Sending Native Mode Commands using EIP Explicit Messaging

Author: Glenn Mushall
Samantha Frost

Published: January 3, 2018
Revision: 1.0

Cognex Corporation

Overview

PCCC commands are used to send Native Mode commands to In-Sight systems running 4.x firmware. The next generation In-Sight vision systems (IS8xxx, IS2xxx, IS75xx-79xx and IS5705) running 5.x firmware do not support PCCC commands embedded in Ethernet PLC Read/Write MSG instructions. However, there is another method that supports sending native mode commands from the PLC to the In-Sight: EIP Explicit Messaging.

Benefits

In-Sight Native Mode protocol is an ASCII protocol that allows an In-Sight system to be controlled from an external source such as a PLC. Native Mode commands include controls like: setting a value in the job, getting a value in the job, and loading a different job file. Please reference the In-Sight help files for a complete list of Native Mode Commands.

Sending Native Mode Commands using EtherNet/IP CIP Generic messaging

Rockwell MicroLogix PLC's with newer FW/OS (MicroLogix 1100 Series B and MicroLogix 1400 Series B, Rev A, FRN4), SLC5/05, Micro800 series (820 & 850), CompactLogix and ControlLogix all support CIP explicit messaging via the MSG or EEM instruction using a message type of CIP Generic. This allows common service attribute access (get/set) and object specific service access to the Vision object (**0x78**) in the In-Sight vision system using EtherNet/IP explicit messaging. This is how the native mode commands are sent to In-Sight.

Custom service code 0x34 is the code for the SendNativeCmd service which allows the PLC to send Native Mode commands to the In-Sight vision system. The SendNativeCmd Service sends any documented Native Mode command string to the In-Sight vision system. The request data for the SendNativeCmd should include the Native Mode command string that is to be sent to the In-Sight vision system; the reply data will contain the string result of the Native Mode command if the command returns a string result (unless an XML response is sent, depends on PLC platform).

The Vision Object (class 0x78) provides the following Object Specific Services:

Service Code	Service Name	Description of Service
0x32	Acquire	Triggers a single acquisition.
0x33	Reserved	Deprecated.
0x34	SendNativeCmd	Sends a Native Mode command to the In-Sight vision system.
0x35	GetInspectionResults	Gets the InspectionResults attribute.
0x36	Transfer UserData	Transfers data into a portion of the UserData Holding Buffer.
0x37	Set UserData	Indicates to the vision system that it should latch the User Data

Service Code	Service Name	Description of Service
		Holding Buffer into the User Data field to give the In-Sight access to the user data.

The configuration on the PLC side to accomplish this is somewhat different depending on the PLC platform being used. The samples that follow were written using a MicroLogix 1100, SLC5/05 and ControlLogix L71. They perform a certain set of functions using native mode commands, one of which can only be accomplished using native mode commands (no service or attribute).

The following commands are stored in a string file on the PLC and then the logic indexes through them and sends the command for that index to the In-Sight vision system:

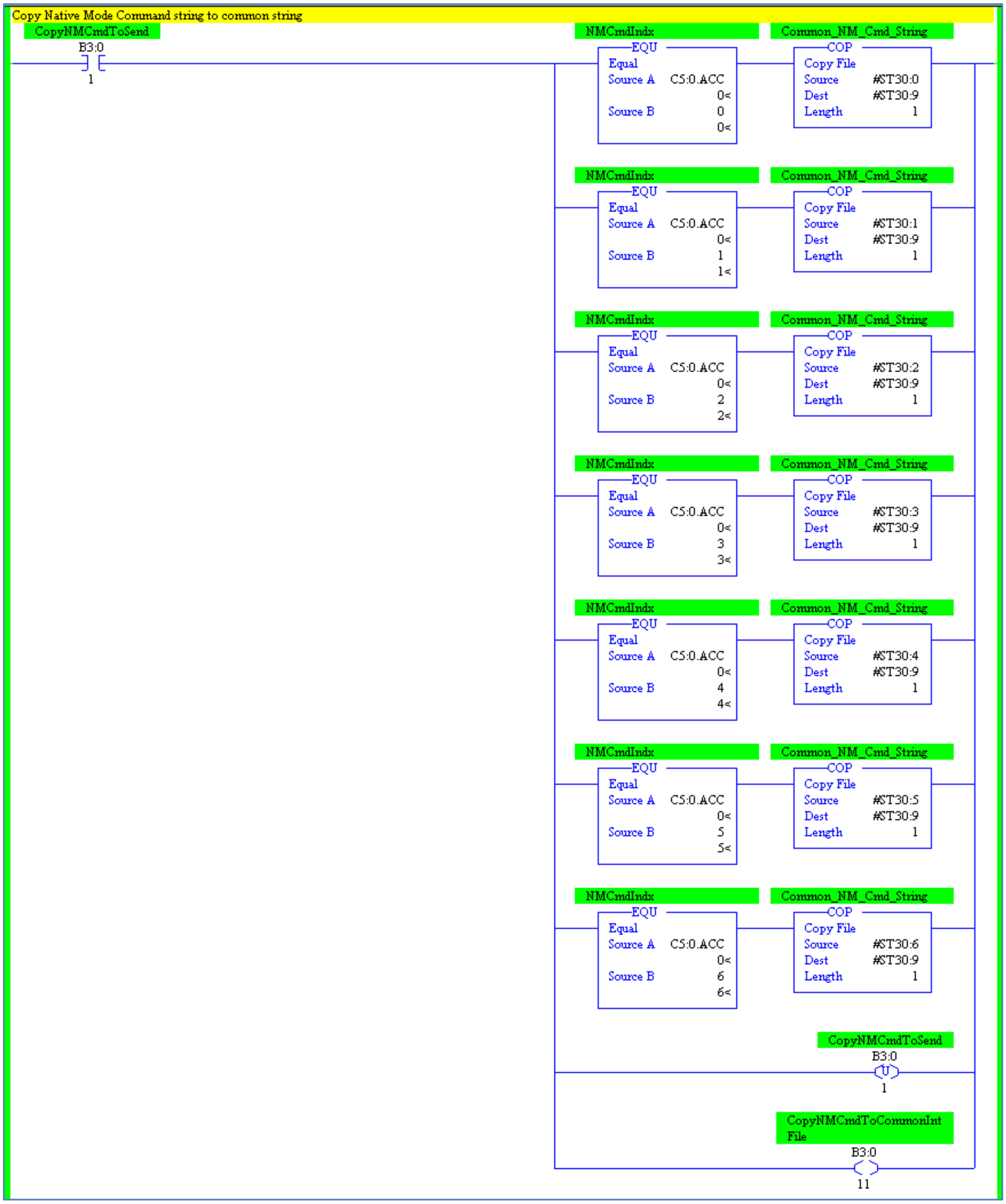
Commands	ST File address	Index C5:0.ACC	Functionality/Behavior
SO0	ST30:0	0	Set/Force In-Sight offline ("Comms Online?" status shown in ISE)
GO	ST30:1	1	Get Online (returns "0" since not online)
TFJobName-Stored.job	ST30:2	2	Saves loaded job in flash using the name "JobName-Stored.job" and it becomes the active job. Job will be in file list when refreshed in ISE.
GF	ST30:3	3	Returns the file name of the active job - "JobName-Stored.job"
LFJobName.job	ST30:4	4	Loads the previous version of the job
SO1	ST30:5	5	Set/Force online ("Online" status shown in ISE)
GO	ST30:6	6	Get Online (returns "1" since online)

Setting the bit CopyNMCmdToSend in the first rung in all samples will write the native mode command stored at the index of the string in the string file (ST30).   The bit is cleared (unlatched) after copying the command string, sending the command (MSG/EEM), return a result string if the native mode command returns a response and increments the string file element index (C5:0).   The bit in rung 0 should be set again to send the next native mode command. The index C5:0 increments each time until it reaches 7 (meaning 7 native mode commands have been sent, index = 0-6) and then it resets back to 0.

The following sections describe the ladder functionality for each platform. Sample code is included in the zip file this pdf is extracted from. Following that are steps to install the library files containing the rungs into a new or existing PLC project.

MicroLogix 1100:

Rung 0 - Summary: This rung copies the native mode command strings stored in ST30:0 – ST30:6 based on index C5:0 to common location ST30:9. Each time the user sets bit B3:0/1 (CopyNMCmdToSend) and the index was incremented the next command is copied to the common string storage location.

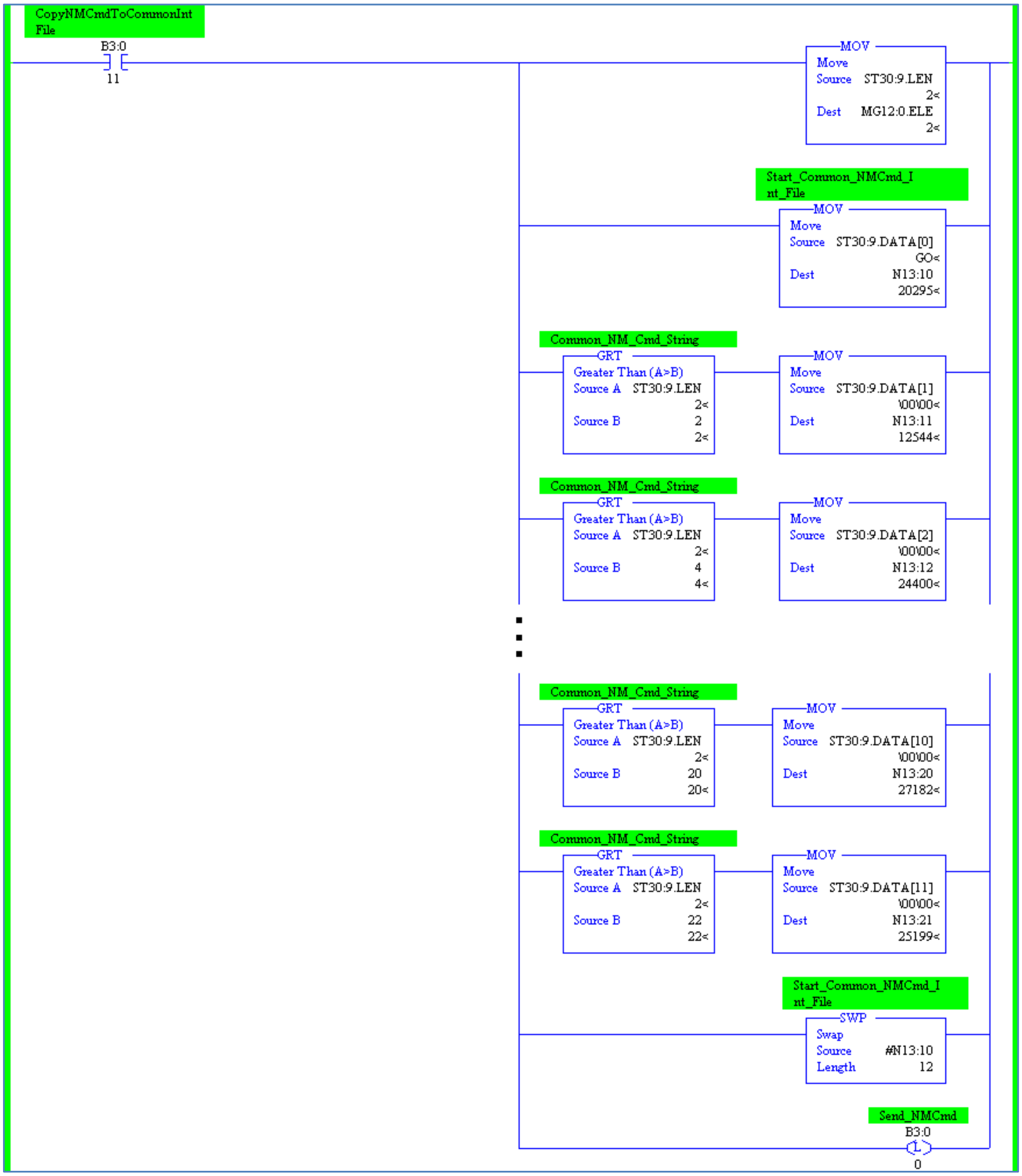


```
Copy Native Mode Command string to common string

CopyNMCmdToSend
B3:0
] [
1

NMCmdIndx                Common_NM_Cmd_String
EQU                      COP
Equal                    Copy File
Source A   C5:0.ACC      Source    #ST30:0
                 0<      Dest      #ST30:9
Source B         0       Length          1
                 0<

NMCmdIndx                Common_NM_Cmd_String
EQU                      COP
Equal                    Copy File
Source A   C5:0.ACC      Source    #ST30:1
                 0<      Dest      #ST30:9
Source B         1       Length          1
                 1<

NMCmdIndx                Common_NM_Cmd_String
EQU                      COP
Equal                    Copy File
Source A   C5:0.ACC      Source    #ST30:2
                 0<      Dest      #ST30:9
Source B         2       Length          1
                 2<

NMCmdIndx                Common_NM_Cmd_String
EQU                      COP
Equal                    Copy File
Source A   C5:0.ACC      Source    #ST30:3
                 0<      Dest      #ST30:9
Source B         3       Length          1
                 3<

NMCmdIndx                Common_NM_Cmd_String
EQU                      COP
Equal                    Copy File
Source A   C5:0.ACC      Source    #ST30:4
                 0<      Dest      #ST30:9
Source B         4       Length          1
                 4<

NMCmdIndx                Common_NM_Cmd_String
EQU                      COP
Equal                    Copy File
Source A   C5:0.ACC      Source    #ST30:5
                 0<      Dest      #ST30:9
Source B         5       Length          1
                 5<

NMCmdIndx                Common_NM_Cmd_String
EQU                      COP
Equal                    Copy File
Source A   C5:0.ACC      Source    #ST30:6
                 0<      Dest      #ST30:9
Source B         6       Length          1
                 6<

                         CopyNMCmdToSend
                         B3:0
                         (U)
                         1

                         CopyNMCmdToCommonInt
                         File
                         B3:0
                         ( )
                         11
```

Details: Setting the B3:0/1 (CopyNMCmdToSend) bit will copy one of the commands in the string file, depending on the index 0 – 6 from the String file ST30:0 – ST30:6 to the common ST30:9 location. Note that indexed addressing (#) is used in the COP since it is a file operation.

Once the correct COP executes the B3:0/1 (CopyNMCmdToSend) bit is unlatched and the B3:0/11 (CopyNMCmdToCommonString) bit is set.

<u>Rung 1 - Summary:</u> The currently selected native mode command is copied 2 ASCII values at a time into an integer file (N13) and the bytes are then swapped in the integer file.



```
CopyNMCmdToCommonInt File
B3:0/11  -] [-

MOV  Move
  Source  ST30:9.LEN   2<
  Dest    MG12:0.ELE   2<

Start_Common_NMCmd_Int_File
MOV  Move
  Source  ST30:9.DATA[0]   GO<
  Dest    N13:10           20295<

Common_NM_Cmd_String
GRT  Greater Than (A>B)
  Source A  ST30:9.LEN   2<
  Source B  2            2<
MOV  Move
  Source  ST30:9.DATA[1]   \00\00<
  Dest    N13:11           12544<

Common_NM_Cmd_String
GRT  Greater Than (A>B)
  Source A  ST30:9.LEN   2<
  Source B  4            4<
MOV  Move
  Source  ST30:9.DATA[2]   \00\00<
  Dest    N13:12           24400<

:

Common_NM_Cmd_String
GRT  Greater Than (A>B)
  Source A  ST30:9.LEN   2<
  Source B  20           20<
MOV  Move
  Source  ST30:9.DATA[10]   \00\00<
  Dest    N13:20            27182<

Common_NM_Cmd_String
GRT  Greater Than (A>B)
  Source A  ST30:9.LEN   2<
  Source B  22           22<
MOV  Move
  Source  ST30:9.DATA[11]   \00\00<
  Dest    N13:21            25199<

Start_Common_NMCmd_Int_File
SWP  Swap
  Source  #N13:10
  Length  12

Send_NMCmd
B3:0/0  -(L)-
```

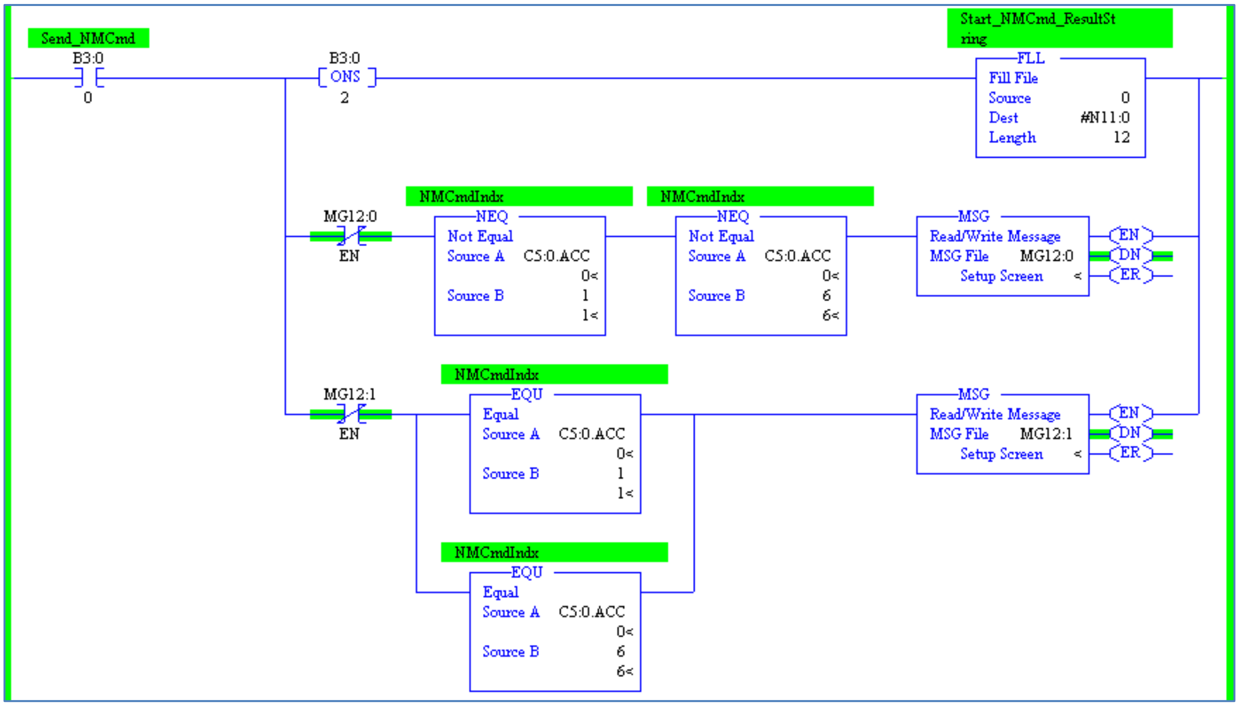
Details: The B3:0/11 (CopyNMCmdToCommonString) bit is still set so the ST30:9.LEN (native mode command string length) is copied (MOV) into the MSG control MG12:0 control file's .ELE element (this sets the numbers of bytes in the MSG's Size of Bytes (Send). This is done to make sure the correct data is sent if a short command is sent (like SO1), but you just copied and sent a string with length of 26 then the bytes after SO1 are not null. You could also just clear the N file you are copying the string into to make sure no extra characters follow the string you want to send. Another reason would be if you set static length of 26 in the MSG Send Bytes, but the command is longer for some reason. Commands (MSG) would fail in both cases.

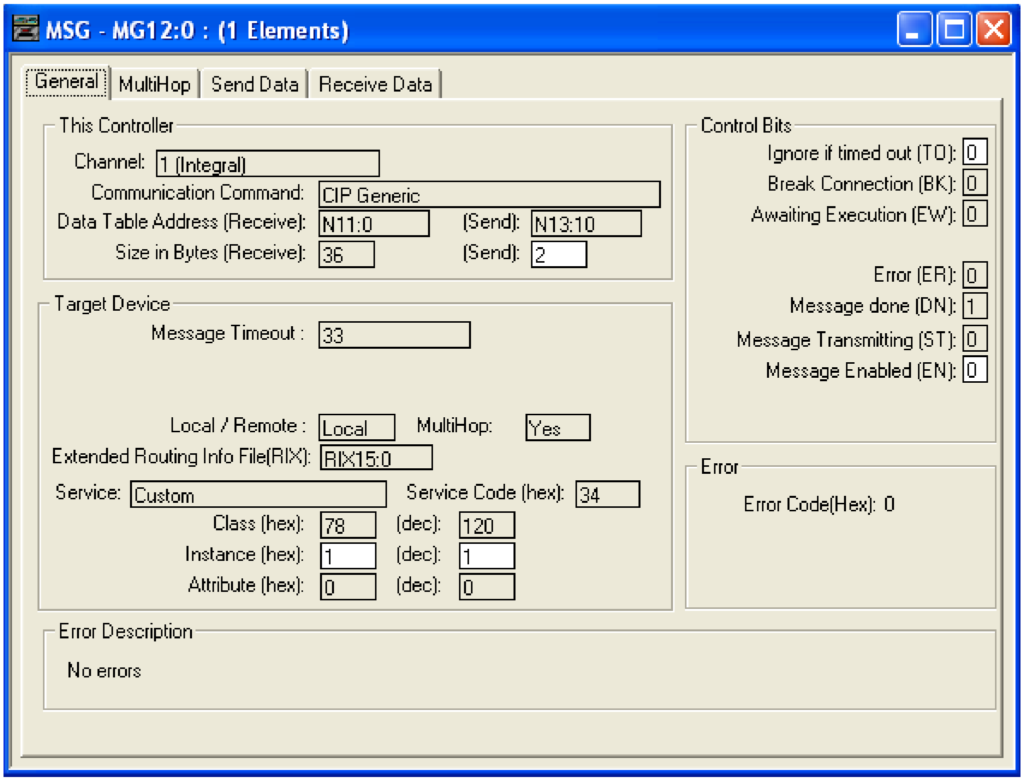
The native mode command string data is then copied (MOV) 2 bytes at a time into integer file N13 starting at word 10 (N13:10) as ASCII characters. This needs to be done because in a MicroLogix1100 you can only send and receive string data using the N file type. The command string length is checked and only the correct number of bytes of the string is copied into the N13 words. See Rockwell Knowledgebase #596235 – CIP Generic Message String Data regarding this and using the CPW instruction if you use a MicroLogix 1400.

Once the whole command string is copied all the words in N13 need to be byte-swapped using indexed addressing starting at N13:10. The Length field must be a constant so it needs to be as long or longer than half the length of native mode command string since it is 2 bytes (characters) per N13 word. So, if the longest string is 24 bytes, then the length needs to be 12 words or more.

Bit B3:0/0 is then latched for use in the next rung. It is latched because the MSG instruction runs over several scans and the logic that depends on the results would not be valid if it was not latched.

Rung 2 - Summary: Sends the native mode command using one of 2 MSG instructions and clears the result N11:0 file to store the response if one is returned.





This Controller

Channel: 1 (Integral)

Communication Command: CIP Generic

Data Table Address (Receive): N11:0 (Send): N13:10

Size in Bytes (Receive): 36 (Send): 2

Target Device

Message Timeout : 33

Local / Remote : Local MultiHop: Yes

Extended Routing Info File(RIX): RIX15:0

Service: Custom Service Code (hex): 34

Class (hex): 78 (dec): 120

Instance (hex): 1 (dec): 1

Attribute (hex): 0 (dec): 0

Control Bits

Ignore if timed out (TO): 0

Break Connection (BK): 0

Awaiting Execution (EW): 0

Error (ER): 0

Message done (DN): 1

Message Transmitting (ST): 0

Message Enabled (EN): 0

Error

Error Code(Hex): 0

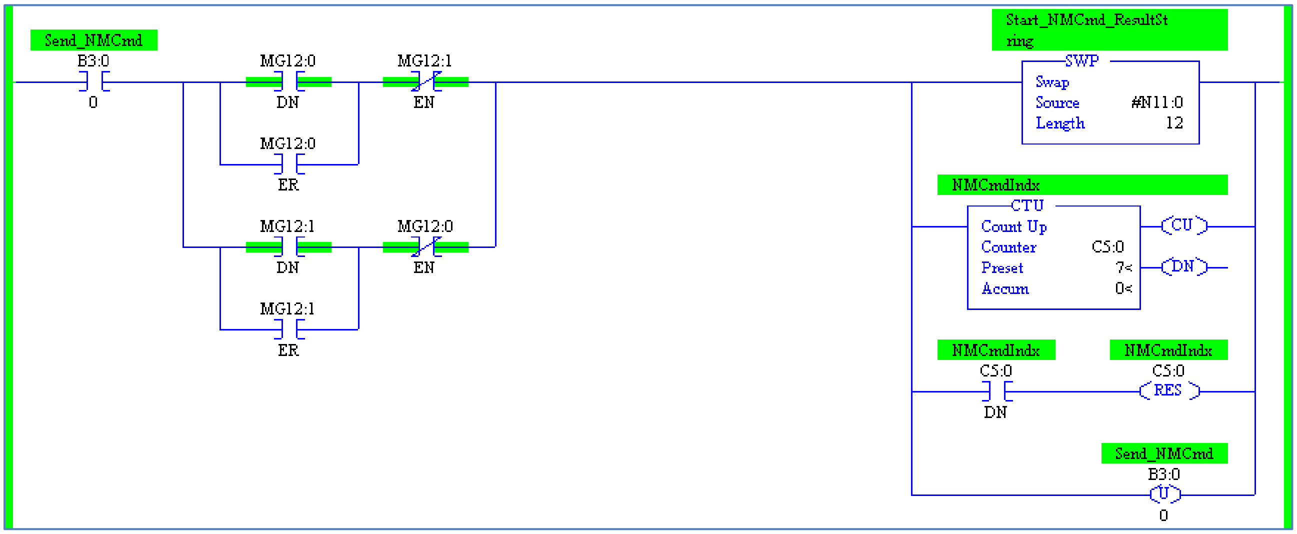
Error Description

No errors

Details: The destination data table file words are cleared (FLL used to fill 0's in the N11:0 – N11:11) so that any response data can be read easily from the native mode command sent.

There are 2 MSG's in this sample. The first one is used for sending the native mode commands that do not return a response or send a response with an even number of bytes. The 2nd MSG is for commands that return only 1 byte of response data. There is a bug in the MicroLogix firmware/OS, where if your MSG to the In-Sight system responds with an odd number of bytes and you put in a Size in Bytes (Receive) that is even or several bytes longer than that last odd byte will not be returned in the response.   You must set that to the exact number of odd bytes returned to get all the data. So, if you send the GF (Get File) command and it returns a job name that is 15 characters long, but your Size in Bytes (Receive) setting is 16 or more bytes, the job name will be returned with the “b” missing in jobname.job. The same thing occurs with the GO (Get Online) command which returns 1 byte (character). If I send the MSG set to 2 or more bytes to be received, you will not receive any bytes in the response so that is why there are 2 MSG’s. The 2 MSG’s are conditioned to run depending on the index used for the native mode command.

<u>Rung 3 - Summary:</u> Once the MSG is done, meaning the command was sent, the result string (if returned) is stored in file N11 and the bytes are swapped for easier reading. The index for changing native mode commands is also incremented until it reaches 7 and it is then reset.



<u>Details:</u> When either of the MSG’s (MG12:0 or MG12:1) complete and the index equals 3 (command sent was GF) the bytes in the receive data table address N11 are swapped in each word so the job name string is in the correct byte order and easily readable.

The index is then incremented (range 0 – 6 using counter C5:0) and will automatically reset to 0 after all the commands have been sent.

Bit B3:0/0 is then unlatched completing one native mode command cycle.

If desired a rung similar to rung 1 could be written to copy the result string in the N11 words to a common string file (ST) element for easier viewing/use in the logic.

SLC 5/05:

Notice that the program is much more compact due to some nice capabilities available in the SLC 5/05 that do not exist in the MicroLogix1100. The first item is that I can use indirect addressing to copy a full string from the ST data file based on the index into the ST common element location (ST30:9) and then into the N integer send file location. Note that the string length is also copied into the N integer file in the first word before the string data.

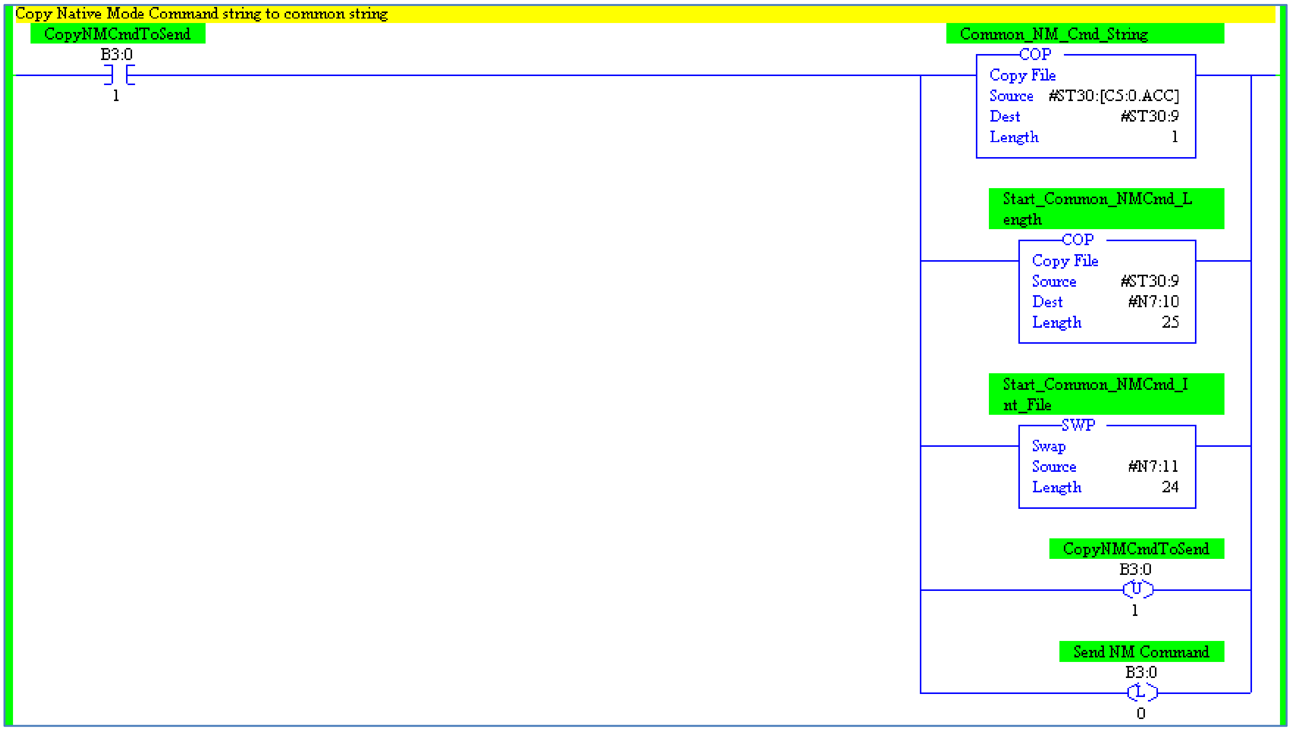
The SLC5/05 does not use a MSG instruction set to CIP Generic for the Communication Command. Instead it uses an EEM (EtherNet/IP Explicit Message) which uses an N integer (instead of MG and RIX) file for the control block and routing information. The control block length is 58 words.

If a string is returned as a response it is also very easy to copy the N integer file words back into a string file element.

The EEM instruction does not have the issue with an odd number of bytes returned in the response either.

Rung 0 - Summary:

This rung copies the native mode command strings stored in ST30:0 – ST30:6 based on index C5:0 to common location ST30:9. Each time the user sets bit B3:0/1 (CopyNMCmdToSend) and the index was incremented the next command is copied to the common string storage location. The currently selected native mode command is copied into an integer file (N7) and the bytes are then swapped in the integer file.



Copy Native Mode Command string to common string

CopyNMCmdToSend
B3:0
1

Common_NM_Cmd_String
COP
Copy File
Source #ST30:[C5:0.ACC]
Dest #ST30:9
Length 1

Start_Common_NMCmd_Length
COP
Copy File
Source #ST30:9
Dest #N7:10
Length 25

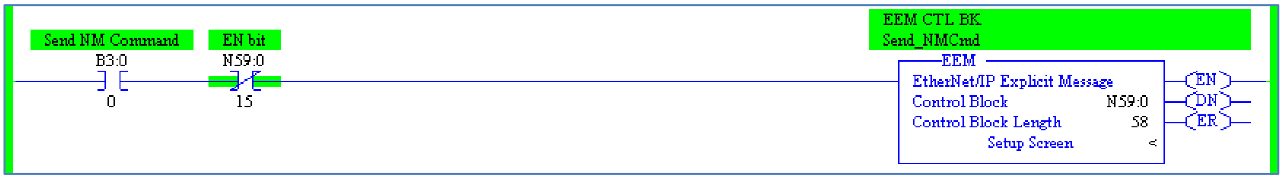
Start_Common_NMCmd_Int_File
SWP
Swap
Source #N7:11
Length 24

CopyNMCmdToSend
B3:0
U
1

Send NM Command
B3:0
L
0

Details: Set bit B3:0/1 to start sending the first native mode command. The first command is copied to the common string location in ST30:9. Since the string cannot be sent in the EEM, it needs to be copied into an N file and is copied into N7 starting at word 10 (N7:10). Note that since the string is copied using indirect indexed addressing, the whole string structure is copied into N7 (both LEN and DATA[]), so N7:10 contains the 16 bit string length and N7:11 is the start of the ASCII characters. The string structure is copied into 25 words in the N7 file. Then starting at N7:11 24 words are swapped in each word of the N7 file (48 bytes).
Bit B3:0/1 is then unlatched and bit B3:0/0 is then latched for the following rungs.

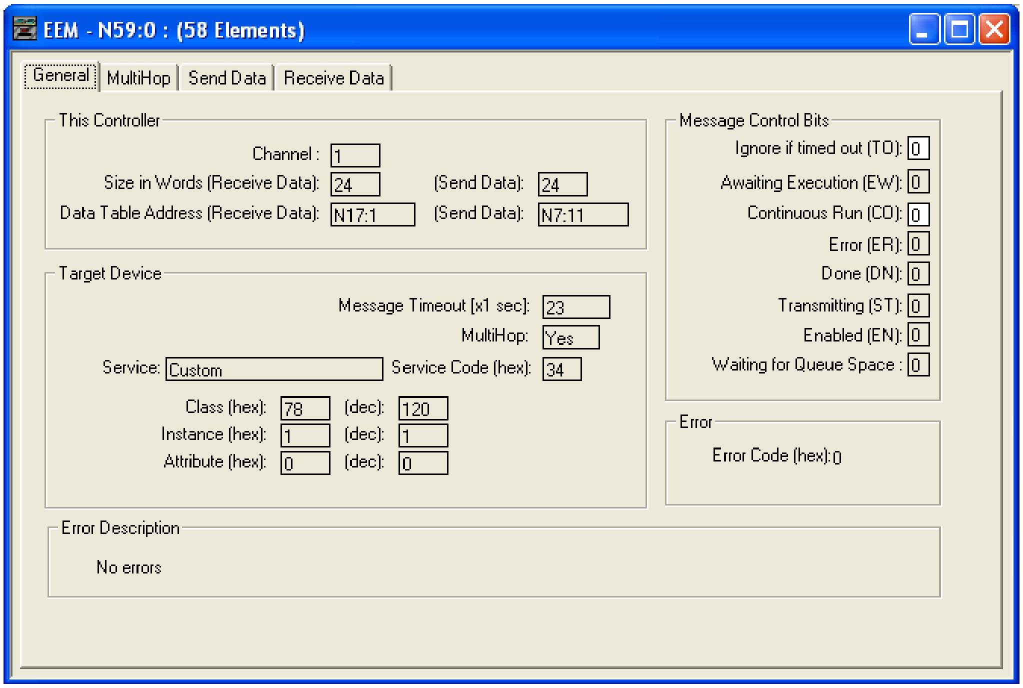
Rung1 - Summary: This rung sends an EEM (EtherNet/IP Explicit Message) containing the native mode command and stores a response string if one is returned.



Send NM Command
B3:0
0

EN bit
N59:0
15

EEM CTL BK
Send_NMCmd
EEM
EtherNet/IP Explicit Message
Control Block N59:0
Control Block Length 58
Setup Screen <



EEM - N59:0 : (58 Elements)

This Controller

Channel : 1

Size in Words (Receive Data): 24 (Send Data): 24

Data Table Address (Receive Data): N17:1 (Send Data): N7:11

Target Device

Message Timeout [x1 sec]: 23

MultiHop: Yes

Service: Custom Service Code (hex): 34

Class (hex): 78 (dec): 120

Instance (hex): 1 (dec): 1

Attribute (hex): 0 (dec): 0

Message Control Bits

Ignore if timed out (TO): 0

Awaiting Execution (EW): 0

Continuous Run (CO): 0

Error (ER): 0

Done (DN): 0

Transmitting (ST): 0

Enabled (EN): 0

Waiting for Queue Space : 0

Error

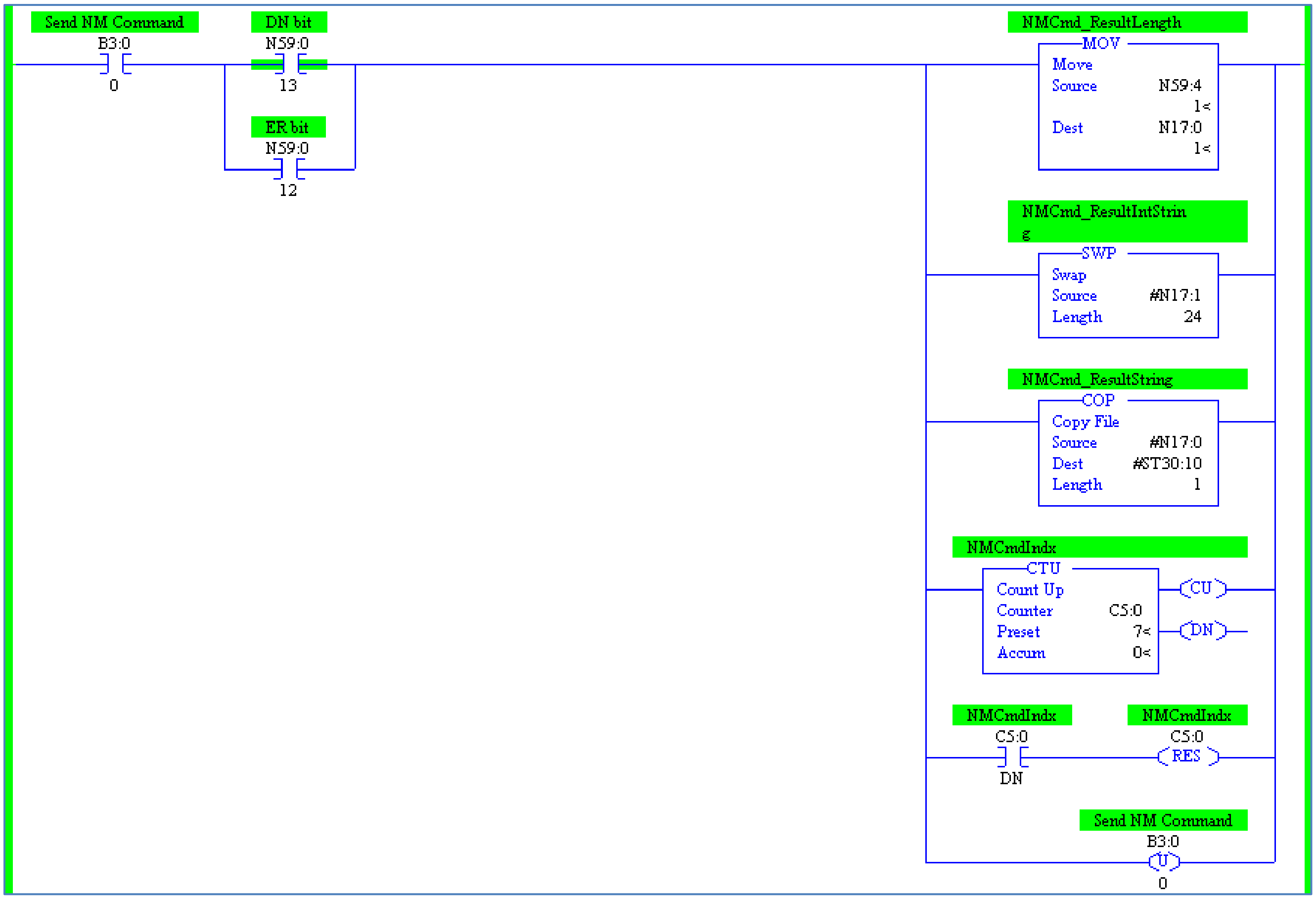
Error Code (hex):0

Error Description

No errors

<u>Details:</u> When bit B3:0/0 is set the EEM instruction is executed and 24 words (48 bytes) of ASCII data is written (the native mode command). There is not a way to set the number of words to send in the N file control block, unlike with the MSG using the MG control block used in MicroLogix 1100, so 48 bytes is used with remaining words set to 0 after the actual native mode command.
If any data is received after the command executes, up to 24 words (48 bytes) are stored in integer file N17 starting at word 1 (N17:1). The receive size is set to 24 words/48 bytes and any data beyond the actual response data will be nulls/0's so if no data is received N17:1 – N17:24 will be nulls/0's.

<u>Rung 2 - Summary:</u> When the EEM is done the native mode command result string and length are copied from the integer file N17 to result string element ST30:10. The index for changing native mode commands is also incremented until it reaches 7 and it is then reset.



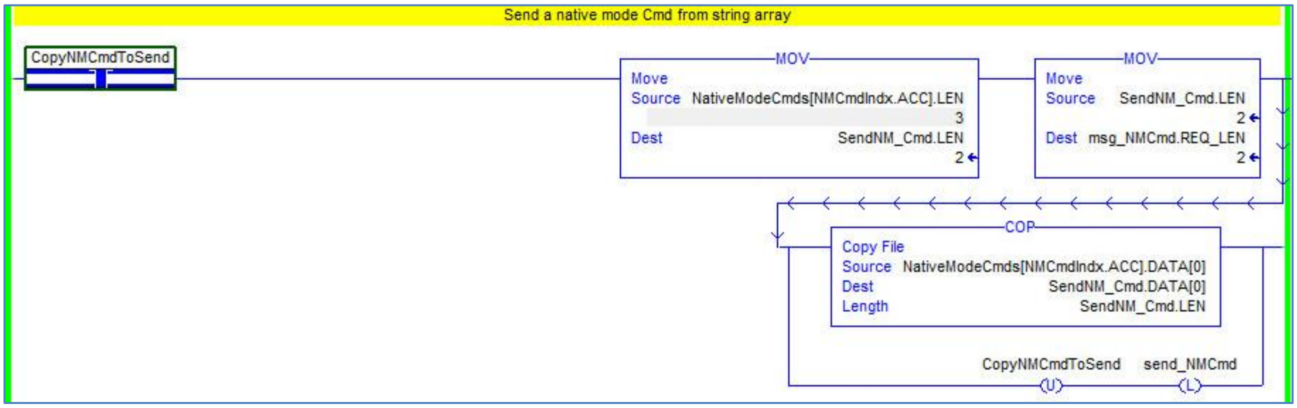
<u>Details:</u> Once the EEM instruction is done and the DN bit is set, the received length in bytes (control block N59, word 4 – N59:4) is moved into N17:0. Next the response bytes are swapped. We know in this sample that only index 3 (GF – Get File) returns a string longer than 1 byte so the bytes need to be swapped in the N17 file starting at word 1 (N17:1). 24 words in the N17 file have the bytes swapped in them for a total of 48 bytes. Note that the SWP instruction only allows a constant in the Length parameter so 24 was used since it is more than long enough for any strings returned in this sample.

Then 25 words in the N17 file are copied into a string (ST30:10) via one COP instruction using indexed addressing which is nice compared to using multiple copies 2 bytes at a time. Note that the string length was copied into N17:0 and the ASCII data into N17:1 and the following 23 words. After that the counter index is incremented until the last index (6) is reached and then the counter is reset to 0. Bit B3:0/0 is then unlatched.

ControlLogix/CompactLogix:

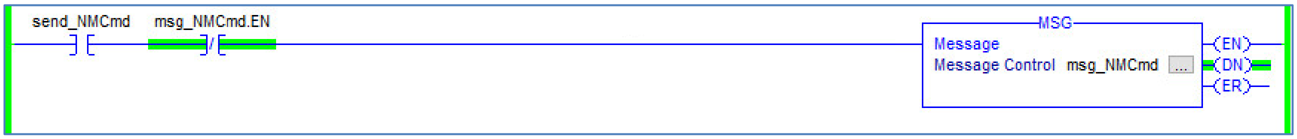
The Logix platform is the easiest and most flexible platform to program the use of native mode commands on. You create and use tag names with user assigned data types as needed. You can have multiple output instructions on the same branch instead of branching for each output instruction. You can use a tag name for the Length parameter in COP instructions, in the previous samples you had to use constants. The MSG instruction allows you to set the request length and get the receive length from the control block so you have more control over the string data sent and returned and not seeing nulls/0's after the string data.

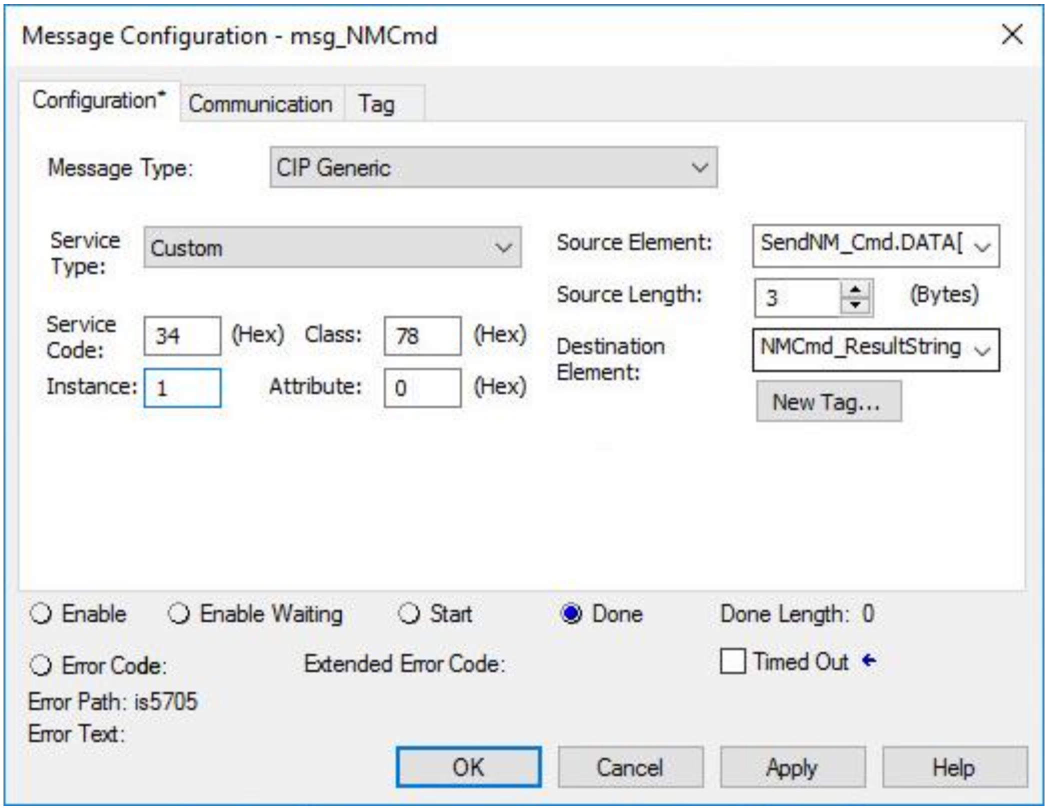
Rung 0 - Summary: This rung copies a native mode command and length from the string array to the string tag SendNM_Cmd.



Details: Set the CopyNMCmdToSend bit. This will copy the native mode command length to both the common string location (SendNM_Cmd) and the request length of the MSG control block (msg_NMCmd). Then using the counter (NMCmdIndx.ACC) as an index into the string array (NativeModeCmds) holding the commands, the string ASCII characters are copied into the common string location. This common string tag data is used in the MSG source element. The CopyNMCmdToSend bit is then unlatched and the send_NMCmd bit is latched for use in the following rungs.

Rung 1 - Summary: The native mode command is sent using the MSG instruction and stores a response string if one is returned.





Message Configuration - msg_NMCmd

Message Type: CIP Generic

Service Type: Custom

Service Code: 34 (Hex) Class: 78 (Hex)

Instance: 1 Attribute: 0 (Hex)

Source Element: SendNM_Cmd.DATA[

Source Length: 3 (Bytes)

Destination Element: NMCmd_ResultString

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

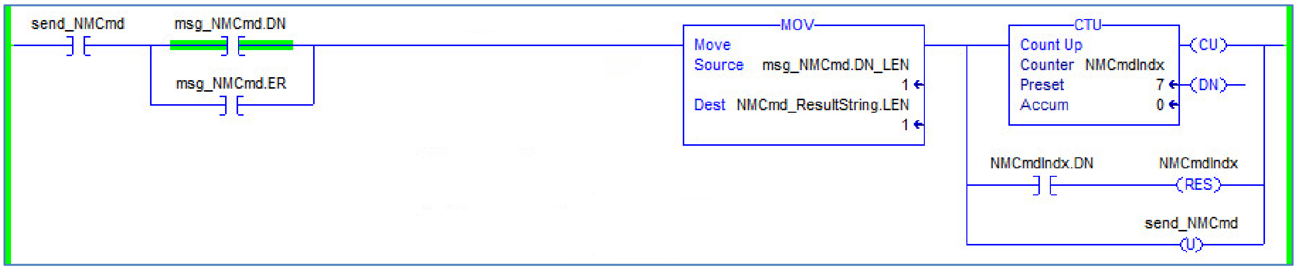
○ Error Code: Extended Error Code: □ Timed Out

Error Path: is5705

Error Text:

Details: The MSG instruction is enabled after the send_NMCmd bit is set in rung 0. Remember the source length field is set is above in rung 0. If the native mode command returns a response string the data is stored in string tag NM_Cmd_ResultString.

Rung 2 - Summary: When the MSG is done the native mode command result string length is set per the received length in the MSG. The index for changing native mode commands is also incremented until it reaches 7 and it is then reset.



Details: When the MSG completes (DN bit set) the length of the data received (if any) is copied into NMCmd_ResultString.LEN. Doing this makes it easier to read the result string without a bunch of “$00” at the end of the string after the data read if the length is left at 82 bytes. The NMCmdIndx count is incremented by 1. Once 7 commands are sent the NMCmdIndx count index is reset to 0. Then the send_NMCmd bit is unlatched to complete the cycle.

Installing the Library files

Steps to Import Rungs into an RSLogix500 project for MicroLogix 1100 or SLC5/05:

1. Open a new or existing project *.RSS project for the PLC platform being used.
2. Configure Ethernet settings using Channel 1 in the Channel Configuration dialog if not already configured.
3. While offline and in the ladder editor, right mouse click on a rung and select “Paste from SLC Library”.
4. Browse to the directory where you copied the sample files and select the NativeModeCommands-5_05.SLC for a SLC5/05 or NativeModeCommands-MLgx1100.SLC for a MicroLogix 1100.
5. Click the Open button.
6. In the Import SLC Format dialog that displays select the appropriate options for importing the rungs, comments, descriptions, symbols and data table addresses. If this is a new project the default settings are fine. If you are importing into an existing project, click the Help button to understand what the different settings will do.
7. Click OK once the settings are suitable.
8. The 3 rungs (SLC5/05) or 4 rungs (MicroLogix 1100) will be added to the ladder editor with all e’s to the left of them meaning they are in edit mode.
9. Edit the EEM instruction in rung 1 (SLC5/05) or the 2 MSG’s in rung 2 (MicroLogix 1100) and in the MultiHop tab enter the IP address of the In-Sight vision system running EtherNet/IP.
10. Click the menu item Edit -> Verify File. All the rungs should have the e’s removed and there should not be any errors. If there are errors, address them according to what is displayed in the Verify Results tab in the Results pane at the bottom of the RSLogix 500 window.
11. Download the project and set the PLC to Run mode.
12. Read the rung descriptions below for functional descriptions and bits to toggle on/off.

Steps to Import Rungs into an RSLogix5000/Studio5000 project for ControlLogix or CompactLogix:

1. Open a new or existing project *.ACD project for the PLC platform being used.

2. Add the appropriate In-Sight vision system under the Ethernet node in the Controller Organizer if it does not already exist. This is not necessary especially if you are not using an implicit connection, but it makes it easier to define the path in the MSG instruction. You can inhibit the connection if necessary.
3. While offline and in the ladder editor of a routine right mouse click on a rung and select “Import Rungs…”.
4. Browse to the directory where you copied the sample files and select the NativeModeCommands-CLgx.L5X.
5. Click the Open button.
6. In the Import Configuration dialog that displays check the settings and then click the OK button since the tags used in the import rungs most likely do not exist in the project.
7. The 3 rungs will be added to the ladder editor at the position specified in the import dialog.
8. Edit the MSG instruction in rung 1 and in the Communication tab click the Browse button by the Path textbox and select the In-Sight vision system running EtherNet/IP from the tree. The name is then filled in for the Path.
9. Click OK and then OK on the Message Configuration dialog.
10. Download the project and set the PLC to Run mode.
11. Read the rung descriptions below for functional descriptions and bits to toggle on/off.