine and clicking **Export Routine...** Save the file to a location on your hard drive.



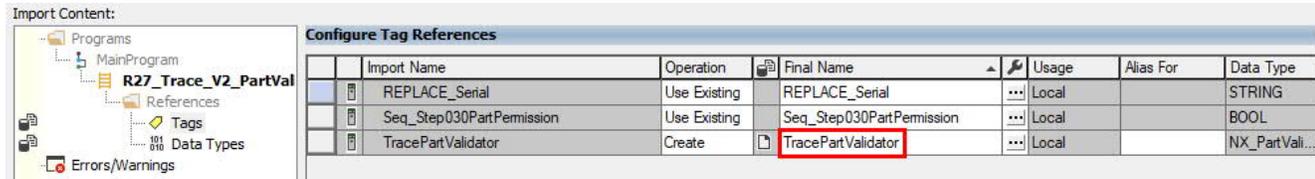
1. Right click on the main program that **R27\_Trace\_V2\_PartValidator** needs to be imported into (Typically you will only have one Part Validator routine per PLC unless the number of stations is excessive) and click **Add > Import Routine** in the context menu. Browse to the location that exported routine was saved to and click **Open**.



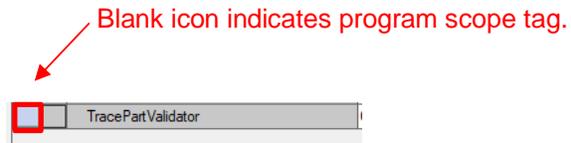
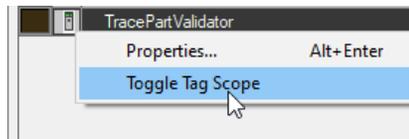
2. The **Import Configuration** dialog will appear. Ensure that **Operation** is set to **Create** by entering a unique value for the **Final Name** setting. Typically, the name should be "R27\_Trace\_PartValidator" + Name of the Master Component specified in the Traceability Input Document.



- Click on the **Tags** option in the **Import Content** tree. There is 1 tag that may need to be renamed prior to importing the routine if there are more than one **R27\_Trace\_V2\_PartValidator** routines on the same PLC.

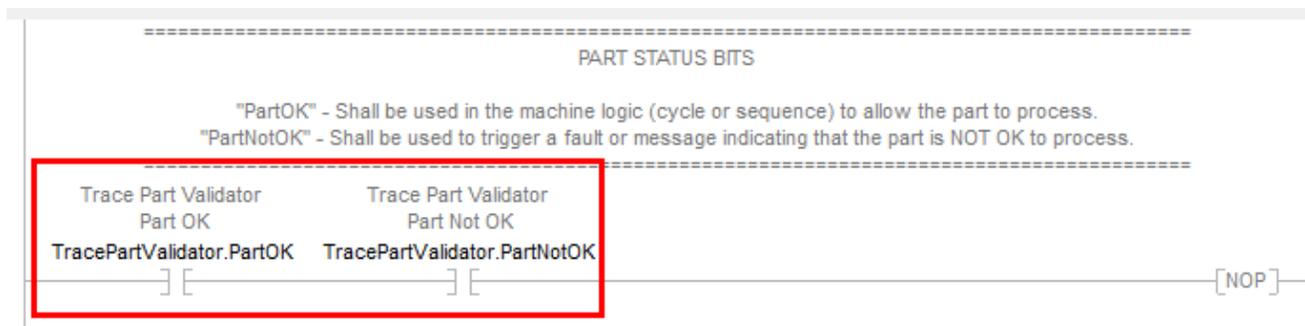


- For a single PLC with many stations, the default tag name 'TracePartValidator' can be used.
- For special cases, where multiple **R27\_Trace\_V2\_PartValidator** routines are to be imported, the tag name 'TracePartValidator' can be changed to match the operation where it will be used ("OPxxx\_TracePartValidator" or "STxxx\_TracePartValidator")
- The **TracePartValidator** tag is also allowed to be Controller or Program scope. The scope of the tag can be changed by right clicking the  icon and choosing **Toggle Tag Scope** in the context menu.

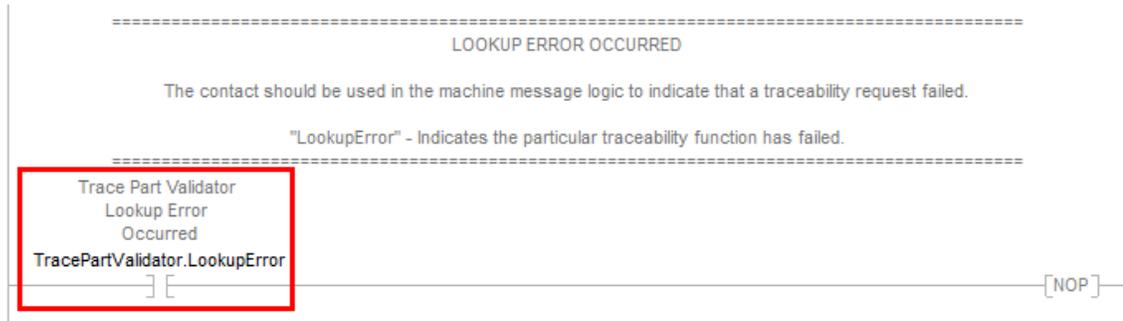


#### 4.3.2 General Configuration

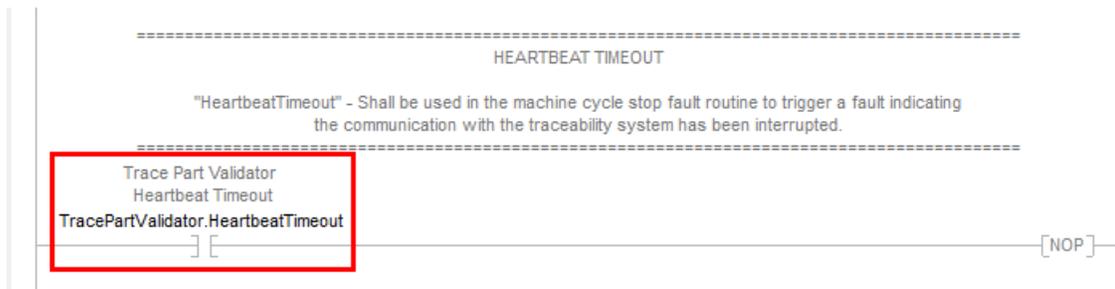
- The **TracePartValidator.PartOK** tag shall be used in the sequence routine to allow the sequence of the machine to continue processing the part and shall also be used to control the part status message display to indicate the part is ok to run.
- The **TracePartValidator.PartNotOK** tag shall be used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



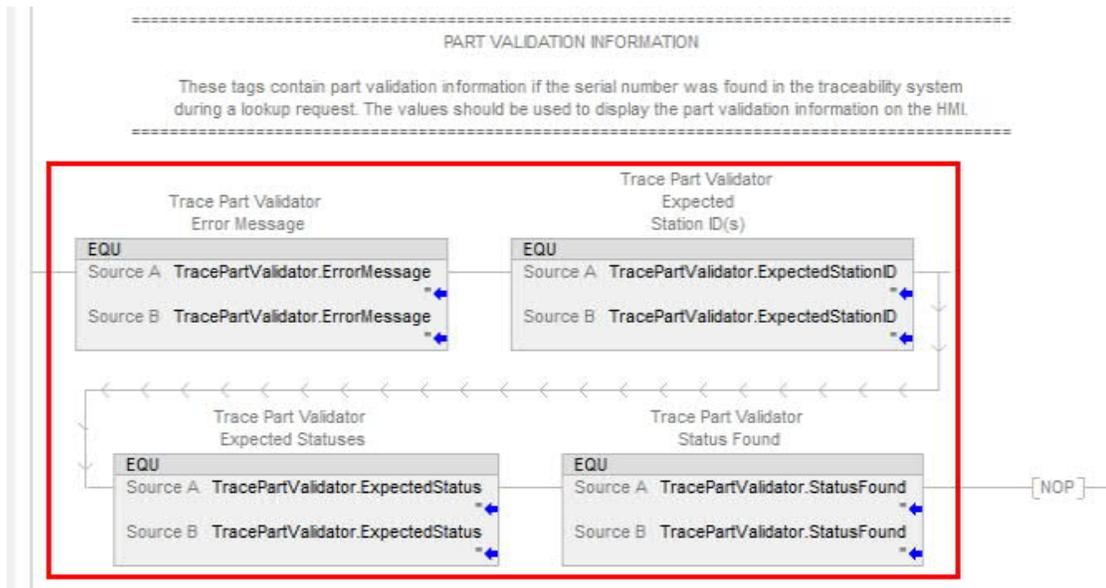
- The **TracePartValidator.LookupError** tag is used to indicate that a traceability request failed. This tag shall be used in the **R08c\_Fault\_CycleStop** routine to inhibit machine cycling. A loss of communication should prohibit the next cycle from initiating.



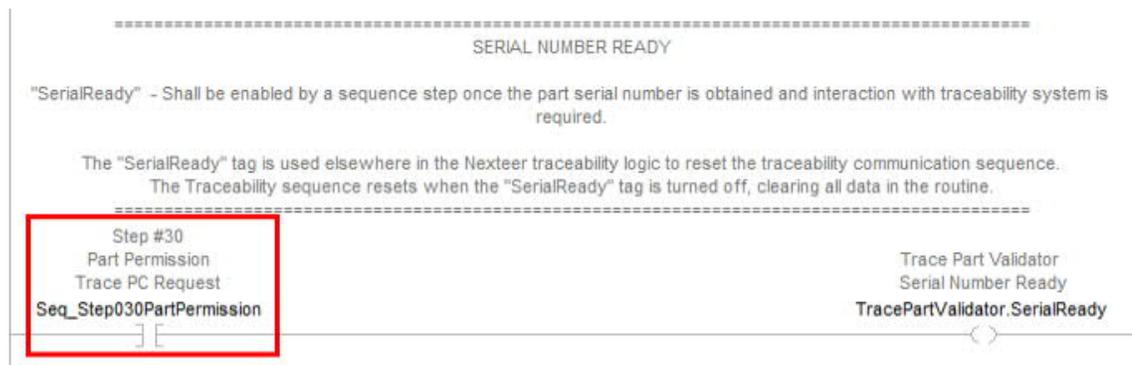
- The **TracePartValidator.HeartbeatTimeout** tag is used to signal a loss of communication with the Trace PC and shall be used in the **R08c\_Fault\_CycleStop** routine to inhibit machine cycling. A loss of communication should prohibit the next cycle from initiating.



- The **TracePartValidator.ErrorMessage** tag is used to signal any errors that occurred during the Part Validation lookup process. The **TracePartValidator.ExpectedStationID** tag is used to signal which Station ID failed the validation process. The **TracePartValidator.ExpectedStatus** and **TracePartValidator.StatusFound** tags are used to signal what part status was found on the failed Station ID and what is the expected part status to pass on the failed Station ID.



- The **TracePartValidator.SerialReady** coil shall be turned on by a step in the sequence logic after a serial number has been placed in the **TracePartValidator.Serial** tag.



- The scanned serial number shall be copied to the `TracePartValidator.Serial` tag prior to turning on the `TracePartValidator.SerialReady` coil.

=====

SERIAL NUMBER

"REPLACE\_Serial" - Shall be replaced with the appropriate tag containing the part serial number.

The serial number is the unique identifier for the part being processed, and must be obtained from a valid source such as a barcode reader, RFID tag, or pallet memory map.

=====

Trace Part Validator Serial Number	
COP	
Source	REPLACE_Serial
Dest	TracePartValidator.Serial
Length	1

- The `R27_Trace_V2_PartValidator` shall not be edited below the phrase "DO NOT EDIT ANYTHING BELOW THIS RUNG"

=====

!! IMPORTANT !!

DO NOT EDIT ANYTHING BELOW THIS RUNG

=====

[NOP]

## 5 Test Cycle Application Technical Notes

### Note: Requires Traceability Application Version 12.0.9 (Serialized Part Required)

The Test Cycle is used in cases where qualified personnel can run a cycle without the checking for Traceability permission to run a cycle. Traceability is bypassed and the serial number that is scanned will be added to the Database Blacklist Table to ensure the part cannot run at subsequent stations.

#### 5.1 Nexteer Standard 'Test Cycle' Process Sequence Steps:

1. Engineer is Logged In ('Model' login password is used to access 'Test Cycle' PB)
2. 'Test Cycle' Enable PB Pressed – HMI (Cannot activate test enable if in cycle)
  - o Timeout if Cycle Not Initiated within 60 seconds
3. Part is Loaded
4. Start of Machine Cycle
5. Part serial is Scanned
6. Traceability is Enabled, (Park Ok status forced on in Logic)
7. **Lookup:** Trace app will **not** do a lookup in the DB but will send a Test Cycle 'in-process' record of '4900' to the Database, and then send the **Lookup Complete** signal to the PLC.
8. **In-Process:** Trace app will not send a record to traceability DB, but will send **In-Process Complete** to PLC
9. **Send Results:** Will override the status by changing the first digit to a '4' Examples '9999' becomes '4999', '1010' becomes '4010' etc. and then send a **Results Complete** signal to PLC
10. Normal Machine Cycle Indicators (Green, Yellow, and Red lights)
11. 'Test Cycle' Disabled at End of Cycle, Change of Mode, or Fault Condition (Allowed for 1 cycle only)
12. Part is Unloaded
13. Use a color-coded tag (Green, Red, Yellow) with descriptions that are attached to the parts that are ran with 'Test Cycle'.
14. Parts are not allowed to be recovered for production if ran under 'Test Cycle'.
15. NOTE: Once 'Test Cycle' is started and the PLC does a 'Lookup, In Process or Send Results' the Trace Application will add the serial number to the blacklist table

#### 5.2 Trace Application Notes:

1. 'Test Cycle' must be enabled before **Trace Active** coil is on.
2. Use Trace Station routine to enable Part ok in 'do not touch area' logic.
3. No changes to LOT Tracking functionality.
4. If a part is only going to run at a specific station, then that station can use the HMI to enable 'Test Cycle' to run the part at that station.
5. Location of 'Test Cycle' PB: Shall be on the Traceability screen.
6. 'Test Cycle' Enabled Message/Prompt: Machine Message banner and Operator Prompt message displayed on the HMI when 'Test Cycle' is enabled.
7. **'Test Cycle' Timeout Display:** 'Test Cycle' timeout value display to the Enable/Disable button as well as make it a different color or flash while enabled.
8. **Notes: Version 12.0.9 or later of Traceability is required to work with 'Test Cycle'**

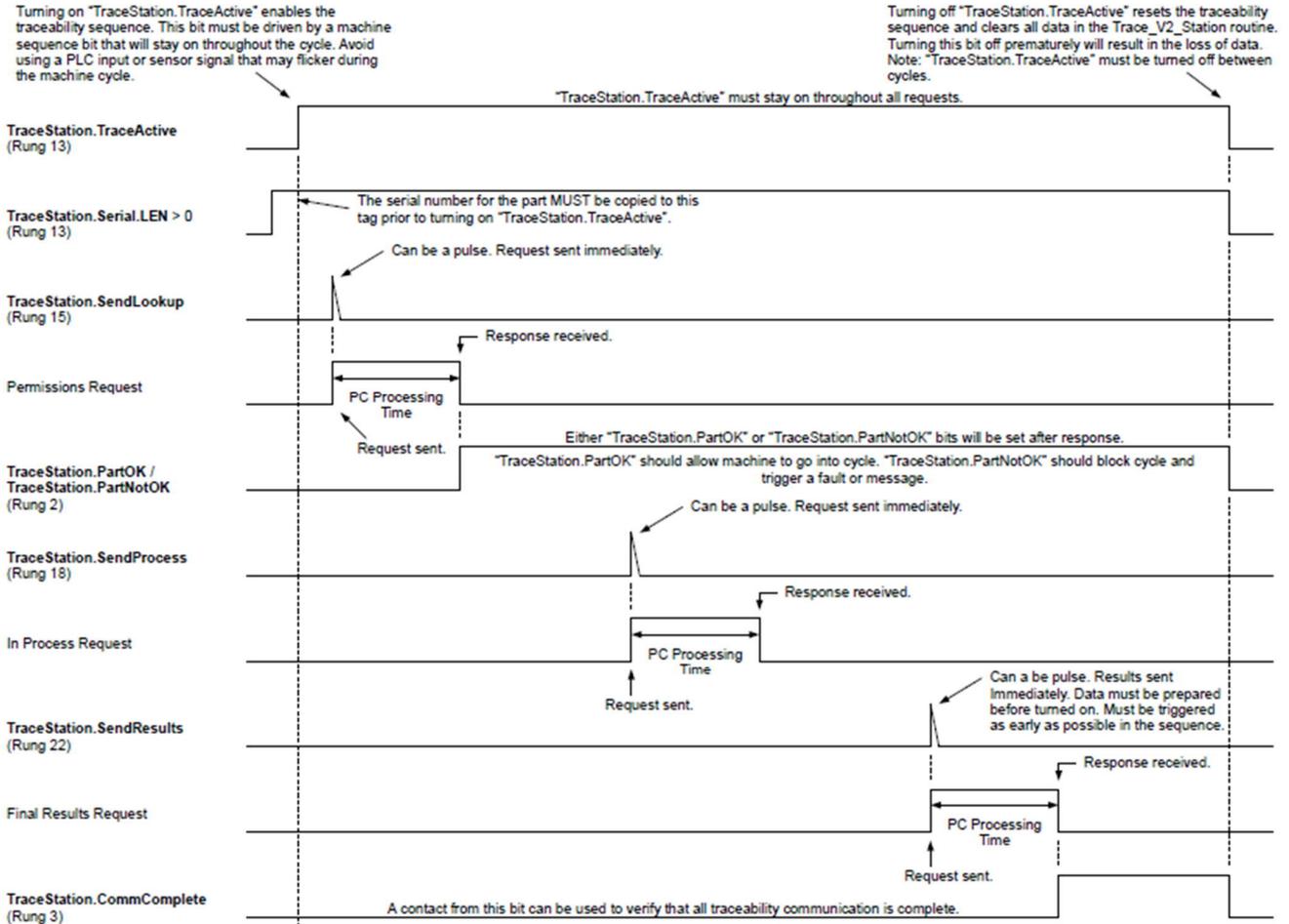
#### 5.3 Test Cycle HMI Button



## A. Traceability Timing Chart

### Trace\_V2\_Station routine sequence timing guide

Last Updated: July 26<sup>th</sup>, 2018  
By: Adam Romzek



## B. Traceability Field Guidelines

Timestamp,PLCID,StationID,DataFunction,SerialNumber,Status,Model,StationName,Name000,Data000,Name001,Data001

Cat	Field Name	Example	Comments
Header	Timestamp	2015-06-26 09:06:57.007	<p>“Timestamp” name may not be used for any other field.</p> <p>Time values should differ for each record.</p> <p>All PLC’s and PC’s should be synchronized to a time server to ensure accurate timestamps (reference Annex F for syncing to a timeserver PC).</p>
	PLCID	SD800788X	<p>“PLCID” name may not be used for any other field.</p> <p>Must begin with “SD”.</p>
	StationID	SD800788X01	<p>“StationID” and “StationOP” names may not be used for any other field.</p> <p>Must begin with “SD”.</p> <p>Must be unique – verify after copying PLC Code.</p> <p>Typically, 11 characters in length.</p> <p>StationID is typically referenced on the Manufacturing Sequence chart.</p>
	DataFunction	10	<p>“DataFunction” name may not be used for any other field (reference Annex B for description of all Data Functions).</p>
	SerialNumber	38017975151762436302	<p>“SerialNumber” name may not be used for any other field.</p> <p>Serial Numbers cannot be blank (the text “No Serial” should be used if the serial scan failed).</p> <p>Format depends on the value stream (reference process sheet where format can be retrieved).</p> <p>Some value streams use Direct Part Marking Spec (34000869) to specify the format. The Plant will provide format for master parts, these can be different length and/or format.</p> <p>SerialNumbers may be referenced on manufacturing sequence chart.</p> <p>Component Serial Numbers will not be displayed in this column; they will be one of the early “Data” fields.</p> <p>Only primary and assembly serial numbers will be displayed in this column.</p> <p>Serial Numbers that include Julian dates must have values within range of 1 – 366.</p> <p>Serial Numbers that include shift must have values within 1 – 3.</p> <p>Serial Numbers that include the year must have two digits.</p>
	Status	9999	<p>“Status” and “PLC_Status” names may not be used for any other field.</p> <p>Status Codes may be included on the Manufacturing Sequence Chart (reference Annex E for Status Code Examples).</p>
	Model (Part Number)	L RWD LDLA BASE LHD 05154744	<p>Since part numbers change regularly, it is preferable to use text to describe the Model.</p> <p>“Model” name may not be used for any other field.</p> <p>Plant will provide standardized list of model descriptions. This should be documented on the manufacturing sequence chart.</p> <p>Model descriptions will be consistent from one station to another.</p> <p>Model descriptions should be descriptive.</p> <p>If a Model description is not required, the tag should be populated with “N/A.”</p>

	StationName	084-BSI-Line1-Sta220-OP20 to210 Pallet Data	<p>"StationName" name may not be used for any other field.</p> <p>Should include the Three Digit Department Number indicated on the Manufacturing Sequence Chart.</p> <p>Should include the Cell or Line Name (e.g. BSI, Grey Room, Rack, Ball Nut, etc.).</p> <p>Should include the Cell or Line Designation: "-Cell" or "-Line" plus number (one digit) (e.g. 1, 2, 3, ...) or letter (e.g. A, B, C, ...).</p> <p>Could include a three digit Station Number preceded by "-Sta" (e.g. -Sta220).</p> <p>Could include a three digit Operation Number specified on the Manufacturing Sequence Chart preceded by "-OP".</p> <p>Could include a range of three digit Operation Numbers specified on the Manufacturing Sequence Chart preceded by 'OP" (e.g. - OP20 to 210).</p> <p>Should include a high level description (e.g. Pallet Data).</p>
Name / Data	Name[xxx]	RackSN Magnet Press Distance [mm]	<p>No field name is duplicated.</p> <p>Field names are consistent between lines containing same data.</p> <p>Field names include units in brackets (e.g. [N]).</p> <p>Field names are spelled correctly.</p> <p>Cannot use "FTQ_Results", "FTQ_Desc", "FTQ_FirstTime", "FTQ_PreviousRun", "FTC".</p> <p>No field is named a header field (e.g. "Status").</p>
	Data[xxx]	38010205151761061501	<p>Data content from one filed does not conflict with another, e.g. BSI – cannot have a good status with a Fail Code.</p> <p>Where field content is expected to be variable, such as a press load, indicate this as "Variable Values" on the Manufacturing Sequence Chart.</p> <p>Where field content is supposed to be within a range, such as Pallet Numbers, ensure that the data varies within that range. Indicate this as "Range of Values" on the Manufacturing Sequence Chart.</p> <p>Where field content contains fixed values, such as reading NTC Codes from a part, ensure that the data is consistent. Indicate this as "Fixed Values" on the Manufacturing Sequence Chart.</p> <p>If this part is not required for the current running model, populate this tag with "N/A."</p> <p>All fields are populated regardless of errors. If there is no data to report, then display "No Data" for text fields and 0 for Numeric fields.</p>

**RECORD OF REVISIONS**

Revision No	Date	Section	Description
001	27AU21	All	Original Approval & Issue Data
002	09JA25	4,5	Added Sections 4 and 5
003			
004			
005			
006			
007			
008			
009			
010			
011			
012			
013			
014			
015			
016			
017			
018			
019			
020			