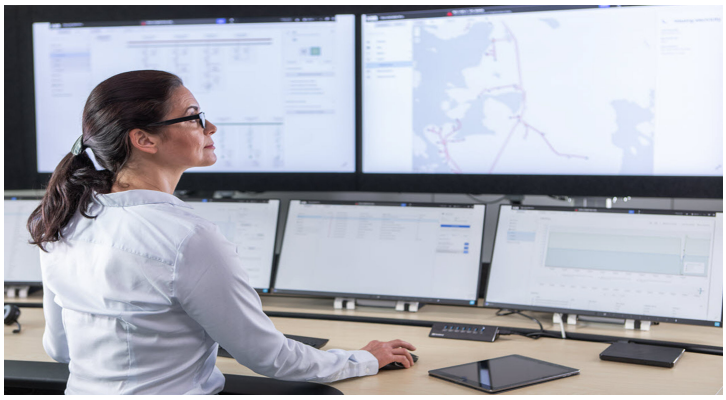
GRID AUTOMATION PRODUCTS

MicroSCADA X SYS600 10.2

DNP 3.0 Slave Protocol



Document ID: 1MRK 511 487-UEN
Issued: March 2021
Revision: A
Product version: 10.2

Table of contents

Section 1 Copyrights

The information in this document is subject to change without notice and should not be construed as a commitment by Hitachi Power Grids. Hitachi Power Grids assumes no responsibility for any errors that may appear in this document.

In no event shall Hitachi Power Grids be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall Hitachi Power Grids be liable for incidental or consequential damages arising from the use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from Hitachi Power Grids, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Trademarks

ABB is a registered trademark of ABB Asea Brown Boveri Ltd. Manufactured by/for a Hitachi Power Grids company. All other brand or product names mentioned in this document may be trademarks or registered trademarks of their respective holders.

Guarantee

Please inquire about the terms of guarantee from your nearest Hitachi Power Grids representative.

Third Party Copyright Notices

List of Third Party Copyright notices are documented in "3rd party licenses.txt" and other locations mentioned in the file in SYS600 and DMS600 installation packages.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (https://www.openssl.org/). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Section 2 Introduction

2.1 This manual

This manual provides thorough information on the use of DNP 3.0 Slave Protocol (Distributed Network Protocol) and information related to it. It describes how to configure the base system and the communication system to establish communication to DNP 3.0 master. DNP 3.0 protocol is standardized as IEEE standard 1815 - IEEE Standard for Electric Power Systems Communications -- Distributed Network Protocol (DNP3). Term DNP 3.0 is used in this manual and in the referenced tools.

In addition to this configuration, the base system needs to be configured for the process communication. For more information on this subject, see other manuals, for example SYS600 Application Objects or SYS600 COM500*i* User's Guide.

The DNP 3.0 master in the Network Control Center needs to be configured together with the DNP3.0 slave in COM500*i*.

2.1.1 DNP 3.0 slave

The DNP 3.0 slave protocol is mainly used for upper level communication between SYS600 (COM500i) and an NCC, as illustrated in Section 2.1.1:

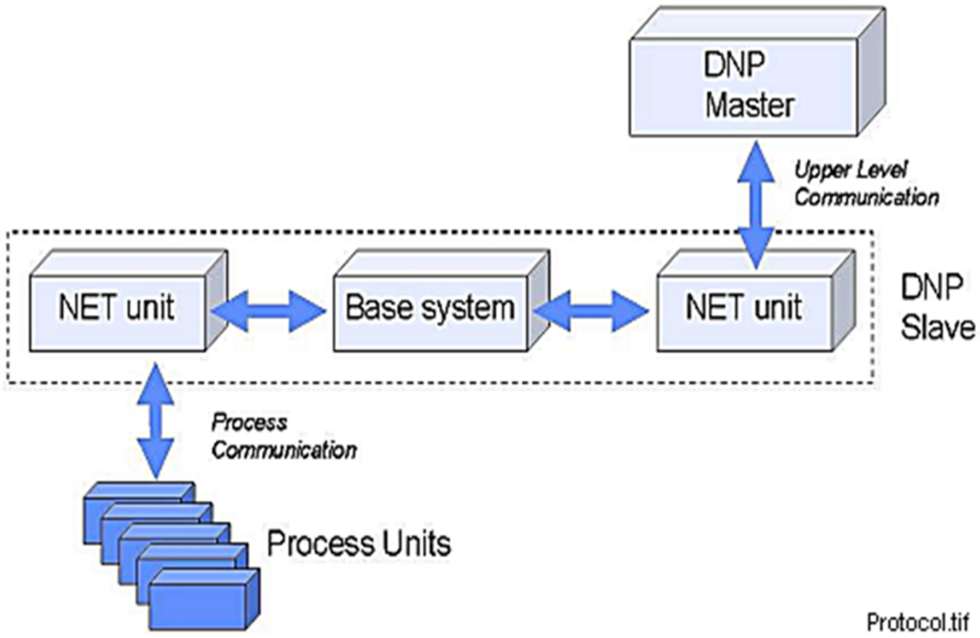


Figure 1: DNP Master sees the NET unit and the process behind it as a slave

The data from the process activates certain event channels and command procedures in the base system. This command procedure forwards the information to the NET unit and the DNP 3.0 master.

DNP 3.0 Slave Protocol (Distributed Network Protocol) can be configured to operate as TCP server and is thus exposed to attacks from the network. For more information, see SYS600 Cyber Security Deployment Guideline, which collects instructions to harden the system as a whole.

2.2 Use of symbols

This publication includes warning, caution and information symbols where appropriate to point out safety-related or other important information. It also includes tips to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

Warning icon indicates the presence of a hazard which could result in personal injury.

Caution icon indicates important information or a warning related to the concept discussed in the text. It might indicate the presence of a hazard, which could result in corruption of software or damage to equipment/property.

Information icon alerts the reader to relevant factors and conditions.

Tip icon indicates advice on, for example, how to design a project or how to use a certain function.

Although warning hazards are related to personal injury, and caution hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all warnings and caution notices.

2.3 Document conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a dialog, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for file names.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the CTRL key. Although the Enter and Shift keys are not labeled, they are written in capital letters, for example, press ENTER.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the space bar, comma key and so on.
- Press CTRL+C indicates that the user must hold down the CTRL key while pressing the C key (in this case, to copy a selected object).
- Press ALT E C indicates that the user presses and releases each key in sequence (in this case, to copy a selected object).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
    - The following convention is used for menu operations: **Menu Name/Menu Item/ Cascaded Menu Item**. For example: select **File/Open/New Project**.
    - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.
- System prompts/messages and user responses/input are shown in the Courier font. For example, if the user enters a value that is out of range, the following message is displayed: `Entered value is not valid`.
  The user may be told to enter the string MIF349 in a field. The string is shown as follows in the procedure: `MIF349`
- Variables are shown using lowercase letters: sequence name

2.4 Related documents

The following SYS600 manuals should be available for reference during the use of this manual:

Name of the manual	Document ID
SYS600 10.2 DNP 3.0 Master Protocol	1MRK 511 486-UEN
SYS600 10.2 Communication Gateway, COM500*i*	1MRK 511 468-UEN
SYS600 10.2 System Configuration	1MRK 511 481-UEN
SYS600 10.2 System Objects	1MRK 511 482-UEN
SYS600 10.2 Application Objects	1MRK 511 467-UEN
SYS600 10.2 Cyber Security Deployment Guideline	1MRK 511 485-UEN

The following documents of the DNP 3.0 protocol are available via the DNP Users Group:

- DNP 3.0 DATA LINK LAYER
- DNP 3.0 APPLICATION LAYER
- DNP 3.0 DATA OBJECT LIBRARY
- DNP 3.0 TRANSPORT FUNCTIONS
- DNP 3.0 SUBSET DEFINITIONS

The same information is presented in IEEE Standard 1815.

Other related documents:

- Product documentation of the used modem

2.5 Document revisions

Revision	Version number	Date	History
A	10.2	31.03.2021	New document for SYS600 10.2

Section 3 Safety information

This section has information on the prevention of hazards and taking backups from the system.

3.1 Backup copies

3.1.1 Taking backup copies

We recommend taking backup copies before making any changes, especially ones that might have side effects. Software and data need to be copied to another place.

Backup copying makes it easier to restore the application software in case of disk crash or other severe failure where stored data is lost. It is therefore recommended that backup copies are taken regularly.

There should be at least two system backup copies and two application copies. A new backup is copied over the oldest backup. This way the latest version is always available, even if the backup procedure fails.

Detailed information on how to take backup copies should be delivered to the customer with the application.

3.1.2 System backup

Usually a system back up is taken after the application is made. It should be taken again when changes are made to the SYS600 system. This is required when the driver configuration or the network setup is changed.

3.1.3 Application backup

An application backup is also taken at the same time with the system backup, after the application is made. It should be taken again when changes are made to the application, for example, if pictures or databases are edited or new pictures are added.

3.2 Fatal errors

A fatal error is an error that causes a breakdown or a locked situation in the SYS600 program execution.

3.2.1 Handling

In case of a fatal error:

1. Write down the possible SYS600 error messages.
2. Shut down the SYS600 main program. If this cannot be done in the SYS600 Control Panel, try to end the task in Windows Task Manager.

Files may be damaged if the base system computers are shut down by switching the power off.

3. The data kept in the main memory at the moment of a fatal error is placed in the drwtsn32.log file with Windows 2003 Server, Windows XP and earlier. By default, it is placed under %SYSTEMDRIVE%\Documents And Settings\All Users\Application Data \Microsoft\Dr Watson. Log and dump file paths can be checked with the drwtsn32 application. (Start -> run -> drwtsn32.exe). Analyze and copy the data in these files. Starting with Windows Server 2008 and Windows 7 the crash handling has changed. The location of the dump files can be read from the registry under the key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting \LocalDumps. The DumpFolder value tells the location of the dump files. Collect the data from this location.
4. Restart the system.

Report the program break-down together with the possible SYS600 error messages and the information from the drwtsn32.log file to the SYS600 supplier.

3.2.2 Status codes

Error messages in SCIL are called status codes. A list of status codes and short explanations for them can be found in SYS600 Status Codes.

Section 4 Instructions

4.1 Communication

In SYS600 the DNP 3.0 Slave protocol is implemented only in the PC-NET software. PC-NET unit communicates over an INTEGRATED link and via the serial or LAN ports of the base system computer.

Setting the attributes of SYS600 system objects can modify the communication parameters.

The base system considers each DNP 3.0 Slave device as a station that has been created to a line of a NET unit. Each DNP 3.0 station works as a protocol converter that converts data between the internal protocol of SYS600 and the DNP 3.0 protocol.

The protocol can be used with virtual serial ports by using a special setting of the line attributes. See NET line attribute CM for more information.

4.2 Installation

The SYS600 installation is required.

4.3 Configuration

Configuration can be made either by using the System Configuration Tool or by using SCIL statements. For more information on the System Configuration Tool, see SYS600 System Configuration manual, chapter "PC-NET start-up with System Configuration Tool" and "System Configuration Tool". The usage of the System Configuration Tool is recommended, but if there is a need to create the communication configuration using SCIL, it is instructed in the following chapters. In this case, the configuration can be divided into two parts:

The configuration can be divided into two parts:

- Base system configuration
- Communication system configuration

The attribute descriptions presented in chapter 'Communication system configuration' are the same for configurations created with System Configuration Tool or with SCIL.

4.3.1 Base system configuration

It is assumed here that the base system configuration for objects other than the communication has been made according to the instructions in the System Configuration manual.

The extra steps needed to configure the communication are:

1. Define a node number for a PC_NET instance.
2. Reserve a link number for a PC_NET instance. Creating the link as instructed in step 6 starts the PC_NET instance.
3. Create the base system STA object for each remote IED (master function) or for each NCC connection (slave function).

- DNP 3.0 Slave protocol uses the station type DNP (STY type 30)

The STA objects are created to SYS_BASCON.COM using the template or with a separate creation sequence. If the template is not used, the sequence should contain the line:

```
#create STA'Sta_Nb':B = %Sta
```

where 'Sta_Nb' is the number of the station object in the base system. %Sta is a list object which should contain at least the following settings: TT = "EXTERNAL", ST = station type, ND = node number defined in step 1 and TN = translated object number (usually the same as 'Sta_Nb'. See SYS600 System Objects manual for more information on the base system object attributes for STA object).

4. Edit the PC_NET.CF1 according to the description in chapter "Start-up definition file PC_NET.CF1" in the SYS600 System Configuration manual
5. Create a command procedure which creates the lines and stations to the NET object (= pc_net instance) using the S-attributes.
   See Section 4.3.2 for more information on the attribute setting. A sample creation script is presented at the end of this manual.
6. Create a command procedure which creates the link of type 'INTEGRATED' to the base system. This procedure should contain the line:

   ```
   #set LIN'i_Integrated_Link_Number':BLT = "INTEGRATED"
   ```

   where 'i_Integrated_Link_Number' is the number of the link reserved in step 2. The PC_NET executable is defined with the SC attribute of the link and it must set before setting of the LT attribute.

The testing of the communication system can be done as follows:

1. Execute the procedure created in step 6. This starts the PC_NET instance and enable the setting of the S-attributes.
2. Execute the procedure created in step 5. If the lines and stations are set to IU = 1 (that is, they are in use) and the configuration is correct and complete in both ends, the communication starts.

For automatic start-up of the communication, the created command procedures must be attached to the APL_INIT_1:C procedure.

4.3.2 Communication system configuration

Each NET instance contains a set of system objects which specify the existence and the usage of the communication lines and the station objects connected to those lines. These objects can be created, modified, and deleted by SCIL, and setting the attributes defines the functionality of these objects.

Access to the attributes can be one of the following:

- **Read-only**: The attribute can only be read. There are still a few exceptions in which the values can be reset.
- **Write-only**: The attribute can only be written (set).
- **Read, conditional write**: The attribute can be both read and written, but the object must be set out of use (IU = 0) before writing.
- **No limitations**: The attribute can be both read and written without limitations.

The configuration of the communication system in SYS600 can be divided into two layers: line layer and station layer. Both of these layers have a specific functionality and a set of attributes of their own.

The purpose of the communication system configuration is to:

- Create all the system objects needed to establish communication between the master and the slave. Related attributes for creation are PO (Line) and DV (Station).
- Adjust the values of the system object attributes to match the physical communication channel and the properties of the remote partner/partners. The menu selection 'Configuration->Preview->PC_NET' in the System Configuration Tool may provide an example of the SCIL based configuration script of any setup.

4.3.2.1 Setting the attribute values

All the line and station attributes have sensible default values, but the value of each attribute must be checked against the requirements of the actual communication system.

The attribute values depend on:

- The physical communication media (for example leased telephone line, radio link, power line carrier), which affects the attributes of the line, such as the baud rate and parity.
- The network topology used (point-to-point, multi-drop), which affects the link type.
- The size (number of stations) of the system, which affects the timeout parameters; the slower the media and larger the system, the longer timeouts are needed.
- The remote system(s), which affects both the line and station attributes, and the message types used.

4.3.2.2 Network topologies

The implementation of the DNP 3.0 Slave protocol in SYS600 supports direct and serial bus topologies. The direct topology (point-to-point) can be a direct physical cable from point-to-point, a two-node radio, or modem network. The serial bus topology (multi-drop) is commonly made up of many modems with their outputs/inputs tied together or connected using a star-coupler.

The DNP 3.0 link layer supports the multiple-slave and peer-to-peer communication methods. In peer-to-peer communication, all the stations act as slave data links and collisions are possible as no station has a higher priority, and all of them can transmit spontaneously. SYS600 uses random delay of retransmission as a collision avoidance method. When DNP 3.0 over LAN/WAN is used (TCP/IP or UDP/IP), the same topologies are used. In version 9.3 FP2 and newer, it is also possible that multiple station objects are sharing the same internet address in the configuration. See Figure 2

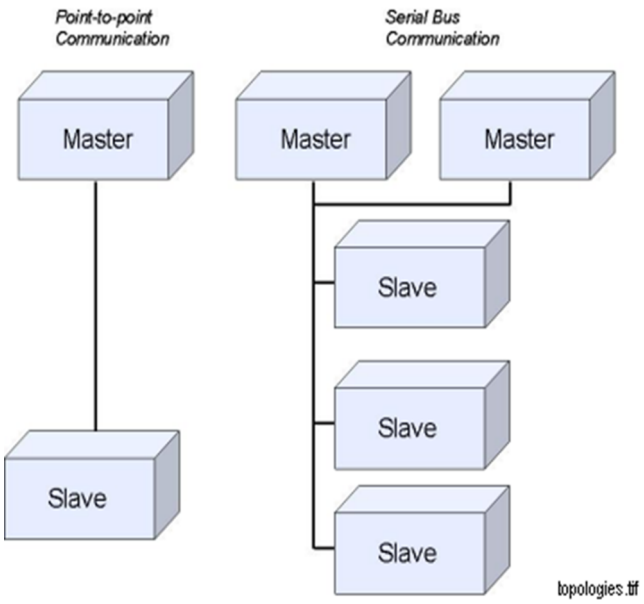


Figure 2: Network topologies

When making the DNP connection, an agreement about the used communication parameters should be made with the supplier or owner of the master system.

4.3.2.3 DNP 3.0 line layer

The line process of a NET unit performs the functions of the line layer. The purpose of the line layer is to send and receive messages to/from external devices using the DNP 3.0 protocol. By using the DNP 3.0 terminology, this means that the data link layer provides transfer of Link Service Data Units (LSDU) across the physical link. LSDUs are user data fragments small enough to fit into the FT3 frame format. The application layer of a NET unit is responsible for assembling and disassembling messages into LSDUs. The line layer provides frame synchronization and link control.

According to the DNP 3.0 documentation, the link layer performs the following functions:

- Exchange of LSDUs between peer DNP 3.0 data links
- Error notification to data link user
- Sequencing of LSDUs
- Prioritized LSDU delivery
- Quality LSDU delivery
- Performing message retries
- Frame Count Bit (FCB)
- Data Flow Control (DFC)
- Synchronizing and handling of the Frame Count Bit in the control word
- Setting and clearing the Data Flow Control bit based on buffer availability
- Packing user data into the defined frame format and transmitting the data to the physical layer
- Unpacking the frames that are received from the physical layer into user data
- Controlling all aspects of the physical layer
- Responding to all valid frames (function codes) received from the physical layer

Line layer attributes
The following attributes can be used for configuring DNP 3.0 master lines in SYS600.

IU	In Use

Indicates whether the line is in use (value 1) or not in use (value 0).

Data type:	Integer
Value:	0 or 1
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	No limitations

LD	Local Address

The IP address that is used locally. It is necessary to set this attribute when the computer has multiple IP addresses and it is defined which address the created line must use. This attribute must be set before the line has been taken into use for the first time. An empty string in LD means that the default IP address of the computer is used. The value of LD cannot be modified after the line has been taken into use for the first time. This attribute is used only in LAN mode of the protocol.

Data type:	Text
Value:	String containing a valid IP address, max 230 characters.
Default value:	Empty string
Index range:	1... 12 (NET line numbering)
Access:	Read, conditional write

This attribute accepts the IP address in form:

```
#SET NET1:SLD1="192.168.1.10"
```

The standard TCP port of 20000 specified for DNP 3.0 over LAN is used if the port number is not explicitly specified. If a non-standard port for the line is needed, the port number should be separated with a semicolon:

```
#SET NET1:SLD1=”192.168.1.1;20001” ;server port = 20001, (no spaces
allowed)
```

The defined port number must be between range 1..65535. The operating system and other applications running in the same computer may cause limitations to the availability of the port numbers. The same local address and port cannot be used by multiple slave lines. This means that if the lines are using the standard TCP port 20000, a unique IP address must defined for each DNP 3.0 slave line.

There is an internal limitation which prohibits the use of the same local IP address and the same line number multiple times. This applies to all PC_NET protocols using LAN.

Example:

It is not possible to have

IEC104 master in Line 1 in PC_NET 1 with LD="192.168.1.1"

and

DNP3.0 slave in Line 1 in PC_NET 2 with LD="192.168.1.1"

the configuration must be changed to

IEC104 master in Line 2 in PC_NET 1 with LD="192.168.1.1"

and

DNP3.0 slave in Line 1 in PC_NET 2 with LD="192.168.1.1"

or to

IEC104 master in Line 1 in PC_NET 1 with LD="192.168.1.1"

and

DNP3.0 slave in Line 1 in PC_NET 2 with LD="192.168.1.2"

The limitation is present only when the same local IP address is used. An easy workaround is to configure multiple IP-addresses which are using the same adapter. If this is not possible, setting a unique value for the NET node attribute LP redefines the internally used ports for the NET node and no conflict takes place. See SYS600 System Objects manual for more information about the NET Node attribute LP.

PO	**Protocol**

The data transfer protocol used on the line. The line is defined to the NET by setting this attribute. By setting the attribute to 0 the line definition including all the line attributes are deleted.

Data type:	Integer
Value:	0...45 Value with DNP 3.0 Slave protocol : 35
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write

SD	**System Device Name**

Associates the NET line numbers of PC-NET with the device names of the physical channels of serial ports. By default, the line number 1 is connected to the COM1, the line 2 to the COM2 and so on. By using the SD attribute, it is possible to override these default values. This may be necessary if the COM ports are used as NET lines or if, for example, a RocketPort card is used.

Data type:	Text
Value:	See above
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write

When using DNP 3.0 over LAN, the SD attribute defines the used connection type. Possible types are TCP and UDP.

```
#SET NET'NET':SSD'LINE' = "TCP" ;line uses TCP connection

#SET NET'NET':SSD'LINE' = "UDP" ;line uses UDP connection
```

PS	**Buffer Pool Size**

Specifies the number of message buffers reserved for the line. Fixed buffer poll sizes are used in versions 9.3 FP1 and newer and this attribute is retained because of the backward compatibility. Setting the value for PS is not possible anymore. See the attributes PS, NB and PB from the System Objects manual for more information.

Data type:	Integer
Value:	1...250
Index range:	1...12 (NET line numbering)
Access:	Read (conditional write accepted but has no effect)

BR	Baud Rate

Transmission rate used on the line. The attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	1...19200 (384 = 38400 bauds, 576 = 57600 bauds)
Unit:	Bits / s
Index range:	1...12 (NET line numbering)
Default value:	9600
Access:	Read, conditional write

PY	Parity

Specifies the parity check (if any) used for the characters transferred on the line. The attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	0 = no parity check
	1 = odd parity
	2 = even parity
Index range:	1...12 (NET line numbering)
Default value:	2
Access:	Read, conditional write

RD	Receiver Data Bit Count

Specifies the number of data bits in each received character. The attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	5, 6, 7 or 8
Unit:	Data bits
Index range:	1...12 (NET line numbering)
Default value:	8
Access:	Read, conditional write

SB	Stop Bits

Specifies the number of stop bits attached to each transmitted character. The attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	1 or 2
Unit:	Stop bits
Index range:	1...12 (NET line numbering)
Default value:	1
Access:	Read, conditional write

TD	Transmitter Data Bit Count

Specifies the number of data bits in each transmitted character. The attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	5, 6, 7 or 8
Unit:	Data bits

Table continues on next page

TD	Transmitter Data Bit Count
Index range:	1...12 (NET line numbering)
Default value:	8
Access:	Read, conditional write

PD	Polling Delay

Delay (in milliseconds) between "test function of link" commands. The purpose of this command is to ensure that the communication to the slave/master is open. If this attribute is set to zero, the "test function of link" command is not sent. The link testing can also be controlled with the line attribute OM, bit 2.

Data type:	Integer
Value:	0... 65535
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	5000
Access:	Read, conditional write

ML	Maximum Message Length

Maximum length of an outgoing data link fragment (LPDU). This length is the amount of user data without checksums.

Data type:	Integer
Value:	50...249
Unit:	Octets
Index range:	1...12 (NET line numbering)
Default value:	230
Access:	No limitations

XR	Maximum Random Delay for Retransmission

Random transmission delay is a simple collision avoidance method used in DNP 3.0 lines. When unsolicited responses are enabled for slave stations on a multi-drop line, there is a possibility that several slave devices send messages at the same time. This message collision is seen as a timeout in a slave station since the master is not responding. The XR attribute limits the possibility that two devices retransmit messages at the same time.
This attribute can also be used for setting the priorities of the slave stations: the station with the smallest retransmission delay has the highest priority. The attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	0...65535
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	No limitations

TW	Transmission Wait Delay

Specifies the transmission delay in milliseconds. The transmission delay is the time that the NET must wait after receiving a CTS (Clear to Send) signal until starting the transmission of a message.

Data type:	Integer
Value:	0...65535
Unit:	Milliseconds

Table continues on next page

TW	Transmission Wait Delay
Index range:	1...12 (NET line numbering)
Default value:	5
Access:	No limitations

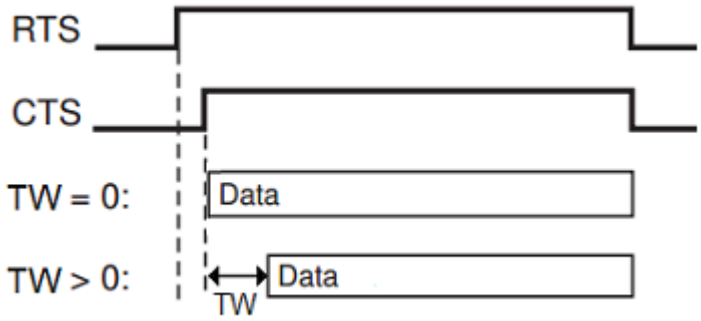


Figure 3: TW attribute

DE	CTS Delay

The maximum waiting time of the rising edge of the CTS signal after the activation of the RTS signal. If CTS is low after the timeout, it causes a CTS error and the transmission does not start. A line disconnection often leads to a CTS error. If value = 0 is given, 500 msec waiting time is applied. If simulated CTS high is used (serial only, line attribute CM, bit 1), there is no rising edge and transmission starts after the waiting time. The attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	0...65535
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	50
Access:	Read, conditional write

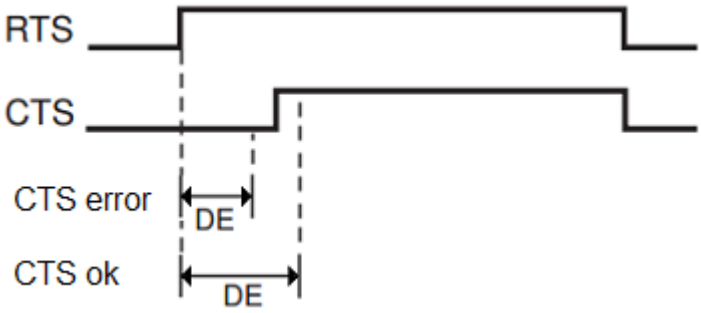


Figure 4: DE attribute

In the Figure 4 the smaller DE value causes a CTS error. A suitable DE value varies with each system.

HT	Header Timeout

Specifies the maximum waiting time in milliseconds within which the first byte of a link layer response should have been received after the transmission of a message. If no response has been received within this time, new attempts are made the number of times specified by the Enquiry Limit (EN) attribute. If a response is still not obtained, the station is suspended.

Data type:	Integer
Value:	0...65535
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	2000
Access:	Read, conditional write

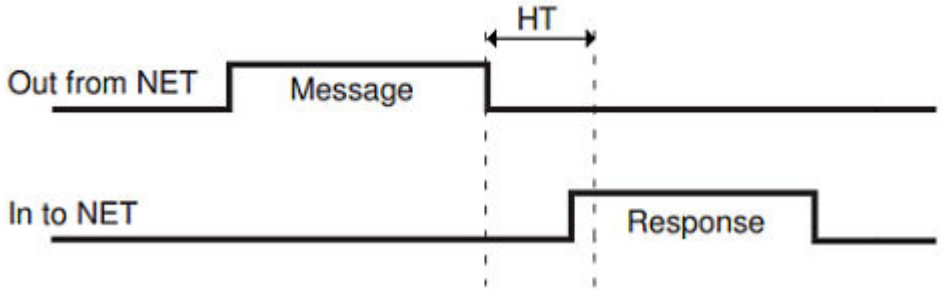


Figure 5: HT attribute

TI	Response Timeout

The time in seconds that the DNP link waits for the end of the line layer response.

Data type:	Integer
Value:	0...255
Unit:	Seconds
Index range:	1...12 (NET line numbering)
Default value:	2
Access:	No limitations

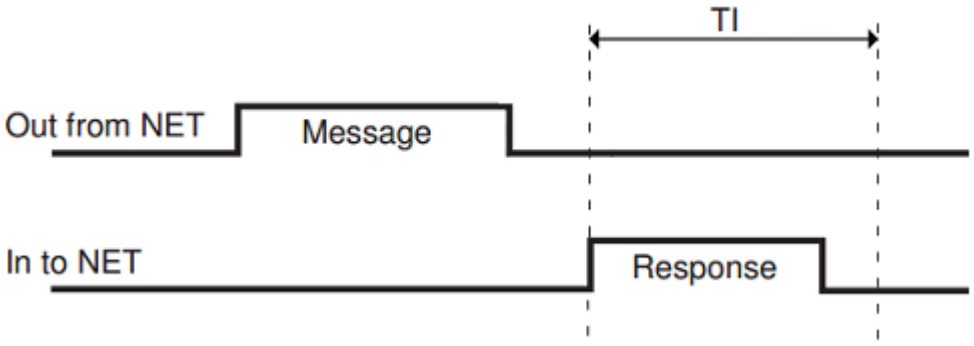


Figure 6: TI attribute

A slow communication speed and a long response might cause error with too small TI attribute value.

RY	RTS Keepup Delay

This attribute defines how long time the RTS-pin of the RS232-port is kept in the signal state after the serial driver completes the write operation. The write operation here means a transmission of any message. See also line attribute CM (Com Port Mode), bit 3. The attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	0...20
Unit:	Bytes (absolute time depends on baud rate)
Index range:	1...12 (NET line numbering)
Default value:	1
Access:	Read, conditional write

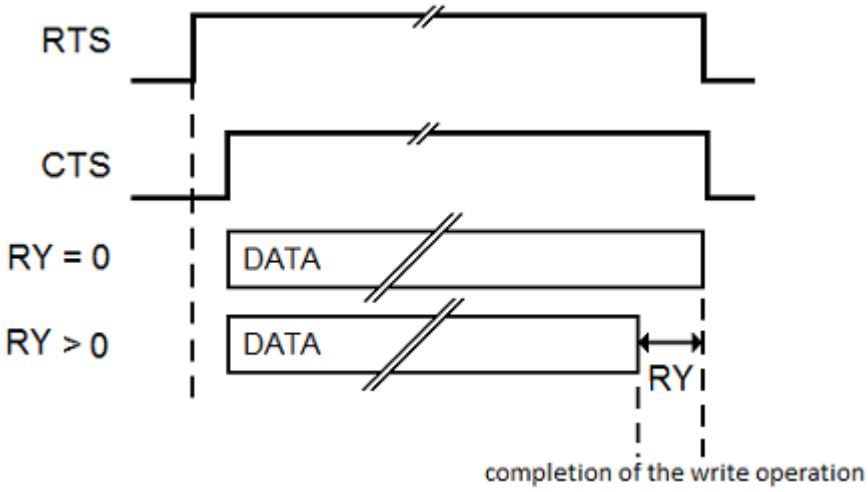


Figure 7: RY attribute

RK	RTS Keep Up Padding Characters

The number of padding characters (null characters) inserted to the end of a telegram to delay the passivation of the RTS (Request To Send) signal.

Data type:	Integer
Value:	0...255
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	Read, conditional write

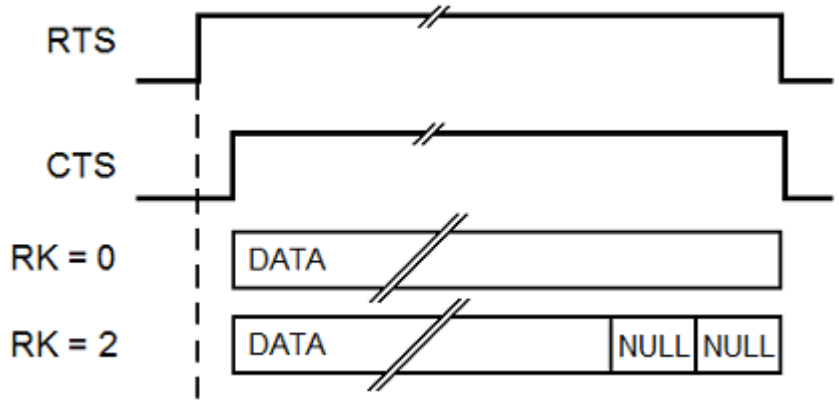


Figure 8: RK attribute

Instead of using RK attribute, the RTS Keepup Delay (RY) attribute can be used.

RI	**Receive Interrupt Enable Delay**

Defines the delay in milliseconds after which the receiver of a NET line is enabled after a message has been issued. If the given value is too big, the first characters of the response are not necessarily received.

Data type:	Integer
Value:	0...255 0 = receiver is always enabled
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	No limitations

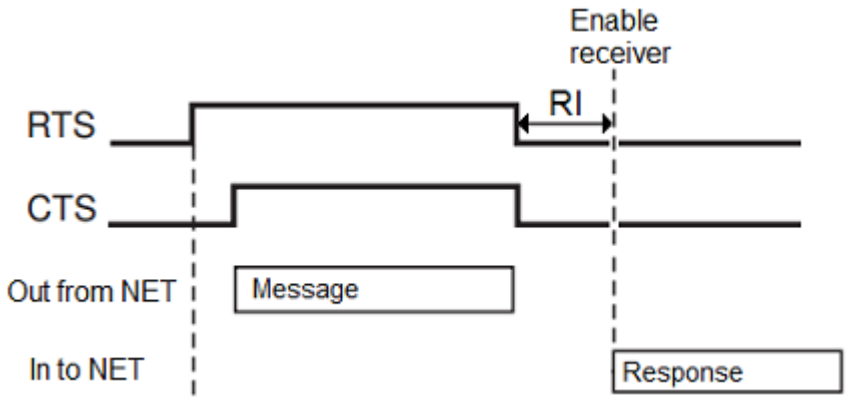


Figure 9: RI attribute

EN	**Enquiry Limit**

Specifies the maximum number of times that a message is retransmitted after a timeout.

Data type:	Integer
Value:	1...255

Table continues on next page

EN	Enquiry Limit
Index range:	1...12 (NET line numbering)
Default value:	1
Access:	Read, conditional write

SG	Modem Signal

An attribute for direct supervision and control of the state of the modem signal. This attribute applies to all protocols. It is used for diagnostics and testing.
If the incoming signal DCD or CTS is wanted to have a simulated high value all the time, value = 1 can be written to these signals. This feature may be necessary for easier cabling or with virtual serial ports. If value = 0 is written to these signals, the actual state of signal is used. The default mode of operation is the actual state. See also attribute CM Com Port Mode.

Data type:	Integer
Value:	0 = Passive signal 1 = active signal
Incoming:	DCD and CTS signals
Outgoing:	DTR signal
Index range:	100 * line no + signal no. Signal no. 5 = CTS, 8 = DCD, 20 = DTR
Access:	Read-only, write possible to signals 5 = CTS and 8 = DCD

Examples:

```
#SET NET1:SSG208 = 1 ; line 2 of NET1 should behave as DCD is 'high'
all the time
#SET NET1:SSG205 = 1 ; line 2 of NET1 should behave as CTS is 'high'
all the time
#SET NET1:SSG208 = 0 ; line 2 of NET1 should use the actual state of
the DCD
#SET NET1:SSG205 = 0 ; line 2 of NET1 should use the actual state of
the CTS
```

CM	COM Port Mode

This attribute consists of a set of flags which control the behavior and functionality of the serial port of the line. Each flag is one bit of this attribute. This attribute is meaningless if the line operates in LAN/WAN mode.

Data type:	Integer
Value:	0 ... 15 (see below)
Default value	0
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write
Bit 0:	UART error handling When this bit is 0, the UART errors are read before the bytes are read from the serial port. This is the default mode. When the bit is 1, the UART errors are read as a separate operation after the bytes are read from the serial port. This mode is similar to PC_NETs older than 9.2SP2 and it does not detect all errors detected by the serial port hardware. If the line has a lot of disturbances, this mode may result in better performance than the default mode.
Bit 1:	Simulated high of the CTS signal When this bit is 0, the actual state of the CTS signal is used in the protocol. This is the default mode. When this bit is 1, the CTS signal is simulated to be in the high state all the time and the line behaves according to that. This setting may be necessary with a virtual serial port or for easier cabling. See line attribute DE how transmission starts when CTS is constantly 'high'.

Table continues on next page

CM	COM Port Mode
Bit 2:	Simulated high of the DCD signal When this bit is 0, the actual state of the DCD signal is used in the protocol. This is the default mode. When this bit is 1, the DCD signal is simulated to be in the high state all the time and the line behaves according to that. This setting may be necessary with a virtual serial port or for easier cabling.
Bit 3:	Calculated RTS Keep up time When this bit is 0, the keep up time of the RTS signal is not calculated using the length of the message but it is assumed that the driver of the serial port blocks the execution of the sending process until the message is actually sent. This setting should be used if the setting 'Wait on physical transmission before completing write' or similar is selected in the driver for the port in question. The tuning of the RTS keep up time should be done with line attribute RY. This is the default setting. When this bit is 1, the keep up time of the RTS signal is calculated using the length of the sent message and the baud rate of the port. The RTS keep up time defined with the line attribute RY is added to the calculated time. This setting is not needed if the setting 'Wait on physical transmission before completing write' or similar is selected in the driver for the port in question. This setting is the most common with the virtual serial ports, too. If the serial driver does not provide setting 'Wait on physical transmission before completing write' or similar and RTS signal is actively used by the modem hardware, it is worth to test both alternatives. For accurate analysis using protocol analyzer function, see also the description of the bit 2 of the line attribute AU Analyzer Usage.

Having a simulated value in CTS or DCD may have an effect on how a RS-232 line disconnection is detected and reported to the MicroSCADA application.

MI	Message Identification
Object address of system messages.	
Data type:	Integer
Value:	1...32760
Index range:	1...12 (NET line numbering)
Default value:	6000 + (100 * NET number) + line number
Access:	Read, conditional write

MS	Message Application
The number of the application that is the receiver of the system messages generated by the line.	
Data type:	Integer
Value:	1...250
Default value:	1
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write

LK	Link Type
The type of data link connection used on the line. The states of the CTS and DCD signals of the serial port can have simulated values. The usage of this feature may be necessary, if the line uses a virtual serial port or the hardware connected to the serial port requires a special cable. See the description of the line attribute CM for more information.	
Data type:	Integer
Value:	14: Collision detection in use, transmission when the Data Carrier Detect (DCD) signal of the line is not set. 15: No collision detection, Data Carrier Detect (DCD) signal is handled as in other protocols.
Index range:	1...12 (NET line numbering)

Table continues on next page

LK	Link Type
Default value:	15
Access:	Read, conditional write

With values 0..10, the behavior is similar to 4: Radio Link, in which RTS/CTS controlling is used and the messages are received in unbalanced fashion. This value should be used in the unbalanced mode. In the balanced mode, the value must be 12 or 13.

LA	Link Layer Confirmations Enabled

Determines whether the link layer confirmations are in use (value 1) or not in use (value 0).

Data type:	Integer
Value:	0 or 1
Index range:	1...12 (NET line numbering)
Default value:	1
Access:	Read, conditional write

DC	Diagnostic Counters

The line protocols gather statistical information about the events on the lines by incrementing a number of diagnostic counters. All the major events and error situations of the communication have their own counters. When accessing diagnostic counters, the attribute is indexed according to the formula:
100 * (line number) + (diagnostic counter number)
The DNP 3.0 Slave protocol supports the following counters:

1. Transmitted telegrams
2. Failed transmissions
4. Transmitted commands
5. Transmitted replies
11. Received messages
12. Parity errors
13. Overrun errors
14. Check sum errors
15. Framing errors
16. Buffer overflow errors
20. TCP connect
21. TCP accept
22. TCP close

Data type:	Integer
Value:	0...30000
Index range:	See above
Access:	Read-only, the values can be reset

OM	Operating Mode

A bit pattern, which defines the operating mode of the line.

Data type:	Integer
Value:	0..65535 (see below)
Index range:	1..12 (NET line numbering)
Default value: 0	
Access:	Read, conditional write

Table continues on next page

OM	Operating Mode
Bit 0:	By default, the IP address of an incoming connection request is checked against the values defined with the station object attribute IA. If the incoming address does not match any station, the connection is not accepted. If bit 0 is set, the incoming connection request is not checked. This configuration may be useful in hot-standby systems in which there are two masters with separate IP addresses. Notice that no more than one connection per station object can be active at the same time. For example, if the master IP address should not be checked: #SET NET1:SOM1=1.
Bit 1:	When bit 1 is not set, new TCP connection attempts do not disconnect an existing connection. This is the default behavior. If bit 1 is set, a new TCP connection attempt disconnects an existing connection. This bit may be useful when there are multiple masters and it is known that the master which is trying to establish a connection is always the one which should continue the communication. This bit is meaningful only in TCP mode.
Bit 2:	When this bit is 1, the transmission of the "Test function for link" frame is disabled. This configuration may be useful when the collision detection is not done by the hardware. When this bit is 0, the transmission of the "Test function for link" frame is enabled as a default.
Bit 3:	When this bit is 1, the transmission of the "Reset of the remote link" frame is disabled. This configuration is useful when the link initialization is not needed in both directions, or if it is possible that this message collides with other transmitted frames from the other devices sharing the line. When this bit is 0, the transmission of the "Reset of the remote link" frame is enabled.
Bit 4:	When this bit is 1, the "Test function for link" messages are sent also when there is other activity on the line. In order to detect the disconnection of the master correctly, this setting may be useful only in serial or UDP modes and if the process data is sent without link layer acknowledgements and the application layer confirmations are not requested either. When this bit is 0, the "Test function for link" messages are sent only when there is no line activity within the time specified by the line attribute PD. This is the default mode and it is recommended in most configurations.

IT **Minimum Idle Time**

This attribute is important only when LK=14, that is, the DCD signal is used to indicate if the line is busy. The IT attribute defines the minimum time from the DCD low state to the start of the transmission. When the defined time has expired and DCD is still low, the random time from 0 to XR is waited before the transmission takes place. The attribute is meaningless if the line operates in LAN/WAN mode.

Value:	0 .. 255 milliseconds
Indexing:	Line number
Access:	Read/write
Default:	0

UI **UAL event Identification**

The UI attribute is used to define the name for the line object, and it is used to identify the source of the UAL (user activity logging) events. This string is added to the identification information of all user activity events from this line object. A unique value within the node is preferred. The node level module name (also in attribute UI) is added to form the full identification information for the user activity event. If a line identifier is not needed, an empty string should be assigned to this attribute.

Data type:	String
Value:	String containing a line level identifier with maximum length of 16 characters
Indexing:	1..12 (if not used, node attribute will be referred)
Default value:	”.LINEx”, where x = line number
Access:	Read, write

4.3.2.4 DNP 3.0 station object

The main purpose of the station layer is the protocol conversion between the DNP 3.0 and the internal protocol of SYS600. The station objects also take care of the application level communication with the master.

The STA objects created in a NET unit perform the functions of the station object. In DNP 3.0 Slave line, creation of only one STA object per line is possible. If more than one STA object is needed for NCC connections with the COM500*i* application, multiple DNP 3.0 Slave lines must be created.

The STA objects created in a NET unit perform the functions of the station object. Some attributes are used for the station configuration and others are used for device communication. The configuration attributes are presented in this chapter and the device communication attributes are presented in Section 5. By using the device communication attributes, messages can be sent and they can, for example, synchronize the slaves on the line.

Station objects can be configured to use DNP 3.0 Secure Authentication v2 or v5 using the attributes described in chapter 'Authentication attributes'. DNP 3.0 Secure Authentication v2 and v5 is based on IEC/TS 62351 and standards IEEE1815-2010 (v2) and IEEE1815-2012 (v5). Version v2 uses pre-shared update keys and does not contain roles for users. Version v5 is able to define users and their roles and keys on-line using DNP 3.0.

The databases for user sets and necessary keys are created using separate tools (see SYS600 System Configuration manual, chapter 'Secure authentication using IEC/TS 62351-5' for more information). This database is called "key storage" in the descriptions of the authentication attributes. The key storage which is used by the PC-NET instance is defined with NET Node attribute KS (see System Objects manual for a detailed description). Key storage file is always encrypted.

Chapter 'Security attributes' describes available options for communication encryption and certificate validation using TLS (Transport layer security). Functionality follows IEC technical specification 62351-3 and IEEE1815-2012. This function is applicable only in TCP/IP mode and should be used only together with the application layer secure authentication.

Station attributes

The following attributes can be used for configuring the DNP 3.0 Slave stations in SYS600.

IU	In Use
Indicates whether the station is in use (value 1) or not in use (value 0).	
Data type:	Integer
Value:	0 or 1
Default value:	0
Access:	No limitations

LI	Line Number
The number of the NET line the station is connected to.	
Data type:	Integer
Value:	1...12 (NET line numbering)
Access:	Read, conditional write

Setting this attribute is not needed, when the station is created by using the DV attribute.

SA	Station Address
The station address of the DNP 3.0 stationIn versions 9.3 FP2 and newer, it is possible to create more than one STA object to a slave line. All STAs connected to the same line are sending data to same master address (MA).	
Data type:	Integer
Value:	0...65534
Access:	Read, conditional write

Example:

In the example of the communication system configuration, the slave address is 1.

MA	Master Address
The station address of the master station , the destination address of the messages sent by the slave.	
Data type:	Integer
Value:	0...65534
Access:	Read, conditional write

IA	Internet Address
The IP address or the hostname of the remote host. The connection is established with a device in this address using port number 20000. If the line operates in the UDP mode and there is only one STA object connected to the line, it is possible to define fixed port numbers for the local port receiving the requests and the remote port number to which responses are sent. See examples below. The line needs to have been taken into use at least once before writing to this attribute. If routers/firewalls are used, it must be ensured that the defined port number is left open for connection.	
Value:	Any string, max 29 characters
Access:	Read/write

When IA is read, the configured IP address is returned, but the special remote and local port numbers are not.

When written, this attribute accepts the IP address in the following form:

```
#SET STA1:SIA=”192.162.1.120”
```

or as an alias name:

```
#SET STA1:SIA=”GRACE”
```

When an alias name is used, it must be defined in the TCP host file: %windir\system32\drivers \etc\hosts.

In UDP mode, a fixed local port is used, if it has been defined on the left side of a > separator. A fixed remote port is used, if it has been defined on the right side of the; separator. The listened IP address is defined with the line attribute LD.

```
#SET STA1:SIA="3113>192.168.0.1;3114"    ; requests are received to port
3113 and responded to port 3114 (UDP only)
```

If the local or remote port is not used, the default value 20000 is used. The local port is freely selectable, but it is not allowed to be used by any other application.

AI	Additional Internet Address

Additional IP addresses of the remote host. The connection is accepted from the master device if the incoming IP address matches the IA or any of the addresses listed in AI.
The given addresses must be separated with a space. The index defines the logical connection for the addresses. The line must be taken into use at least once before writing an address to this attribute.

Value:	Any string, max 200 characters
Index:	No indexing
Access:	Read/write

This attribute accepts the IP addresses in the following form:

```
#SET STA1:SAI="192.168.2.180 192.168.2.181"
```

Example 1:

```
#SET NET3:SOM1 = 0
;Bit 0, 'incoming connection address is not checked' functionality is
disabled
#SET STA1:SIA="192.168.2.180"
#SET STA1:SAI="192.168.2.181 192.168.2.182 192.168.2.183"
```

Connection is accepted from addresses 192.168.2.180, 192.168.2.181, 192.168.2.182 or 192.168.2.183 depending on which one is established first. Other attempts are rejected.

When connection is established and the bit 1 of line attribute OM is 0, a new TCP connection attempt from valid IP-address is rejected. If bit 1 of the line attribute OM is 1, the new TCP connection attempt is accepted and the existing connection is closed.

DR	Direction

States if the DNP Slave station acts as the station A (primary station) or as the station B (secondary station).

Data type:	Integer
Value:	0 or 1
Default value:	0 (secondary station)
Access:	Read, conditional write

IL	Information Address Length

The length of the data object address (index) used in the DNP 3.0 messages.

Data type:	Integer
Value:	1 or 2
Unit:	Octets
Default value:	2
Access:	Read, conditional write

AL	Allocation

Allocates the station to an application. When the AL attribute has the value 1, the station is reserved by the application specified by the AS attribute. All spontaneous messages from the station are sent to this application.

Data type:	Integer
Value:	0 or 1
Access:	No limitations

AS **Allocating Application**

Specifies the allocating application of the station (see the AL attribute). The allocating application gets all spontaneous process data from the station. This application is also the only one that is allowed to set the device communication attributes.

Data type:	Integer
Value:	0...250, 0 = no application
Access:	Read-only

When the AL attribute is set to 0, AS also gets the value 0.

MI **Message Identification**

Object address of system messages.

Data type:	Integer
Value:	1...32760
Default value:	30000 + STA object number
Access:	Read, conditional write

MS **Message Application**

The number of the application, that is the receiver of the system messages generated by the station.

Data type:	Integer
Value:	1...250
Default value:	1
Access:	Read, conditional write

SE **System Messages Enabled**

Specifies whether the system messages generated by the NET and related to the station are sent to applications (value 1) or not (value 0). By using this attribute, it is possible to disable the system messages related to the station.

Data type:	Integer
Value:	0 or 1
Default value:	1
Access:	No limitations

CA **Command Address**

The object address of the bit stream process object in the SYS600 process database, where unidentified messages are sent. If the value of the CA attribute is 0, the unidentified messages are not sent, and the bit stream object is not updated.

Data type:	Integer
Value:	0...65534
Default value:	32000
Access:	No limitations

The unit number (UN attribute) of the bit stream process object must be the same as the STA object number.

ML	Maximum Message Length

The maximum length of an application data fragment (APDU).

Data type:	Integer
Value:	249...2048
Unit:	Octets
Default value:	2048
Access:	No limitations

PC	Process Data Confirmation

By setting the value of this attribute to 0, application level confirmations can be disabled. They can be enabled and by setting the value to 1. In the following cases the DNP 3.0 Slave station sets the confirmation request of a data fragment on, regardless of the value of the PC attribute:

- The sent data fragment contains event data.
- The response message consists of multiple data fragments.

Data type:	Integer
Value:	0, 1 or 2
Default value:	1 (application level confirmations enabled)
Access:	Read, conditional write

The station configuration PC=0 disables the application layer confirmations in most cases, but multifragment messages and messages containing event data still request for the application layer confirmation. This implementation follows the recommendation in the DNP 3.0 standard. A special value PC=2 is provided to also disable the application layer confirmations with these messages.

AR	Application Message Data Retries

The maximum number of retransmissions of an application data fragment (APDU). It is recommended to keep this setting at 0 and use link layer retries instead. This is also the recommended configuration of dnp.org.

Data type:	Integer
Value:	0...5
Default value:	0
Access:	No limitations

TC	Time Synchronization

Determines the behavior of the slave device when it receives a time synchronization message as follows:

Data type:	Integer
Value:	0...2 Value 0 = The synchronization message is handled and the clock of the base system is set to the received time. 1 = The message is acknowledged (positive acknowledgement), but the clock of the base system is not set. The slave station never sets the "time synchronization needed" bit in its responses. 2 = The message is acknowledged (negative acknowledgement), but the clock of the base system is not set. The slave station never sets the "time synchronization needed" bit in its responses.
Default value:	0
Access:	No limitations

RM	Running Mode

Consists of a set of flags that control the behavior and functionality of the DNP Slave station. Each flag is one bit of this attribute. The bits are as follows:

Bit 0:	Sending messages while waiting for a confirmation. When this bit is 0 the sending of a new message other than confirmation may not be started, if the DNP 3.0 slave station is waiting for a confirmation from a remote station. The message, other than confirmation, may be a response to a request. When this bit is 1, the sending of a new APDU (other than confirmation) may be started, even if the STA object is waiting for a confirmation from the master.
Bit 1:	Variations in response messages. When this bit is 0, NET uses dynamic variations in response messages. The variations depend on the status flags of the data object and they can vary between with and without status types. When this bit is 1, the variations are fixed. The NET unit always replies with the same variation as the one used in the master's request.
Bit 2:	Address offset usage in command receiving. When this bit is 0, no address offset is used. The address/index of the object in command is used "as it is" in process object updating. Thus, the address of the process object in the database and the address of the incoming object are equal. When this bit is 1, an offset TYPE*(2^24) is added to address/index of the incoming object and the process objects must be created with these addresses. The possible TYPEs are 12 (Control relay output block) and 41 (Analog output block).
Bit 3:	Unsolicited responses disabled in start-up. When this bit is 1 and the station object is taken into use (IU set to 1), unsolicited responses are not sent until the master has enabled the transmission using the function 20 (Enable Unsolicited Responses) request. When this bit is 0, the transmission of the unsolicited responses is enabled as a default. Unsolicited responses are used only if the NET database has been configured to quiescent mode.
Bit 6:	Handling of the responses containing event data. When this bit is 0, the response containing events consists of only one APDU. This is a safe behavior, especially when the master issues a class 1/2/3/0 request. This is the default behavior. When this bit is 1, the response containing events may consist of multiple APDUs. If the master issues a class 1/2/3/0 request and there are several events in the queue, the response may be too long for the application layer response timeout of the master and the communication may be disturbed.
Bit 7:	When this bit is 0, the responses containing time are in local time (default behaviour). When this bit is 1, the responses are UTC time. Bit does not have an effect on process data events generated using attribute EV where optional parameter EVCTRL is used instead.
Bit 8:	Analog event overwrite mode. When this bit is 1, a new analog event overwrites an existing unsent analog event with the same index. This feature may be used if there is a need to limit the analog updates sent from the system. This overwrite is applicable only to Object group 32. When this bit is 0, all data items defined as events are queued and transmitted. This is the default behaviour.

Data type:	Integer
Value:	0...65535
Default value:	2
Access:	No limitations

DC **Diagnostic Counters**

The values of the diagnostic counters which the NET unit keeps for the station. The counters have the following meaning:

1. Suspension information (0 = OK, 1 = suspended)
2. Suspension counter
3. Transmitted data messages
4. Transmitted command messages
5. Transmitted confirmation messages
6. Received data messages
7. Received command messages
8. Received confirmation messages
9. Received unknown messages
10. APDU in queue length
11. APDU out queue length
12. TSDU in queue length
13. TSDU out queue length
14. WAIT CONFIRM queue length
15. SYS transition queue length
16. Confirmation transition queue length
17. Select transition queue length
18. Free APDUs queue length
19. Free events queue length
20. Free SYS transitions queue length
21. WAIT RESPONSE queue length
22. Unsolicited enabled bitmask
23. WAIT RESPONSE timeouts
24. WAIT CONFIRM timeouts
25. LINK CONFIRM timeouts
26. SYS transition timeouts
27. APDU receive timeouts
28. Delayed scm timeouts
29. Select timeouts
30. Cf write timeouts
31. Commanded station address
32. TCP connects
33. AUTH out queue length
34. WAIT AUTH REPLY queue length
35. WAIT PROCESSING queue length

Data type:	Integer
Value:	1...65535
Index range:	1...35
Access:	Read-only, the values can be reset

OS **Object Status**

The current status of the DNP station object. When value 1 is written to this attribute, the station object retransmits its current status code .

Data type:	Integer
Value:	when Read, 0 = OK_STATUS or non-zero value = communication is not normal at the moment
Access:	No limitations

ST **SYS Waiting Time**

The maximum time in milliseconds that the station waits for a reply from the base system.

Data type:	Integer
Value:	0...60000
Unit:	Milliseconds
Default value:	5000
Access:	No limitations

RT **Reply Timeout**

Not in use at the moment.

IN **Internal Indications**

The current value of the internal indications of the DNP 3.0 Slave station. See the DNP 3.0 protocol documentation for details of the internal indications. If a value is written to IN, the given value is stored as IIN_VALUE and the IIN of the next response is a result of the operation defined with parameter OPERATION between IIN_VALUE and current IIN set by the DNP station object. If the parameter PERMANENT is given and its value is 1, the mentioned operation is made to all responses until otherwise specified. If OPERATION is not given, value 0 = no operation becomes active. If PERMANENT is not given, value 0 (= in next response only) becomes active.

Data type: Integer(when read) or vector of integers (when written)

Value: when read, 0...65535
when written, vector (IIN, [OPERATION], [PERMANENT]) where
IIN = 0..65535 is IIN_VALUE
OPERATION :
0= no operation
1 = bitwise OR
2= bitwise AND
3= bitwise XOR
PERMANENT:
0 = in next response only
1 = in all responses

Access: Read/write

Examples:

```
#SET sta1:sin=(256*255+0, 1, 0) ; IIN in next response has bits 2.0-2.7
set (Current IIN or 0xFF00)
#SET sta1:sin=(256*0+128, 1, 1)    ; IIN in response has bit 1.7 set
permanently (Current IIN or 0x0080)
#SET sta1:sin=(256*255+241, 2, 1); IIN in response bits 1.1-1.3 are not
set permanently (Current IIN and 0xFFF1)
```

CT **Confirmation Timeout**

The maximum time in seconds that the slave station waits for an application layer confirmation from the master.

Data type: Integer

Value: 0...600

Unit: Seconds

Default value: 10

Access: No limitations

TT **Transport Layer Timeout**

The maximum time in seconds that the transport layer is allowed to assemble one application message fragment.

Data type: Integer

Value: 0...600

Unit: Seconds

Default value: 10

Access: No limitations

ET	Execute Waiting Time After Select

The maximum time in seconds that the slave waits for an execute command after receiving an operate command.

Data type:	Integer
Value:	0...600
Unit:	Seconds
Default value:	30
Access:	No limitations

SB	SuBset

This attribute defines and indicates the subset level that is currently used. This value changes automatically, if the remote end transmits a message that belongs to a higher subset level.

Value:	2..3
Access:	Read/write
Default:	2

QC	Queue Clear

If the master accesses the object group and variation defined with this attribute, the event queues are cleared. This function can be used with redundant communication links where the master indicates the standby state of the line with special requests. When a read request matching the QC is received from the master, the unsolicited data transmission is also disabled for all classes, and the master must enable it again by using function 20 (Enable Unsolicited Responses). Group 0 variation 0 means that the feature is not used. This is the default functionality.

Data Type:	Integer
Value:	0..255
Indexing:	1 = Object group 2 = Object variation
Unit	No unit
Default:	0,0 (feature not used)
Access:	No limitations

Example:

If QC(1) is 80 and QC(2) is 2, a READ request of object group 80 (Internal indications) variation 2 (not defined) clears the event queues of classes 1,2 and 3 and also disables the unsolicited transmission of events from these classes.

SF	Status Filtering

This attribute specifies a filter time for returned error statuses of attribute writings/readings. If a non-zero value is given to the SF-attribute, the station object checks whether the same error status is to be returned within the time specified with SF, it is replaced with OK_STATUS for the SCIL application. This behaviour limits the error code amount in notification window and in the sys_error.log file. The status is not modified to OK_STATUS in following conditions:

1. SF is 0

2. The status is different from the previous status

3. The time defined with SF has been changed

Table continues on next page

SF **Status Filtering**

If the previous returned status is modified to OK_STATUS and the status to be returned changes back to OK_STATUS, status
13953 = DNPC_DATA_WRITE_STATUS_CHANGES_TO_OK is returned.
This is needed to see the timestamp of the actual moment of the status returning back to OK_STATUS. If the feature provided with the SF attribute is used, The SCIL application should handle the mentioned status as OK_STATUS.
If the value of SF is 0, filtering is not used and the functionality is the same as in version 9.4 or older. The value of SF can be set on-line without taking the station object out of use and it is effective immediately. If index 2 of the SF attribute is read, it returns the amount of filtered error statuses from previous returned error. The feature can be used to verify the actual amount of data updates when error filtering is used (SF>0).

Data Type:	Integer
Value:	0..180
Unit	Seconds
Default:	0 (= filtering not used)
Access:	No limitations (index 2 is read-only)

Example:

#SET STA1:SSF=10

An error status from Station object 1 is printed out with 10 second interval. This configuration may be used with a station object connected to a DNP 3.0 Slave line with the COM500*i* application. When the NCC connection is broken and the event queues are full, without this setting each new event or analog update would generate a printout which fills the notification window and the sys_error.log. With the example setting above, the amount of the printouts is much less and the off-line analysis of the problem situation is easier.

UA **UAL Event Used**

Attribute UA defines whether the UAL (User activity logging) events are generated by the station object. Generation of the UAL events are recommended if the secure authentication is used, see attribute ZA. With ‘Standard logging’, all user status changes are logged. With ‘Special logging’, also all successful authentications, key change negotiations, authorization failures and error situations are logged. The event codes with 'Special logging' are supported by SYS600 but not necessarily, for example, with SDM600 product. In case the secure authentication is used and more detailed information from the system behaviour is needed, the usage of value 3 = 'Special logging' is recommended.

Data type:	Integer
Value:	0 = Disabled 1 = Standard logging 2 = Extended logging 3 = Special logging
Indexing:	No indexing
Default value:	1 (Enabled)
Access:	No limitations

UI **UAL Event Identification**

The UI attribute is used to define the name for the station object, and it is used to identify the source of the UAL (user activity logging) events. This string is added to the identification information of all UAL events from this station object. A unique value within the node is preferred. The node level module name (also in attribute UI) is added to form the full identification information for the user activity event. The default value of the string is based on the translated station number which is equal to B-attribute TN (Translated Object Number) in the base system. If a station identifier is not needed, an empty string should be assigned to this attribute.

Data type:	String
Value:	String containing a station level identifier with maximum length of 16 characters
Indexing:	No indexing
Default value:	”.STA(TN=x)”, where x = translated station number
Access:	No limitations

4.3.2.5 Authentication attributes

This chapter describes only the attribute interface related to secure authentication. See SYS600 System Configuration Manual, chapter 'Secure authentication using IEC/TS 62351-5' for detailed steps for configuring the system for secure authentication according to IEC/TS 62351-5.

ZA **Authentication Used**

The ZA attribute defines whether secure authentication is used by the station object or not.

Value:	0 = Disabled 1 = Enabled, pre-shared update keys (v2) 2 = Enabled, update key negotiation (v5)
Indexing:	No indexing
Default value:	0 (Disabled)
Access:	Read/write

ZG **Aggressive Mode**

The ZG attribute defines whether the aggressive mode of authentication is used by the station object. The aggressive mode uses less bandwidth and using it is recommended. The value of this attribute is meaningless if authentication is disabled (see attribute ZA). Modifying this attribute is possible only if it is enabled in the key storage using the setting 'Allow external modification of security attributes' (see attribute DZ, index 255).

Value:	0 = Aggressive mode disabled 1 = Aggressive mode enabled
Indexing:	No indexing
Default value:	1 (Enabled)
Access:	Read, write possible if the modification is enabled in the key storage

ZT **Key Storage Id**

The ZT attribute is used to define the keys and user set of the STA object in the key storage. In case there is a need to change the station number and/or its translated number TN and the corresponding user set and key configuration is already configured, keeping the original value in the ZT attribute associates the existing user set with the new station object. The ZT value must be unique within the STA objects accessing the same key storage. Error is returned when setting to a reserved value is attempted. Value = 0 means that the station object is not attached to any user set and the enabling of the authentication using attribute ZA is not possible.

Value:	Identification of the STA object in the key storage, range 0..65535
Default value:	Same as the TN of the STA object
Indexing:	No Indexing
Access:	Read/write

ZU **Default User**

The ZU attribute is used to define the user in case the user is not or cannot be received from the MicroSCADA application. This attribute is meaningless in STA objects connected to slave lines.

Value:	When read, the number of the active user is returned When written, number or the name of the wanted active user
Default value:	1 (Default user)
Indexing:	No indexing
Access:	Read/write

ZR	Authenticated Users

The ZR attribute is used to read the user numbers and names defined in the key storage for the station. See also the ZU attribute for defining the default user. This attribute is indexed using the user number and it is not supported by the System Configuration tool.

Value:	String containing a user name with the maximum length of 32 characters.
Indexing:	1..65535 (User number) If index = 0 is given when read, a vector of configured user numbers for the STA object is returned. If only one user is configured, the returned value is a scalar of type integer. If ZA is 0 or no users is configured, no object is returned.
Access:	Read

ZI	Association Id

The ZI attribute is used to define the association identification value of the user. This attribute is used to fully identify the user. The attribute is indexed using the user number and is not supported by the System Configuration tool.

Value:	The association id of the user defined by the index
Default value:	0
Indexing:	1..65535 (User number)

ZV	Authentication Vector

The ZV attribute defines the constants used by the specific user of the station object. Modifying these constants is possible but it may require some testing to be usable with the remote IED and its configuration. The given index defines the user, value 0 has a special meaning. The values of the vector cannot be modified one-by-one but when written, only a vector containing all values is accepted.

The session key length indicated with ZV(1) automatically adapts to the key length defined in master when session keys are negotiated. Length range is 16..32 bytes.
If the interval defined with ZV(8) has expired twice without session key renegotiation between the expirations, the session keys for the user are invalidated and the corresponding UAL event is reported. After this, no critical operations for the user are authenticated in either directions and session keys must be renegotiated. See also attribute ZP, index 1.
If the count of the authentication errors with same session keys reaches 2*ZV(6), slave stops sending the authentication error messages. A successful session key negotiation for the user restores the functionality.
The attribute is indexed using the user number and is not supported by the System Configuration tool. Index 0 contains special compatibility flags applied for all users.

Value:	For index 0, vector of max 3 integers ZV(1) User name null termination (1 = User name transmitted without null termination (default)) (2 = User 'Common' with null termination, others without) (3 = User names transmitted with null termination) ZV(2) Authority certification key length with SHA-1 Update Key Change Method (16= First 16 bytes used in MAC calculation) (32 = All 32 bytes used in MAC calculation (default)) ZV(3) Not used at the moment For indices > 0, vector of 10 ZV(1) Session key length ZV(2) Key wrap algorithm of the session keys (1=AES-128) (2=AES-256) ZV(3) Challenge data length critical request ZV(4) HMAC algorithm of the critical requests (1 = SHA-1) (2 = SHA-256) ZV(5) HMAC length (4 = SHA-1 serial) (10 = SHA-1 network) (8 = SHA-256 serial) (16 = SHA-256 network) ZV(6) Max error count ZV(7) Session key change count ZV(8) Key change interval in seconds ZV(9) Challenge data length session key status ZV(10) Challenge data length update key reply
Default values:	Index 0: ZV(1) : 1 ZV(2) : 32 ZV(3) : 0 Indices >0: ZV(1) : 16 ZV(2) : 1 (AES-128) ZV(3) : 8 ZV(4) : 1 (SHA-1) ZV(5) : 10 (SHA-1 network) ZV(6) : 2 ZV(7) : 1000 ZV(8) : 900 ZV(9) : 8 ZV(10) : 32
Indexing	1..65535 (User number), 0 (compatibility flags)
Access:	Read-only

Example 1:

```
@USERS = STA1:SZR
#LOOP_WITH I=1..LENGTH(%USERS)
     ; Change each user to use SHA-256 and AES-256 in TCP mode, session
```

```
key change interval 120 secs
    @USER=%USERS(%I)

    #SET STA1:SZV'USER'=(16,2,8,2,16,2,1000,120,8,32)

#LOOP_END
```

Example 2:

```
#SET STA1:SZV0=(1,32,3); No null termination, 32 bytes certification key
for SHA-1 (Triangle Microworks configuration), commands sent using first
'SingleUser'
#SET STA1:SZV0=(3,16,2); Null termination, 16 bytes certification key for
SHA-1 (ASE2000 configuration), commands using first 'Operator'
```

ZS	Authentication Status

The ZS attribute returns the current values of the authentication parameters and states of the specific user. This attribute is implemented only for debugging purposes and modifying these values directly is not possible.
The attribute is indexed using the user number and is not supported by the System Configuration tool.

Value:	Vector of 8 integers ZS(1) State of the authentication ZS(2) Not used at the moment ZS(3) Not used at the moment ZS(4) Challenge sequence number CSQ ZS(5) Key change sequence number KSQ ZS(6) Key status KST ZS(7) Last error in ZS(8) Last error out Values of Key Status in ZS(6) are : 1 = OK 2 = Not initialized 3 = Communication failure 4 = Authentication failure
Default value:	ZS(1) : 0 ZS(2) : 0 ZS(3) : 0 ZS(4) : 0 ZS(5) : 0 ZS(6) : 2 ZS(7) : 0 ZS(8) : 0
Values of ZS(1)	0 : Initial State 10 : Wait for Key Status 11 : Wait for Key Change Confirmation 12 : Security Idle 13 : Wait for Reply 14 : Wait for User Change Response 15 : Wait for Update Key Reply 16 : Wait for Update Key Confirm
Indexing:	1..65535 (User number)
Access:	Read-only

ZO	User Roles

The ZO attribute is used to return user numbers and their roles for the authentication. The user set for the station is created using the authority tool and stored to the key storage file.

Value:	String containing a user role with maximum length of 32 characters. Standard roles are: 0 = 'Viewer' (has 'Monitor Data' permission) 1 = 'Operator' (has 'Monitor Data' and 'Operate Controls' permissions) 2 = 'Engineer' (has 'Monitor Data', 'Change Config', 'File access' and 'Local login' permissions) 3 = 'Installer' (has 'Monitor Data', 'Change Config', 'Change code', 'File Access' and 'Local login' permissions) 4 = 'SecAdm' (has 'Change Security Config', 'Change code' and 'Local login' permissions) 5 = 'SecAud' (has 'Monitor Data' and 'Local login' permissions) 6 = 'RBacMnt' (has 'Monitor Data', 'Change Config' and 'Change Security Config' (roles only) permissions 32768 = 'Singleuser' (has all permissions) See attribute CR for more information. Permissions applicable to DNP 3.0 slave implementation in SYS600 are 'Monitor Data', 'Operate Controls' and 'Change Config'.
Default value:	Empty string
Indexing:	1..65535 (User number)
Access:	Read-only

ZN	Outstation Name

The ZN attribute is used to return the name of the outstation. This attribute is preconfigured using the Authority Tool and must match the value configured to master/slave.

Value:	String containing an outstation name with maximum length of 128 characters
Default value:	Empty string
Indexing:	No indexing
Access:	Read-only

ZD **Authentication Diagnostics**

The ZD attribute provides user specific counters for diagnostic purposes. These counters can be used to verify the activity of each user and to give information in problem situations. The same counters for all users can be read from the attribute DZ.

Value: Vector of 17 integers (0..65535)
ZD(1) Unexpected Messages
ZD(2) Authorization Failures
ZD(3) Authentication Failures
ZD(4) Reply Timeouts
ZD(5) Rekeys Due to Authentication Failure
ZD(6) Total Messages Sent
ZD(7) Total Messages Received
ZD(8) Critical Messages Sent
ZD(9) Critical Messages Received
ZD(10) Discarded Messages
ZD(11) Error Messages Sent
ZD(12) Error Messages Received
ZD(13) Successful Authentications
ZD(14) Session Key Changes
ZD(15) Failed Session Key Changes
ZD(16) Update Key Changes
ZD(17) Failed Update Key Changes

Default values: ZD(1) ..ZD(17) : 0

Indexing: 1..65535 (User number)

Access: Read/write (write only value 0)

DZ **Diagnostics of Authentication**

The DZ attribute provides diagnostic counters of authentication related functions. These counters can be used to verify the activity of communication concerning data which requires authentication. The same counters can be read as user specific from the attribute ZD.
Index 255 has a special meaning. If DZ(255) returns 1, the editing of the attributes CR and ZG is allowed by a special setting in the key storage.

Value: Vector of 17 integers (0..2147483647)
DZ(1) Unexpected Messages
DZ(2) Authorization Failures
DZ(3) Authentication Failures
DZ(4) Reply Timeouts
DZ(5) Rekeys Due to Authentication Failure
DZ(6) Total Messages Sent
DZ(7) Total Messages Received
DZ(8) Critical Messages Sent
DZ(9) Critical Messages Received
DZ(10) Discarded Messages
DZ(11) Error Messages Sent
DZ(12) Error Messages Received
DZ(13) Successful Authentications
DZ(14) Session Key Changes
DZ(15) Failed Session Key Changes
DZ(16) Update Key Changes
DZ(17) Failed Update Key Changes

Default values: DZ(1) ..DZ(17) : 0
DZ(255) : 0 (attribute editing not allowed)

Indexing : 255 : editing of the attributes CR and ZG is allowed by a special setting in the key storage
1..17 : Diagnostic counters mentioned above, is reset
101..117 : Diagnostic counters mentioned above, is unreset
201..217 : Diagnostic counters mentioned above, next threshold value for special action

Access: Read/write (write only value 0 for indices 1..17)

CR	Critical Requests

The CR attribute defines which requests are critical. The modification of this vector is possible but not recommended.
The default values follow the DNP3.0 standard. Modifying this attribute is possible only it is enabled in the key storage using the setting 'Allow external modification of security attributes' (see attribute DZ, index 255). Modifying this attribute is not possible for functions which are not applicable. In DNP3, control commands are sent using functions 3..6 which are critical as default. With v5 authentication, the required permissions for a user are 'Operate Controls', see attribute ZO for more information. Index of the CR attribute defines the DNP3 function in question.
This attribute is not supported by the System Configuration Tool.

Value: Vector of 131 integers with values
0 = not critical
1 = critical
2 = critical only remotely (applies to function 0x00 Confirm only)
3 = critical only locally (applies to function 0x00 Confirm only)
128 = not applicable

Default values: Required permissions are given after the function name. If not mentioned, 'Monitor data' permissions for the function are adequate.
CR(0) : 0 (Function 0x00 Confirm)
CR(1) : 0 (Function 0x01 Read)
CR(2) : 1 (Function 0x02 Write)
CR(3) : 1 (Function 0x03 Select) ('Operate controls')
CR(4) : 1 (Function 0x04 Operate) ('Operate controls')
CR(5) : 1 (Function 0x05 Direct Operate) ('Operate controls')
CR(6) : 1 (Function 0x06 Direct Operate - no ack) ('Operate controls')
CR(7) : 0 (Function 0x07 Immediate Freeze)
CR(8) : 0 (Function 0x08 Immediate Freeze - no ack)
CR(9) : 0 (Function 0x09 Freeze and Clear)
CR(10) : 0 (Function 0x0A Freeze and Clear - no ack)
CR(11) : 0 (Function 0x0B Freeze-at-time)
CR(12) : 0 (Function 0x0C Freeze-at-time - no ack)
CR(13) : 1 (Function 0x0D Cold restart)) ('Change config')
CR(14) : 1 (Function 0x0E Warm restart) ('Change config')
CR(15) : 0 (Function 0x0F Initialize Data) ('Change config')
CR(16) : 1 (Function 0x10 Initialize Application)('Change config')
CR(17) : 1 (Function 0x11 Start Application) ('Change config')
CR(18) : 1 (Function 0x12 Stop Application) ('Change config')
CR(19) : 0 (Function 0x13 Save Configuration) ('Change config')
CR(20) : 1 (Function 0x14 Enable Unsolicited Responses)
CR(21) : 1 (Function 0x15 Disable Unsolicited Responses)
CR(22) : 0 (Function 0x16 Assign Class)
CR(23) : 0 (Function 0x17 Delay Measurement)
CR(24) : 1 (Function 0x18 Record Current Time)
CR(25) : 0 (Function 0x19 Open File)
CR(26) : 0 (Function 0x1A Close File)
CR(27) : 0 (Function 0x1B Delete File)
CR(28) : 0 (Function 0x1C Get File Information)
CR(29) : 1 (Function 0x1D Authenticate File)
CR(30) : 0 (Function 0x1E Abort File)
CR(31) : 1 (Function 0x1F Activate Configuration) ('Change config')
CR(32) : 128 (Function 0x20 Authentication request)
CR(33) : 128 (Function 0x21 Authentication request - no ack)
CR(34)..CR(128) : 128 (Functions 0x34 .. 0x80 not defined by the standard)
CR(129) : 0 (Function 0x81 Response)
CR(130) : 0 (Function 0x82 Unsolicited Response)
CR(131) : 128 (Function 0x83 Authentication Response)

Indexing : Index 0..131 (defines the function code)

Access: Read, write if allowed by the key storage config

NU	New Keys

The NU attribute is used to trigger the renegotiation of the new TLS session keys for the TCP/IP. This attribute is meaningful only if the communication method is TCP/IP and the communication encryption using the attributes described in chapter 'Security attributes' is configured. When value 1 is written to attribute NU, the TLS level session key renegotiation is started when the TCP/IP transmission from the DNP3.0 slave line starts for the next time. DNP3.0 standard defines that the renegotiation should be made regularly, preferably with the same interval as the authentication session keys are renegotiated. The default interval for the authentication session key renegotiation presented by the DNP 3.0 standard is 15 minutes (900 seconds).

Value:	1
Indexing:	No indexing
Access:	Write

Example 1:

Renegotiate the TLS session keys for the TCP/IP connection used by the station object

```
#SET STA1:SNU=1 ; renegotiate TLS session keys for STA1
```

4.3.2.6 Security attributes

This chapter describes only the attribute interface related to encryption using TLS (IEC62351-3). These attributes should only be used together with the secure authentication, see Section 'Authentication attributes' and SYS600 System Configuration manual.

All TLS connections in one PC-NET instance must use same certificate and key and trusted certificate authority files, see attribute CI Certificate Information. Furthermore, all TLS connections in one PC-NET instance must use same min/max values of TLS version, see attribute CV Certificate Handling Vector

CI	Certificate Information

The CI attribute is used to define the certificate/key file and the trusted certificate authority file for the TLS communication. When set, it has an effect on all STA objects that have the same remote IP-address in attribute IA. Also, by setting both CI indices the TLS functionality is started. The certificate files location can be chosen freely, but it is recommended to place them in a folder with limited access rights to improve the security of the system. For example, use a folder where only the MicroSCADA user has access rights. Same certificate and trusted certificate authority file should be used in all STA objects configured to one PC-NET instance. This attribute is only used in TCP mode.

Value:	String containing the directory and the name of the certificate file. Only certificate file type .PEM is supported.
Indexing:	1 : Certificate and Key file 2 : Trusted certificate authority file
Default value:	Empty string
Access:	Read/conditional write

Example:

```
#SET STA1:SCI1="C:\Users\MicroSCADA\AppData\Roaming\ABB\MicroSCADA Pro
\Device_STA1.pem"
#SET STA1:SCI2="C:\Users\MicroSCADA\AppData\Roaming\ABB\MicroSCADA Pro
\CA_list.pem"
```

The TLS functionality is activated when both CI attribute indices are set.

CP **Certificate Passphrase**

The CP attribute is used to define the passphrase to open the private key in the certificate file defined with attribute CI(1). If the certificate does not require passphrase, this attribute may be an empty string. If the creation of the self-signed certificates is enabled using attribute CV(2), the contents of this attribute is included to the passphrase of the private key of the created certificate. When set, it operates like attribute CI, that is, it has an effect on all STA objects that have the same remote IP-address in attribute IA. This attribute is used only in TCP mode.

Value: String containing the passphrase with maximum length of 16 characters

Indexing: No indexing

Default value: Empty string

Access: Read/conditional write

Example:

```
#SET STA1:SCP="96gd21"
```

CN **Common Name**

The CN indices 1, 2 and 4 are used to define the Common Name, the Country Code and the Organization name of a self-signed certificate. All 3 values must be set when creating a self-signed certificate. The CN(1), the Common Name, should be the IP address of the station. In CN(2) the Country Code should be given with two capital letters. The CN(4), the Organization name, should be a single name without spaces. The name and location of the created certificate is defined using attribute CI(1). When self-signed certificate is used, the CN attribute operates like CI attribute, that is, it has an effect on all STA objects that have the same remote IP-address in attribute IA. CN indices 5 and 6 refer to Organization Unit and Domain Component and can be left empty. If set, corresponding OU and DC strings must be found from CN(3). For detailed instructions for creating a self-signed certificate, see SYS600 System Configuration manual.

If the certificate file contains multiple certificates, the CN index 3 is used for selecting a certain certificate that will be used. In CN(3), the CN(1), CN(2) and CN(4) needs to be listed and they must have the exact values that were set to the certain certificate. The format of CN(3) can be seen in the example below. If only one certificate exists in the certificate file, CN(3) can be left empty. CN attribute is used only in TCP mode.

The CN attribute indices 1, 2 and 4 are needed only when a self-signed certificate is created. The creation of self-signed certificate is selected with CV attribute bit 2 values 1 or 2.

Value: String

Indexing:
1 : Common name of the certificate with maximum length of 64 characters
2 : Country code with maximum length of 2 characters
3 : Certificate name with maximum length of 228 characters
4 : Organization name with maximum length of 64 characters
5 : Organization Unit name with maximum length of 64 characters
6 : Domain Component with maximum length of 64 characters

Default value: Empty string

Access: Read/conditional write

Example:

```
#SET STA1:SCN1="10.10.10.1"
#SET STA1:SCN2="FI"
#SET STA1:SCN4="ABB"

#SET STA1:SCN3="CN=10.10.10.1 O=ABB C=FI"
```

Example 2:

```
#SET STA1:SCN1="10.10.10.1"
#SET STA1:SCN2="FI"
#SET STA1:SCN4="ABB"
#SET STA1:SCN5="PSS1" ; optional
#SET STA1:SCN6="COM" ; optional

#SET STA1:SCN3="CN=10.10.10.1 O=ABB C=FI OU=PSS1 DC=COM"
```

CV	Certificate Handling Vector

The CV attribute defines a set of values which control error logging and certificate handling, creation and accepting. The modification of these attributes from the defaults may decrease the total security of the system. When set, it operates like attribute CI, that is, it has an effect on all STA objects that has the same remote IP-address in attribute IA. Attribute is used only in TCP mode.
CV(1) indicates if possible error messages are logged into the MicroSCADA Notify window. If given value is 2, all SCA printouts are directed to Notify Window. This mode should be used only temporarily.
CV(2) indicates if self-signed certificates are created when the communication is started. 0, 'Never', should be used when certificates already exist. This also applies to situations in which self-signed certificates are used. With option 1, 'If not found', a self-signed certificate is created in case the certificate set in CI(1) doesn't already exist. With setting 2, 'Always', a self-signed certificate is always created even if the certificate file set in CI(1) already exists. In this case the existing file is replaced with a new file.
CV(3) indicates the action that will occur when the remote certificate validation fails. With value 0, 'Ignore, continue communication', the communication is not terminated even though the remote certificate is found faulty or cannot be authenticated. This option should only be used in special situations, for example in testing. Value 1, 'Close connection', terminates the connection when the remote certificate validation fails. This is the default behavior and it is recommended to be used. Functionality with value 3 is similar to value 1 but the certificate from remote system does not require to have the IP-address in its 'Common Name' field.
CV(4) defines the accepted certificate file type. Currently only type .pem files are supported.
CV(5) defines the minimum TLS protocol version that is accepted from the remote certificate. In case the remote device uses TLS version lower than what is defined in CV(5) the communication is terminated (unless the CV(3) is set to 0). For highest security it is recommended to use the highest TLS version possible.
CV(6) defines the maximum TLS protocol version that is accepted from the remote certificate. In case the remote device uses TLS version higher than what is defined in CV(6) the communication is terminated (unless the CV(3) is set to 0). For highest security it is recommended to use the highest TLS version possible.

Value:	Integer
Indexing:	CV(1) Error logging (0 = No error logging) (1 = Error logging to Notify window) (2 = All SCA printouts to Notify window) CV(2) Self-signed certificate generation (0 = Never) (1 = If not found) (2 = Always) CV(3) Operation when the remote certification validation fails (0 = Ignore, continue communication) (1 = Close connection) (2 = Not used) (3 = Close connection, no IP-address checking) CV(4) Type of the certificate store (0 = .pem file) CV(5) Minimum TLS version (31 = TLS 1.0/SSL 3.1) (32 = TLS 1.1/SSL 3.2) (33 = TLS 1.2/SSL 3.3) CV(6) Maximum TLS version (31 = TLS 1.0/SSL 3.1) (32 = TLS 1.1/SSL 3.2) (33 = TLS 1.2/SSL 3.3)

Table continues on next page

CV	Certificate Handling Vector
Default values:	CV(1) : 0 CV(2) : 0 CV(3) : 1 CV(4) : 0 CV(5) : 33 CV(6) : 33
Access:	Read/conditional write

For the configured TLS versions to become valid, they must be set before the station and the line are taken into use for the first time. If the TLS versions are changed afterwards, the PC-NET must be restarted for the changes to become valid.

The minimum TLS version must be equal or lower than the maximum TLS version.

```
#SET STA1:SCV1=1 ; enable error logging to notify window
#SET STA1:SCV5=31 ; Set minimum TLS version to 1.0
Example:
#SET STA1:SCV6=33 ; Set maximum TLS version to 1.2
```

CX	TLS Key renegotiation interval

The CX attribute defines the session renegotiation interval for TLS in seconds. Timer is triggered when the authentication level session keys for user "Common" are negotiated or when STA object is taken into use. If the CX value is 0, automatic triggering is not used but the same operation can still be done from SCIL application using attribute NU. If secure authentication is configured, the value of the attribute should be slightly less than the configured authentication level session key level interval defined in master (default in DNP3 standard is 15 minutes).

Data Type:	Integer
Value:	0..86400
Indexing:	No indexing
Default value:	830 (14,5 minutes)
Access:	No limitations

EE	Encryption Error

The EE attribute is used to return the last error code occurred in encryption. The value is the same for all STA objects that have the same remote IP-address in attribute IA. See Status Codes manual for error descriptions, chapter CSA SCA error codes. Attribute is used only in TCP mode.

Value:	Integer
Indexing:	No indexing
Default value:	0
Access:	Read-only

4.3.2.7 Autodialing attributes

SYS600 provides support for the Autocaller functionality of the DNP 3.0 Slave protocol. Autocaller is a modem with functions for automatic dial-up. The dial-up can be initiated by the DNP master or the DNP slave.

The Autocaller must use the AT (Hayes) command set. Note that when using odd or even parity, the modem must support 11-bit word length. In some cases, this feature must be enabled by using the AT commands. Refer to the documentation of the modem in use for further details.

The following Autocaller attributes are valid for the DNP 3.0 Slave lines:

AC	Autocaller Enabled

The AC attribute states whether an Autocaller is connected to the line (value 1) or not (value 0).

Data type:	Integer
Value:	0 or 1
Default value:	0
Access:	No limitations

AS	Autocaller State

This attribute indicates the state of the Autocaller.

Data type:	Integer
Value:	0...4 0 = IDLE, ready to make a call 1 = CONNECTED, transmission is activated 2 = BUSY, Autocaller is dialing 3 = INITIAL, Autocaller is uninitialized 4 = CONFIGURE, the IU attribute of the line is set to 0
Default value:	0
Access:	Read-only

CL	Connection Time Limited

This attribute determines whether a time limit has been set to the connection (value 1) or not (value 0). The maximum duration of the connection is determined by the CT attribute.

Data type:	Integer
Value:	0 or 1
Default value:	1
Suggested value:	A time limit is necessary on certain radio telephone lines. Limiting the connection time may also be good practice in other cases, if there is a risk that the connection is not otherwise broken.
Access:	No limitations

CT	Connection Time

The maximum time that a connection is allowed to last (in seconds). This attribute is significant only if time limiting is activated (CL = 1).

Data type:	Integer
Value:	0...600
Unit:	Seconds
Default value:	120
Access:	No limitations

CN	Connection

The CN attribute is used for dialing devices from the NET and for breaking telephone connections.

A call to a station or workplace is initiated by writing the phone number to the CN attribute. The NET unit then commands the autodialing modem to dial the number. The success of the dialing is reported as a system message. Writing an empty string to CN breaks the connection.

Data type:	Text
Value:	Text string of maximum 25 characters
Default value:	Empty text string

Table continues on next page

CN	Connection
Access:	No limitations

Example:
```
#SET NET1:SCN5 = "123456789S11"
```

CS	Connected Station

The link address of the station a NET unit is communicating with.

Data type:	Integer
Value:	0...65535 0 = Autocaller not defined or no communication
Default value:	0
Access:	Read-only

DD	Radio Disconnection Delay

Delay between the last data transfer and the line disconnection.

Data type:	Integer
Value:	0...32767
Unit:	Seconds
Default value:	0
Access:	No limitations

MC	Modem Command

Using this attribute, a modem can be controlled directly from SCIL with the AT/Hayes commands. When an AT command is written to the MC attribute, it is transmitted to the modem on the line. The response from the modem is read using the same attribute.

Data type:	Text
Value:	Text string, an AT/Hayes command
Default value:	0
Access:	No limitations

Example:
```
#SET NET1:SMC3 = ("AS0?")'
```

PU	Pulse Dialing

This attribute determines the dialing principle used.

Data type:	Integer
Value:	0 = tone dialing 1 = pulse dialing
Default value:	0
Access:	No limitations

RC **Remote Calls Enabled**

The RC attribute states whether remote calls are enabled on a line, meaning that the NET unit can be called from the stations connected to the line in question. This attribute applies to lines with autocaller (AC = 1).

Data type:	Integer
Value:	0 = remote calls not enabled 1 = remote calls enabled
Default value:	0
Access:	No limitations

RW **Radio Connection Wait Time**

Normally, the DCD (Data Carrier Detect) signal is used to indicate an active connection. However, there are cases where this is not possible, for example, on radiotelephone lines using half-duplex links. The RW attribute defines the waiting time in seconds in such a situation from the finishing of the dialing until the transmission is started.

Data type:	Integer
Value:	0...32767
Unit:	Seconds
Default value:	0
Access:	No limitations

SR **Autocaller AT S Register**

The S registers used by the Autocallers follow the AT (Hayes) de facto standard.

All the Autocallers that use the AT command set have a number of S registers. The number of registers used and the meaning of the individual registers varies slightly from one Autocaller model to another. The contents of the S registers are therefore not described in this document. Refer to the modem manuals.

Using the SR attribute, the S register number 2, 6, 7, 8, 9, 10, 11 and 12 are accessed. By using the MC attribute (see above), other S registers can also be accessed. The S registers 11 and 12 cannot be set.

Data type:	Integer
Value:	See the modem manuals
Indexing:	Seconds
Access:	100 * line number + register number

Example:
The S register number 6 of line 2 in NET1 is set = 4:

```
#SET NET1:SSR206 = 4
```

4.4 After configuration

For each input signal received from the process devices the process database should contain a process object whose value changes when process data is received. The change activates an event channel, which in turn starts a command procedure. The command procedure changes a value in the NET unit. From the NET unit, data can be transferred to the DNP master in two ways:

- As a response to a request (poll) from the master
- As an unsolicited (spontaneous) message

Which type of data transfer is used depends on the properties of the DNP 3.0 master and the attribute used when data is written to the NET unit. This is described in detail in the next chapter.

Besides the configuration of the base system and the communication system the following are also needed:

1. Configure the DNP master.
2. Configure the base system for process communication.
3. Configure the process units.
4. Define the cross-references for signal rerouting, if COM500*i* is used. For more information, see SYS600 COM500*i* User's Guide.
5. Create and define the input and output process objects for the process communication. This is usually done when creating the station picture by using standard functions from an application library.
6. Define event channels for the process objects.
7. Define command procedures for the event channels. If COM500*i* is used, the command procedures that are already in it can be used. Otherwise, the command procedures are programmed. For more information on how to program the command procedures and the values of the attributes, see Section 5.

4.5 How to test the configuration

When the slave and master stations have been physically tested and the configuration has been completed, the connection and configuration can be tested based on the following methods:

- For serial lines, Clear to Send (CTS) and Data Carrier Detect (DCD) signals. With the DNP 3.0 Slave protocols, both of these signals should be active in the slave end of the line as follows:
- When collision detection is not in use (the LK attribute of the line is 15), the Carrier Detect signal is always active and the Clear to Send signal is active when the master station is transmitting.
- When collision detection is in use (the LK attribute of the line is 14), the Clear to Send signal is active when the master station is transmitting, and the master station transmits only when the Carrier Detect signal is inactive, which means that no other station on the line is transmitting.
  The Data Terminal Ready (DTR) signal is kept in signaled state when the line in use. This signal can be used to control the fallback switches in a HSB configuration
- Diagnostic counters. When the communication between the slave and master is running properly and data is moving on the line, the diagnostic counters indicating the number of received and transmitted data messages should be incrementing.
- Object status. The OS attribute of the DNP slave station should be 0.
- By connecting a serial line analyzer to the DNP line (serial mode) or listening to the LAN using a network analyzer software (TCP and UDP modes).

For MicroSCADA version 9.3 and newer, the protocol analyzer included in PC-NET can be used. See the NET line attributes AO and AU in the SYS600 System Objects manual.

4.6 Serial cable wiring diagram

When connecting a remote partner to a DNP line using a direct serial cable, the wiring is illustrated in Figure 10. If the used port is a virtual COM port or easier cabling is needed, the line attribute CM provides the possibility to simulate DCD and CTS changes. The line attribute CM is available in SYS600 version 9.3 FP2 and newer.

Figure 11 should be used if collision detection is used (the LK attribute of the line is 14).

When connecting several remote devices to one line when collision detection is used, a more sophisticated wiring is needed. The connections should be made in a way that when the Request to Send signal of any slave device is set as an indication that the station is transmitting, the Carrier Detect signal in all other slave IEDs are set.

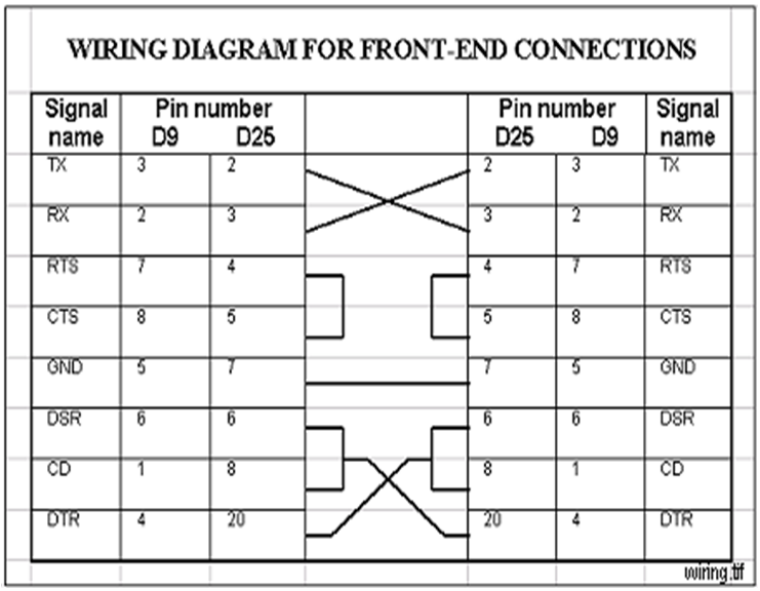


Figure 10: Serial cable wiring diagram when collision detection is not used

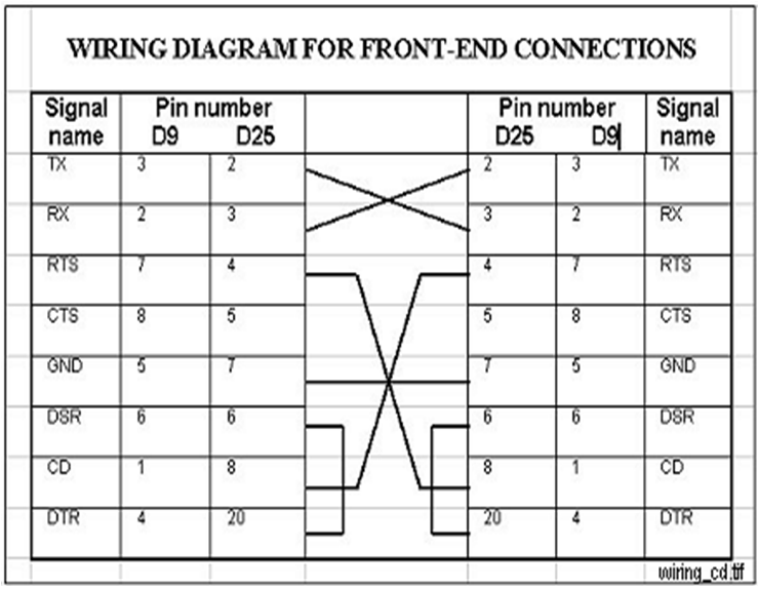


Figure 11: Serial cable wiring diagram when collision detection is used

4.7 Communication adjustment guidelines

MicroSCADA has multiple adjustable attributes that affect the communication. The default values might not be suitable for each system and can sometimes cause communication problems. Here are listed the steps of what should be done when the communication is not working correctly. Note that these advices are for serial line only.

4.7.1 Basic checks

In case of a communication does not start properly, following checks are worth to do before any further investigation:

- Check that the serial port defined with line attribute SD matches with the computes serial port. Check also that no error is reported to the notify window when the system or PC_NET is started.
- Check that the stations addresses match in master and slave.
- Check that the baud rates and other Data Transmission Attributes (in MicroSCADA System Configuration Tool) match in master and slave.

4.7.2 Message sending

If the remote end does not receive the messages sent from the line in question, following hints may help:

- Check if CTS errors exist. The DE attribute sets how long time the CTS signal is waited after the RTS signal has been raised. If the rising edge of the CTS does not occur at all (line disconnected) or if the DE attribute value is too small, timeout occurs, and the line enters to a CTS error state. The status point of the line is updated correspondingly
- Check the FIFO buffers of the used serial port. If RTS signal is used to control the data carrier, communication may function better if the FIFO buffers are disabled (or set to 1).
- Check the RTS Keepup Delay (RY) attribute value. If the RTS signal is used to control the data carrier and it is reset too early, the end of the message might be left out when the message is sent. With RY attribute the closing of the RTS signal can be delayed. On the other hand, if the RY delay is too long, the remote device may start transmission when the data carrier is not available. This causes an error.
- Check the COM Port Mode (CM) attribute. With the bit 3 setting the length of the message is used to calculate the RTS keep up time. See attribute documentation for more information.

4.7.3 Message reception

If the remote end send message but those are not received correctly by the line in question, following hints may help:

- Check the TW attribute value. The TW attribute delays the sending of the message after the RTS and CTS signals are raised. If the beginning of a message is missing when it is received (recorded) in the other end, increasing the TW attribute value might help.
- Check that the Header Timeout (HT) and Response Timeout (TI) attributes lengths are long enough for the communication. The TI time could be too short in cases when the response is long and the baud rate small.
- If the messages are sent and received correctly but are not processed, check CRC or other errors from the Diagnostic Counters in System Configuration Tools online-mode.
- The usage of the protocol analyzer and the bit 1 or line attribute AU (Analyzer usage) will help in the tuning of the attributes DE, HT, TI, RI and RY. If bit 2 of the attribute AU is set, internal information related to completion of the write operation to the serial driver is display in the analyzer output.

4.7.4 Data updating

If the remote end sends messages and it seems like those are received correctly by the line in question, but the process objects are not updating, following hints may help:

- Check that the attached process objects addresses are correct.
- Check that the STA object is allocated to the MicroSCADA application in which the process objects are located. Related STA object attributes are AL (Allocation) and AS (Allocating application).
- If the base system attribute TN (Translated Number) or ND (Node number) for the STA object has been modified by the SCIL application, check that those refer to correct STA object in correct PC_NET node.

Section 5 Technical description

5.1 DNP 3.0 Protocol

The Distributed Network Protocol (DNP) 3.0 is a standard-based communication protocol designed for electric utility, water, oil and gas, and security systems. DNP is hardware-independent and works with a variety of networks enabling communication between substation computers, Remote Terminal Units (RTUs), Intelligent Electronic Devices (IEDs) and master stations over serial or LAN-based systems. DNP3 is standardized as IEEE standard 1815.

DNP is designed according to the Enhanced Protocol Architecture (EPA) and it specifies the following Open Systems Interconnection (OSI) layers:

- Physical layer
- Data link layer
- Transport layer
- Application layer

The DNP 3.0 transport layer is a pseudo-transparent layer that provides minimum message assembly and disassembly. The purpose of the transport layer is to provide support for application messages larger than the frame length of the data link.

The physical layer can be any bit-serial physical layer, for example RS-232 C, RS-485, a fiber transceiver or Ethernet. In SYS600, the communication takes place using the serial port(s) of the base system computer. The interface used is RS-232 C.

A lot of the functionality of the protocol is implemented in SCIL by using the application and system objects as presented in this document. The reason for this is flexibility and versatility. Although different DNP masters are implemented according to the same standard, functionality and requirements still vary from one system and application to another. When some of the central parts of message handling are implemented in SCIL, changes can easily be made, even online.

5.2 Level of implementation

DNP 3.0 has three subset levels from 1 to 3, which each include a specific subset of DNP message types and functionality. In SYS600 the DNP 3.0 protocol has been implemented according to the Subset Level 2 of the protocol (DNP 3.00–L2) as presented in Table 11

The default operation mode of DNP V3.0 Slave in SYS600 is subset 2. Subset 3 is an extension to subset 2 and supported when the attribute SB is explicitly set to value 3. During the run-time operation, the SB is also raised to value 3 automatically if the master issues a command belonging to subset 3 only.

The main differences between subset 2 and subset 3 are:

- Extended range of objects and variations
- Dynamic assignment of points to report classes
- Unsolicited message enabling/disabling during runtime
- Simultaneous control operations of multiple points

The two device profiles are presented at the end of the manual. The first one is for the subset 2, which is the default operation mode, and the second one is for subset 3.

For more details about the implementation of the DNP 3.0 protocol in SYS600, see the device profile in the end of this document.

5.3 Supported process object types

Since in the DNP 3.0 Slave protocol input data is sent to the master by using SCIL statements, there is no strict relation between the SYS600 process object types and DNP data object types. However, the following relation can be made, see Table 1.

Table 1: Relation between the SYS600 process object types and DNP data object types

Object	Description	Process Object Type
1	Binary input	Binary input, double binary indication
3	Double bit input	Double bit input, double binary indication
10	Binary output status	Binary output
12	Control relay output block	Binary input
20	Binary counter	Pulse counter
30	Analog input	Analog input
40	Analog output status	Analog output
41	Analog output block	Analog input

Frozen binary counters are handled internally by the NET unit after the corresponding freeze command has been received.

5.4 Communication

This section gives a more detailed description of the implementation of the DNP 3.0 Slave protocol in SYS600, describing also the attributes that can be used for device communication. Examples of how to exchange data between the master and the slave are also given in this section along with information of the DNP 3.0 Slave status codes.

5.4.1 Communication modes

A communication mode describes how messages are sent between the DNP 3.0 master and the slave by polling, by means of unsolicited messages, or by both methods. The following four cases can be identified:

- Quiescent Operation. In this mode the master does not poll the slave, all the communication is based on unsolicited (spontaneous) report-by-exception messages. The master can send application layer confirmations to the slave.
- Unsolicited Report-by-Exception Operation. The communication is basically unsolicited but the master occasionally sends integrity polls for Class 0 data to verify that its database is up-to-date.
- Polled Report-by-Exception Operation. The master frequently polls for event data and occasionally for Class 0 data.
- Static Report-by-Exception Operation. The master polls only for Class 0 data or the specific data it requires.

When configuring DNP 3.0 stations in SYS600, the communication mode is set when the NET database is initialized by using the SD attribute. The communication mode must be selected to match the behavior of the master. The most important issue is whether or not the master accepts unsolicited messages.

5.4.2 Protocol converter

Each DNP 3.0 Slave station configured on a line of a NET unit acts as a protocol converter between the DNP 3.0 protocol and a base system. An internal protocol of SYS600 is used in communication between the SYS600 nodes, for example, between a base system and a NET unit.

In DNP, there are basically two kinds of data: static data and event data. Static data in DNP is called class 0 data. Event data can have three different classes or priorities: 1 (high priority), 2 (medium priority) and 3 (low priority). In DNP terminology static data objects are called by their data type. For example, binary input and analog input. Corresponding events are called by adding "change" to the static data object. For example, "binary input change" and "analog change".

Data, both static and event data can be sent from the slave to the master in two ways: either the master polls it cyclically or when needed, or it is sent to the master spontaneously as unsolicited messages. A combination of these two ways is also possible. From this point of view DNP 3.0 implements both the balanced and unbalanced transmission procedures.

Each DNP 3.0 slave station has a database in the NET unit. This database contains a static storage for the static data, and three queues for the event data, one for each class. When data is sent from the slave to the master, the message is written to the NET database by using the SD attribute (static data) or the EV attribute (event data). These attributes are described later in this document. When data is sent from the master to the slave, it is forwarded directly to the SYS600 process database without going through the NET database.

5.4.3 Handshaking

By default, NET sends spontaneous "test function of link" messages to the DNP 3.0 line in order to check that the communication link is open. The message is sent at intervals specified by the PD attribute of the line. If the DNP 3.0 master does not acknowledge this message, the communication is marked as broken.

If the DNP 3.0 master does not accept spontaneous "test function of link" messages from the slave, sending these messages can be prevented by setting the PD attribute of the line to zero or by setting the line attribute OM, bit 2. In this case, the communication link is not supervised cyclically. With bit 3 of the line attribute OM, it is possible to disable the initial handshaking with the message "Reset of remote link". Disabling "Test function for link" and "Reset of remote link" messages may be needed with masters which do not accept spontaneous messages from the slave devices.

5.4.4 DNP 3.0 in LAN/WAN

The DNP V3.0/LAN protocol is used in LAN and WAN networks to connect central stations and outstations to each other. Since the stations use an open TCP/IP interface as a connection to the network, the structure and characteristics of the network used are invisible to the application. The transferred data messages are equal to the ones used in the serial line based on the DNP 3.0 protocol. When operating as a DNP 3.0 slave, SYS600 is the server-end of the connection.

When operating as a controlled station (slave), only one connection to the controlling station can be open at the same time. Each connection has a configurable and unique IP address. There can only be one DNP 3.0 slave line operating in the TCP or UDP mode in the same IP address defined with the line attribute LD.

The connection type is defined with the line attribute SD. The client internet address is defined with the station attribute IA. Configuration examples for the LAN connection are given in the Appendix.

5.4.5 Addressing

In the DNP 3.0 protocol each link layer message has two addresses: SOURCE address indicates from which station the message is coming and DESTINATION address tells to which station the message is going. When configuring a DNP 3.0 Slave station in SYS600, the SA (Slave Address) attribute should be equal to the SOURCE address and the MA (Master Address) equal to the DESTINATION address.

In the DNP 3.0 terminology a data point address is called an index. Several data points of the different data object type, for example, binary input and analog input, can have the same index. An individual data point is identified with the combination of data object type and index.

5.4.6 Internal indications

Each response message sent by the slave contains a two-octet field called internal indications, which indicates the current state of the slave device. This information can be used to monitor the state of the slave or for troubleshooting purposes.

The internal indications are set automatically by the DNP slave station object but they can also be controlled by the SCIL application by using the station attributes IN and CF.

The bits of the internal indications are as presented in Table 2.

Table 2: Bits of the internal indications

Octet	Bit	Description
1	0	All stations message received
1	1	Class 1 data available
1	2	Class 2 data available
1	3	Class 3 data available
1	4	Time-synchronization required from the master
1	5	Some or all of the Outstation's digital output points are in local state
1	6	Device trouble
1	7	Device restart
2	0	Function code not implemented
2	1	Requested object(s) unknown
2	2	Parameters in the qualifier, range or data fields are not valid or out of range
2	3	Event buffer(s), or other application buffers overflow
2	4	Request understood but the requested operation is already executing
2	5	The current configuration in the slave is corrupt
2	6	Currently always returned as zero (0)
2	7	Currently always returned as zero (0)

For example, if the value of the IN attribute is 144 = 0000000010010000, it means that the slave should be synchronized (octet 1, bit 4) and the slave device has restarted (octet 1, bit 7).

5.4.7 Data flow

Figure 12 describes the data flow between the process devices and the DNP master. Both directions are described separately since the data is handled in a different way depending on the direction.

See manual COM500*i* User's guide for detailed information about the data flow configuration.

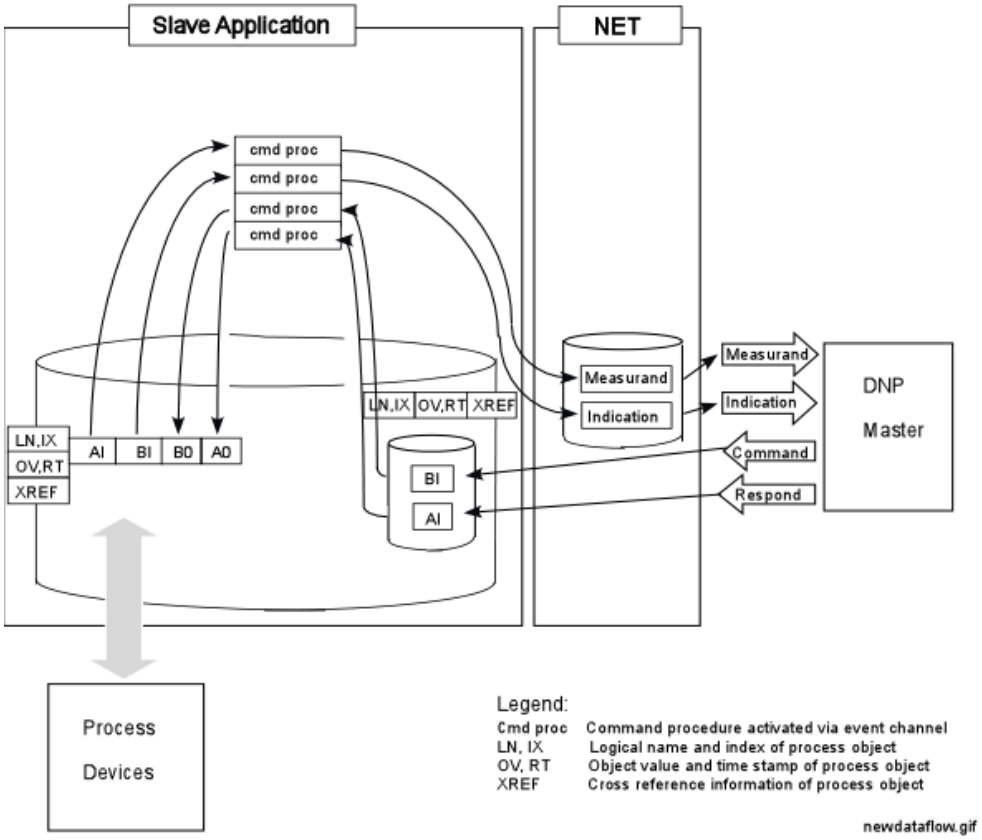


Figure 12: Data flow between the process devices and the DNP master in COM500i

5.4.7.1 Input data

When input data, for example indications and measured values, are sent from the process devices to the DNP master, the following steps are taken:

1. The process devices send data to the SYS600 process database.
2. The updated process object activates an event channel.
3. The event channel executes a command procedure. Some of the attributes of the process object are given as arguments to the command procedure.
4. The command procedure sends the data to the NET database by using the SD and EV attributes based on specific cross-reference information.
5. The NET unit sends data to the DNP master either as static or unsolicited data.

The cross-reference information is the data needed to deliver the data to the NET database, for example object address and message type.

The number event channels and command procedures needed to deliver data to the master depends on the application. One solution is to have one command procedure for each process object type. Examples of the command procedures are given later in this document.

In the quiescent operation mode static data is sent to the master as unsolicited messages only at start-up, and when the status of a data point changes. Besides these situations, data must be sent as change events or by means of polling.

5.4.7.2 Output data

When output data, for example object commands and analog setpoints, are sent from the DNP master to the process devices, the following steps are taken:

1. The command is received by the SYS600 process database. There must be a separate input process object for each DNP command address. This part of the process object is created manually when the signal engineering is done.
2. The updated process object activates an event channel.
3. The event channel executes a command procedure. Some of the attributes of the process object are given as arguments to the command procedure.
4. The command procedure sends the process devices by setting the corresponding output process object(s) and, if required, sends a confirmation to the DNP master via the NET unit.

Cross-reference data can also be used with commands. It can contain, for example, information about the logical names and indices of the output process objects. Examples of the command procedures are given later in this document.

5.4.7.3 Other messages

In addition to the input and output data described above, there are also some other messages transferred between the master and the slave. These messages contain, for example, synchronization commands and restart commands. Some of these commands are handled automatically by the NET unit while others require SCIL programming. The ones that require SCIL programming are described in detail later in this document.

System and application commands and DNP 3.0 messages that are not recognized by the slave station are forwarded to a bit stream process object with an address as defined by the CA address of the slave station. By using this process object the messages can be interpreted and processed by SCIL, if necessary.

5.4.8 Device communication attributes

SD	**Spontaneous Data**
Writing data to the SD attribute updates the corresponding static data object (class 0) of the NET database. It does not create an event even if the value changes; events must be generated by using the EV attribute. The second purpose of the SD attribute is to construct the NET database. When a data object is updated while the DI attribute is 0, the object is added to the database if it is not previously known. There must be a data object for each input DNP 3.0 address.	
Data type:	Vector
Value:	See above
Index range:	0...3, see above
Access:	No limitations
Indexing:	
While writing:	0…3. The index determines which class is used as the default class, when an unsolicited message is created for this address using the EV attribute as follows: 0: the event generation is disabled for this object 1: default class is 1 2: default class is 2 3: default class is 3
While reading:	(TYPECODE..IOA+ TYPECODE)
Description of the vector parameters:	
TYPECODE:	A code for the data object type as presented in Table 3.
IOA:	Object address, see the description below.

Table 3: Codes for data object types

Code	Description
0	Binary Input
1	Binary Output
2	Binary Counter
3	Analog Input
4	Analog Output
5	Double binary input

Value:

When writing: vector (OBJ, VAR, IOA, VAL, FLAG, [CTRL])

When reading: vector (EV_CLASS, TYPE, VAR, IOA, VAL, FLAG, CTRL)

Description of the vector parameters:

OBJ: Data object type

VAR: Object variation, depends on the data object type

In a DNP 3.0, message types are identified with the combination of data object type and variation. One data object usually has several variations, for example binary input without status (data object 1, variation 1) and binary input with status (data object 1, variation 2).

When the static data object in the NET database is updated, the parameters OBJ and VAR define the default variation used in responses to the master, that is, the variation used if no variation is specified in the request from the master. Data object types and variations presented in Table 4 are possible.

Table 4: Message types are identified with the combination of data object type and variation

Object	Description	Variations
1	Binary Input	1, 2
3	Double Binary Input	1, 2
10	Binary Output	2
20	Binary Counter	1, 2, 3 4, 5, 6, 7, 8
30	Analog Input	1, 2, 3, 4, 5
40	Analog Output	1, 2, 3

IOA: Object address

Value: 0...255,
when IL attribute = 1
0...65535
when IL attribute = 2

VAL: The value of the object. The value range depends on the data object type. See the examples later in this document. If the object is of binary type the values 10…13 are converted to double indications as follows: 10 -> 0, 11 -> 1, 12 -> 2, 13 -> 3.

FLAG: Integer, bit mask of the status bits of object. Status bits indicate the current state of the data object. The bits in the flag byte are different for different data object types (see the examples later in this document). The relation between the DNP 3.0 status bits and process object attributes is as shown in Table 5.

Table 5: Status bits indicate the current state of the data object

Bit	Description	
On-line	The on-line bit indicates that the binary input point has been read successfully. If this field is set to off-line, the state of the digital point might not be correct.	OS
Restart	The restart bit indicates that the field device which originated the data object is currently restarting. This can be the device reporting this data object.	-
Communication lost	The communication lost bit indicates that the device reporting this data object has lost communication with the originator of the data object.	OS
Remote forced data	The remote forced data bit indicates that the state of the binary input has been forced to its current state at a device other than the end device.	-
Local forced data	The local forced data bit indicates that the state of the binary input has been forced to its current state at the end device.	SB
Chatter filter	The chatter filter bit indicates that the binary input point has been filtered in order to remove unneeded transitions in the state of the point.	BL
Roll-over	When set, the roll-over bit indicates that the accumulated value has exceeded the last reported recordable. The counter value has been reset to 0 upon the roll-over and counting has resumed as normal. This bit is cleared when the counter value (plus the roll-over state) is reported.	-
Over range	The out of range field indicates that the digitized signal or calculation has exceeded the value range.	OR

CTRL: Integer, a bit mask of the control bits of the data object. The control bits define in which conditions the data object is reported to the master, that is sent as a response to a request. The control bits also define whether an unsolicited message is generated or not. The data bits are as follows:

Bit 0: 0 = object is not reported to the master at all
1 = object is reported to the master

Bit 1: 0 = unsolicited messages of this object are not generated
1 = object is reported as an unsolicited message

Bit 2: 0 = the static object is not sent as an unsolicited message
1 = the static object is sent as an unsolicited message at start up and when the status of the object changes

CTRL is optional when the value of the DI attribute of the station is 1. For the different communication modes the value of CTRL can be as presented in Table 6.

Table 6: The different communication modes and corresponding control bits

Communication Mode	CTRL
Quiescent Operation	7
Unsolicited Report-by-Exception Operation	3
Polled Report-by-Exception Operation	1
Polled Static Operation	1

EV **Event Data with Time Stamp**

By setting the EV attribute, the user can send time-tagged events to the DNP 3.0 master. An event is generated if:

- The value of the data object changes. In the case of an analog data object, the change must be greater than the DELTA parameter.
- The status (FLAG parameter) of the data object changes.

The EV attribute updates also the static data object in the NET database. Whether an unsolicited message is generated or not depends on the value of the CTRL element used when the data point was initialized by using the SD attribute. By reading the EV attribute the user can enquire the number of items in the event queues. If index=0 is given when reading, total amount of events in queues is returned. If index is 1..3, it specifies the class and the amount of queued events in that class is returned.

Data type:	Vector
Value:	See above
Index range:	0...3, see above
Access:	No limitations

Indexing:

0...3:	The unsolicited message class as follows:
0 or not given:	The default class determined with the SD attribute is used
1...3:	Class 1, 2 or 3

Value:

When writing:	Vector (TYPE, VAR, IOA,VAL, FLAG, DELTA, RT, RM, [EVCTRL])
When reading:	Integer, the amount of events Index 0 : total amount of events Index 1..3 : amount of events in class specified with index

Description of the vector parameters:

OBJ:	Data object type, integer
VAR:	Object variation, depends on the data object type, integer

In DNP 3.0, message types are identified with the combination of data object type and variation. A data object usually has several variations, for example binary input change without time (data object 2, variation 1) and binary input change with time (data object 2, variation 2).

The following data object types and variations showed in Table 7 are possible.

Table 7: The message types are identified with the combination of data object type and variation

Object	Description	Variations
2	Binary Input Change	1,2,3
4	Double Binary Input Change	1, 2, 3
22	Binary Counter Change Event	1,2,3,4,5,6
32	Analog Change Event	1,2,3,4,5,7

IOA: Object address, integer

Value: 0...255,
when IL attribute = 1
0...65535,
when IL attribute =2

VAL: The value of the object. The value range depends on the data object type. See the examples later in this document.

FLAG: Integer, bit mask of the status bits of object. Status bits indicate the current state of the data object. The bits in the flag byte are different for different data object types. See the examples later in this document.

DELTA: Integer. An event is generated only if the change of value is greater than the delta value. This parameter can be used to establish a deadband handling in the NET database.

RT: Registration time. The registration time included in the time stamp of the unsolicited message.

RM: Registration milliseconds, integer 0..999. The millisecond part of the time stamp of the unsolicited message.

EVCTRL This is an optional field, which can be given to define the contents of the information more specifically. The value is a bit pattern. Bit 0 controls the generation of the event and bit 1 controls the conversion of the timestamp to UTC time.
Bit 0 = 0: An event is generated only if the change of value is greater than the delta value or its status changes
Bit 0 = 1: An event is always generated, even if the value does not change
Bit 1 = 0: No conversion from local time to UTC (Coordinated Universal Time) is requested. The given timestamp is sent as such.
Bit 1 = 1: The timestamp is given as local time and the conversion to the UTC time is requested. The timestamp of the event message is converted to the UTC time as defined in the DNP 3.0 standard.

CF Command Confirmation

The CF attribute is used to accept commands received by the NET unit. The NET unit does not send a response message to the command before the command is confirmed using the CF attribute. The NET automatically stores the received commands to a table. When the parameter ADDR is given when writing data to the CF attribute, the confirmed command is searched (latest first) from the table. If no command with a matching address is found, the error 13925 = DNPC_NO_ACTIVE_COMMAND is returned and a response message is not sent.
If the optional IIN_W is given, a bitwise OR operation is done with the current IIN of the station object and the result value is sent as a part of the response message. The least significant byte of IIN_W refers to the first byte of the internal indication word and the most significant byte of IIN_W refers to the second byte of the internal indication word.

Data type: Vector

Value: Vector (CS, OBJ_ADDR, [IIN_W])

Access: Write-only

Description of the vector parameters:

CS: Control status, integer 0.255. Control status indicates the status of the command to the master.

Valid status codes (values are the same as in DNP standard):

0: Request accepted or initiated

1: Request not accepted, the time between select and operate was too long

2: No previous select message

3: Request not accepted because of a formatting error

4: Control operation is not supported for this point

5: Request not accepted because the point is already active

6: Request not accepted because of control hardware problems

7: Request because Local/Remote switch is in Local position

8: Request not accepted because too many object appeared in the same request

9: Request not accepted because of insufficient authorization

10: Request not accepted because it was prevented or inhibited by a local automation process

11: Request not accepted because the device cannot process any more activities than are presently in progress

12: Request not accepted because the value is outside the acceptable range permitted for this point

13-125: Undefined

By adding 128 to the values above, the slave reports that some of the binary outputs are in local state.

OBJ_ADDR: Object address of the confirmed command, integer. In most cases this is the address of the process object receiving the command.

Value: 0...255,
when IL attribute = 1
0...65535,
when IL attribute =2

IIN_W Optional parameter to provide extra information in the IIN word of the next response

Value: 0..65535

Examples:

if it is required that the bit IIN2.2—Parameter error need to be set to the response message of a control command, the SCIL statement could be

```
#SET STA'UN':SCF=vector(4, 'OA', 256*4+0)
```

DI Database Initialized

When this attribute is set to 0, the NET database can be initialized by using the SD attribute. When the value of this attribute is returned to 1 after the initialization, the communication between the master and the slave can be started. When this attribute is set to 2, the NET database is cleared and it can be initialized by using the SD attribute. While DI = 0, the slave does not send data as response to requests from the master.

Data type: Integer

Value: 0, 1 or 2

Access: No limitations

RS	Reset Event Queues
The RS attribute can be used for clearing the content of one or several event queues.	
Data type:	Integer
Value:	0...3 0 = Events cleared from classes 1,2 and 3 1 = Events cleared from class 1 2 = Events cleared from class 2 3 = Events cleared from class 3
Access:	Write-only

5.5 Command procedures

5.5.1 Command procedures in COM500*i*

Signals are sent from the process units to an NCC and commands from an NCC to the process unit. COM500*i* reroutes the signals using command procedures and cross-references, see Figure 13. The command procedures, which are used for signal rerouting, are described here.

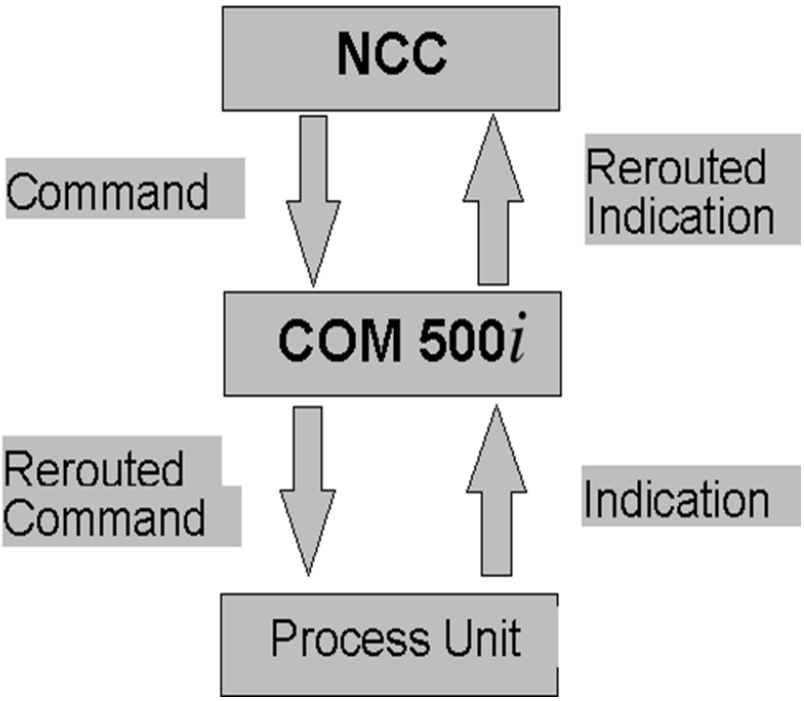


Figure 13: COM500i reroutes the signals

If the DNP 3.0 Slave protocol is used with COM500*i*, the command procedures available for COM500*i* are used. The cross-reference information is entered in the Signal Cross-reference tool. For more information, see SYS600 COM500*i* User's Guide. Table 8 shows the used COM500*i* command procedures and event channels.

Table 8: Used event channels and command procedures

Process Object Type	Event Channel	Command Procedure
Analog Input	COM_USAI	COM_USAI
Single Indication	COM_USDI	COM_USDI
Double Indication	COM_USDB	COM_USDB
Table continues on next page		

Process Object Type	Event Channel	Command Procedure
Pulse Counter	COM_USPC	COM_USPC
Binary Outputs	COM_DSBO COM_USBO*	COM_DSBO COM_USBO*
Analog Outputs	COM_DSAO COM_USAO*	COM_DSAO COM_USAO*

* = Used for reporting the status of the output object to the master.

5.5.2 Command procedures in SYS600

5.5.2.1 Command procedures for process data

The command procedures presented in this chapter cannot cover all combinations and requirements, but they provide examples of the implementation.

The connection between the SYS600 process objects and messages to and from the DNP master is made by using cross-reference data. The cross-reference data is written to the FX (Free teXt) attribute of the process objects by using the Process Object Definition Tool.

The cross-references for data transfer from SYS600 to the DNP master are kept in the FX attribute, which is a string of max 30 characters (10 before revision 8.4.2). The general syntax for a cross-reference is:

```
<STA_NUMBER><ADDRESS><ST_TYPE><ST_VAR><DEF_CLASS>
   <EV_TYPE>][<EV_VAR>][<EV_CLASS>][<EV_DELTA>]
```

STA_NUMBER: Logical STA number (range 1...255). The value is stored as 3 ASCII digits with leading spaces added if necessary. The FX string positions 1...3 are used for this value.

ADDRESS: The address in the database of the DNP 3.0 master. It is stored as 5 ASCII digits with leading spaces added if necessary. The address uses FX string positions 4 ... 8.

ST_TYPE: The data object type of the static data object in the NET database. It is stored as 2 ASCII digits with leading spaces added if necessary. The static data object type uses FX string positions 9...10.

ST_VAR: The data object type of the static data object in the NET database. It is stored as 2 ASCII digits with leading spaces added if necessary. The static variation uses FX string positions 11...12.

DEF_CLASS: The default class used when a change of the process object generates an event. It must be set to 0, if no event is generated. It is stored as one ASCII digit. The static variation uses FX string position 13.

EV_TYPE: The data object type used when a change of the process object generates an event. It can be omitted if no event is generated. It is stored as 2 ASCII digits with leading spaces added if necessary. The event data object type uses FX string positions 14...15.

EV_VAR: The variation used when a change of the process object generates an event. It is stored as 2 ASCII digits with leading spaces added if necessary. The event variation uses FX string positions 16...17.

EV_CLASS: The event class used when a change of the process object generates an event. If it is set to 0, the default class is used. It is stored as one ASCII digit. The event variation uses FX string position 18.

EV_DELTA: The delta value used when an analog event is generated. The change of the value must be greater than the delta value in order to generate an event. It is stored as 12 ASCII digits with leading spaces added if necessary. The event delta uses FX string positions 19...30.

The used STA object attribute depends on whether an event is generated (EV) or not (SD). The value of the FLAG parameter is calculated from the attributes of the process object.

An example of the cross-reference of a binary input signal:

FX			1		7	5	0	0		1		2	1		2		3	0												0
CHAR	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0

Another example of the cross-reference of an analog input signal, which does not generate an event:

FX			1		5	2	3	0	3	0		2	0																
CHAR	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0

The NET database initialization is described in a separate example. Thus, the CTRL parameter is omitted in the following examples when updating the static data object in the NET database by using the SD attribute.

Analog inputs
If a 16-bit variation is used, analog values must be limited or scaled to the values –32768...32767. This can be done in SCIL or by using a specific scale object and the SCALE function.

All the analog values are connected to the same event channel/command procedure combination. The command procedure is activated each time the process object is updated. It reads the updated value, and if necessary, limits or scales it and sends it to NET by using the cross-reference data. An example of the command procedure is listed below:

```
;read cross-reference data
;common data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,3))
@ST_VAR = DEC_SCAN(SUBSTR(%FX,11,2))
@DEF_CLASS = DEC_SCAN(SUBSTR(%FX,13,1))
;data for unsolicited messages
#IF %DEF_CLASS <> 0 #THEN #BLOCK
  @UN_TYPE = DEC_SCAN(SUBSTR(%FX,14,2))
  @UN_VAR = DEC_SCAN(SUBSTR(%FX,16,2))
  @UN_CLASS = DEC_SCAN(SUBSTR(%FX,18,1))
  @UN_DELTA = DEC_SCAN(SUBSTR(%FX,19,0))
#BLOCK_END

;if 16-bit value (even variation), limit value to U16
#IF (EVEN(%ST_VAR) OR EVEN(%ST_VAR)) AND ABS(%OV) > 32767 - #THEN #BLOCK
  @VALUE = ROUND(MIN(MAX(-32767,%OV),32767)
  @OR = 1
#BLOCK_END
#ELSE @VALUE = ROUND(%OV)

;compose the flag byte
#IF %OS == 2 #THEN @COMM_LOST = 1
#ELSE @COMM_LOST = 0
#IF %OS == 1 OR %OS == 10 #THEN @ON_LINE = 0
#ELSE @ON_LINE = 1
@FLAG = %ON_LINE+%COMM_LOST*4+%SB*16+%OR*32

;send data to NET
#IF %DEF_CLASS == 0 #THEN - ;static data
#SET STA'STA_NR':SSD(%DEF_CLASS)=(%ST_TYPE,%ST_VAR,%ADDR,-%VALUE,%FLAG)
#ELSE - ;event
#SET STA'STA_NR':SEV(%UN_CLASS)=(%UN_TYPE,%UN_VAR,%ADDR,-
%VALUE,%FLAG,%UN_DELTA,%RT,%RM,0)
```

Single indications

All the binary inputs are connected to an event channel/command procedure combination that sends the process value to the NET unit. The command procedure is activated each time the process object is updated. It reads the updated value and sends it to NET by using cross-reference data. An example of the command procedure is listed below:

```
;read cross-reference data
;common data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,3))
@ST_VAR = DEC_SCAN(SUBSTR(%FX,11,2))
@DEF_CLASS = DEC_SCAN(SUBSTR(%FX,13,1))
;data for unsolicited messages
#IF %DEF_CLASS <> 0 #THEN #BLOCK
 @UN_TYPE = DEC_SCAN(SUBSTR(%FX,14,2))
 @UN_VAR = DEC_SCAN(SUBSTR(%FX,16,2))
 @UN_CLASS = DEC_SCAN(SUBSTR(%FX,18,1))
 @UN_DELTA = DEC_SCAN(SUBSTR(%FX,19,0))
#BLOCK_END

;compose the flag byte
#IF %OS == 2 #THEN @COMM_LOST = 1
#ELSE @COMM_LOST = 0
#IF %OS == 1 OR %OS == 10 #THEN @ON_LINE = 0
#ELSE @ON_LINE = 1
@FLAG = %ON_LINE+%COMM_LOST*4+%SB*16+%BL*32

;send data to NET
#IF %DEF_CLASS == 0 #THEN - ;static data
#SET STA'STA_NR':SSD(%DEF_CLASS)=(%ST_TYPE,%ST_VAR,%ADDR,-
ROUND(%OV),%FLAG)
#ELSE - ;event
#SET STA'STA_NR':SEV(%UN_CLASS)=(%UN_TYPE,%UN_VAR,%ADDR,-
ROUND(%OV),%FLAG,%UN_DELTA,%RT,%RM,0)
```

Double indications

Double indications are not supported directly in DNP 3.0. It is possible to send two-bit indications from SYS600 by adding 10 to the corresponding binary input value as follows: 10 -> 0, 11 -> 1, 12 -> 2, 13 -> 3. This reserves two binary input objects with consecutive addresses from the NET database. If an unsolicited message is generated, it generates two messages with consecutive addresses. Parameters other than values are as described in the previous example.

Double indications using double bit input object

All the binary inputs are connected to an event channel/command procedure combination that sends the process value to the NET unit. The command procedure is activated each time the process object is updated. It reads the updated value and sends it to NET by using cross-reference data. If the data point is configured to be sent using a double bit input object, the group number 3 (static data) or 4 (event) is selected.

Pulse counters

Pulse counters are termed binary counters in the DNP 3.0 protocol. Frozen counters (counters with frozen values) are handled internally by NET and cannot be written to NET by using SCIL. All the pulse counters are connected to an event channel / command procedure combination that sends the process value to the NET unit. The command procedure is activated each time the process object is updated. It reads the updated value and sends it to NET by using cross-reference data.

```
;read cross-reference data
;common data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,3))
@ST_VAR = DEC_SCAN(SUBSTR(%FX,11,2))
@DEF_CLASS = DEC_SCAN(SUBSTR(%FX,13,1))
```

```
;data for unsolicited messages
#IF %DEF_CLASS <> 0 #THEN #BLOCK
 @UN_TYPE = DEC_SCAN(SUBSTR(%FX,14,2))
 @UN_VAR = DEC_SCAN(SUBSTR(%FX,16,2))
 @UN_CLASS = DEC_SCAN(SUBSTR(%FX,18,1))
 @UN_DELTA = DEC_SCAN(SUBSTR(%FX,19,0))
#BLOCK_END

;compose the flag byte
#IF %OS == 2 #THEN @COMM_LOST = 1
#ELSE @COMM_LOST = 0
#IF %OS == 1 OR %OS == 10 #THEN @ON_LINE = 0
#ELSE @ON_LINE = 1
@FLAG = %ON_LINE+%COMM_LOST*2+%SB*8

;send data to NET
#IF %DEF_CLASS == 0 #THEN - ;static data
#SET STA'STA_NR':SSD(%DEF_CLASS)=(%ST_TYPE,%ST_VAR,%ADDR,-
ROUND(%OV),%FLAG)
#ELSE - ;event
#SET STA'STA_NR':SEV(%UN_CLASS)=(%UN_TYPE,%UN_VAR,%ADDR,-
ROUND(%OV),%FLAG,%UN_DELTA,%RT,%RM,0)
```

5.5.2.2 Command procedures for the status of output objects

In the DNP 3.0 protocol it is possible to report the status of the binary and analog output objects to the master. This means that output information is sent as if it was input data. Only static data objects are provided for the status of output objects, no unsolicited messages are available.

When reporting the status of multiple output objects (for example the four binary objects of a controllable SPA device) to the master, it is often enough to connect to one output object. Another way is to combine the status of all the outputs by using logical AND/OR operations.

All the output objects of the same type that are reported to the master are connected to an event channel / command procedure combination that sends the process value to the NET unit. The command procedure is activated each time the process object is updated. It reads the updated value and sends it to NET by using cross-reference data. In this case, the cross-reference information related to unsolicited messages can be left out.

```
;common data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,3))
@ST_VAR = DEC_SCAN(SUBSTR(%FX,11,2))
@DEF_CLASS = DEC_SCAN(SUBSTR(%FX,13,1))

;compose flag byte
#IF %OS == 2 #THEN @COMM_LOST = 1
#ELSE @COMM_LOST = 0
#IF %OS == 1 OR %OS == 10 #THEN @ON_LINE = 0
#ELSE @ON_LINE = 1
@FLAG = %ON_LINE+%COMM_LOST*4+%SB*16+%BL*32

;send data to NET
#SET STA'STA_NR':SSD(%DEF_CLASS)=(%ST_TYPE,%ST_VAR,%ADDR,-
ROUND(%OV),%FLAG)
```

5.5.2.3 Command procedures for initializing the NET database

All the input signals that are to be sent to the DNP 3.0 master must be written to NET while the DI attribute of the corresponding STA object is 0. This initializes the NET internal database by creating static data objects. The initialization must be made:

- At start-up. This should be done after the input process objects have been updated to avoid unnecessary sending of bad status.
- Whenever signals are added or removed, or any of the cross-reference parameters (address, data object type etc.) are changed.
- After the initialization is completed, the DI attribute must be set to 1 to establish communication to the master. While DI = 0, the slave does not respond with data to the requests from the master.

Listed below is an example of a command procedure that initializes the NET database by using the same cross-reference data as in the examples above. This command procedure can be executed, for example, from the predefined command procedure APL_INIT_1. When using this procedure, the value of the control byte (CTRL) must be changed to correspond to the master.

```
;initialise variables
@XREF_STA_NRS = VECTOR()

;find all process objects with cross-reference data
#LOOP
@XREF_OBJECTS = APPLICATION_OBJECT_LIST(0,"IX","A","F","",-
”IU==1 AND SS>0 AND FX<>””””””,(“FX”,"OS","PT","OV","LN",-
"IX","RT","RM","BL","SB","OR"),10000)
#IF LENGTH(XREF_OBJECTS:VLN) > 0 #THEN #BLOCK
  ;control byte (depends on communication mode)
  @CTRL = 3 ; unsolicited report-by-exception mode
  ;initialise static data objects in NET
  #LOOP_WITH SIGNAL = 1.. LENGTH(XREF_OBJECTS:VLN)
;cross-reference data
  @FX = XREF_OBJECTS:VFX(%SIGNAL)
  @STA_NR = DEC_SCAN(SUBSTR(%FX,1,3))
  @ST_VAR = DEC_SCAN(SUBSTR(%FX,11,2))
  @DEF_CLASS = DEC_SCAN(SUBSTR(%FX,13,1))
  ;set DI to zero, if not already set
  #IF LENGTH(SELECT(%XREF_STA_NRS,”==%STA_NR”))== 0 -
  #THEN #BLOCK
    @XREF_STA_NRS = APPEND(%XREF_STA_NRS,%STA_NR)
    #SET STA’STA_NR’:SDI = 0
  #BLOCK_END
  ; substitution value and flag for each data type
  @OS = XREF_OBJECTS:VOS(%SIGNAL)
  @BL = XREF_OBJECTS:VBL(%SIGNAL)
  @SB = XREF_OBJECTS:VSB(%SIGNAL)
  #IF %OS == 2 #THEN @COMM_LOST = 1
  #ELSE @COMM_LOST = 0
  #IF %OS == 1 OR %OS == 10 #THEN @ON_LINE = 0
  #ELSE @ON_LINE = 1
  #CASE XREF_OBJECTS:VPT(%SIGNAL)
    #WHEN 3 #BLOCK ; binary input
      @SUBS_VALUE = 1
      @FLAG = %ON_LINE+%COMM_LOST*4+%SB*16+%BL*32
    #BLOCK_END
    #WHEN 9 #BLOCK ;analog input
      @SUBS_VALUE = 0
      @FLAG = %ON_LINE+%COMM_LOST*4+%SB*16+%BL*32
    #BLOCK_END
    #WHEN 12 #BLOCK ;double binary input
      @SUBS_VALUE = 11
      @FLAG = %ON_LINE+%COMM_LOST*4+%SB*16+%BL*32
    #BLOCK_END
    #WHEN 13 #BLOCK ;pulse counter
      @SUBS_VALUE = 0
      @FLAG = %ON_LINE+%COMM_LOST*2+%SB*8
    #BLOCK_END
   #CASE_END
   ;if not sampled or error, give substitution value
   #IF %OS >= 10 #THEN @VALUE = %SUBS_VALUE
   #ELSE @VALUE = XREF_OBJECTS:VOV(%SIGNAL)
```

```
   #SET STA'STA_NR':SSD(%DEF_CLASS)= -
   (%ST_TYPE,%ST_VAR,%ADDR,%VALUE,%FLAG,%CTRL)
  #LOOP_END
 #BLOCK_END
 #IF NOT XREF_OBJECTS:VMORE #THEN #LOOP_EXIT
#LOOP_END

;set DI attribute to 1
#IF LENGTH(%XREF_STA_NRS) > 0 #THEN #BLOCK
 #LOOP_WITH I = 1.. LENGTH(%XREF_STA_NRS)
  @STA_NR = %XREF_STA_NRS(%I)
  #SET STA'STA_NR':SDI = 1
 #LOOP_END
#BLOCK_END
```

5.5.2.4 Command handling in DNP 3.0 protocol

Requests and responses
Messages from the master to the slave are called requests in the DNP 3.0 protocol. A request can be, for example, a request for data or a latch relay command. The slave's response to a request can contain, for example the requested user data or the status of a command.

In the SYS600 implementation of the DNP 3.0 slave protocol, only one command transaction can be open at any one time. This means that the latest command received from the master is kept in a buffer in the NET unit. When a command is confirmed by using the CF attribute, a command with a matching data object type and address is searched from the buffer. If a match is found, a confirmation message is sent to the master and if not, status 13925= DNPC_NO_ACTIVE_COMMAND is returned.

Process object receiving commands
As stated before, commands received from the DNP master are seen as input data by the slave and command messages can be received by REX type process objects.

These process objects activate an event channel, which executes a command procedure. The following attribute values should be the same for all the process objects receiving a command from the master:

- The value of the AN (Action Enabled) attribute should be 1
- The value of the AA (Action Activation) attribute should be 2 (update)
- The value of the AF (Action at First Update) attribute should be 1

5.5.2.5 Command procedures for data commands

The commands sent from the DNP master to SYS600 are seen as input data and received by REX type input process objects.

When this kind of input is updated, the value is read by a command procedure and if necessary converted, before it is written to the actual output objects that send the command to the process devices. The command procedure is activated through an event channel which is bound to the input object.

The number of the output process objects and the SCIL statements used for sending the commands to the process devices depend on the protocol of the device. Each controllable SPA object, for example the circuit breaker, has four output process objects, whereas each RTU or IEC object has only one.

It is also possible to utilize the input data to perform arbitrary internal operations in the application program. The procedures that are presented below cover the basic cases when DNP commands and setpoints can be mapped directly to the corresponding SYS600 outputs. If a more complex relationship between input and outputs is desired, it is necessary to build application specific command procedures.

Object commands

Object commands, such as switching device open/close commands or tap changer raise/lower commands, are received as control relay output block messages by the DNP slave station. These messages can be received by BI (Binary Input) process objects.

The unit number (UN attribute) of the input process object must be the same as the STA object number of the corresponding DNP slave station, and the address of the process object must equal to the address of the command in the DNP master. Table 9 shows the updated attributes of the input process objects which receive commands.

Table 9 shows the updated attributes of the input process objects, which receive commands. An operate command must have been received after a select command within the time determined by the ET attribute of the station.

Table 9: The updated attributes of the input process object receiving object command

Attribute	Values	Description
OV	0,1	Direction of the command 0 = off, 1 = on
RA	3,4,5,6	Function code of the command: 3 = select, 4 = operate,
RB	Integer	On-time, the pulse length of the command
QL	0,1,2	Qualifier derived from the control code of the command

In the following examples, the process object receiving the command has the same logical name as the output objects. The station number of the DNP 3.0 slave station is assumed to be 1. The first example is for a SPA device with output process object indexes as follows: open select 11, close select 12, execute 13 and cancel 14.

```
;reset status
@S = STATUS
;route the command to the output objects
;function code of the command
#CASE %RA
  ;select
  #WHEN 3 #BLOCK
    #IF %OV == 0 #THEN #SET 'LN':POV11 = 1 ; open select
    #ELSE #SET 'LN':POV12 = 1 ; close select
  #BLOCK_END
  ;operate
  #WHEN 4 #SET 'LN':POV13 = 1 ; execute
  ;direct operate, direct operate no ack
  #WHEN 5,6 #BLOCK
    #IF %OV == 0 #THEN #SET 'LN':POV11 = 1 ; open select
    #ELSE #SET 'LN':POV12 = 1 ; close select
    #SET 'LN':POV13 = 1 ; execute
  #BLOCK_END
#CASE_END

;confirm the command, if required
#IF %RA < 6 #THEN #BLOCK
  #IF STATUS == 0 #THEN @CS = 0 ; report ok status
  #ELSE @CS = 6 ; report hardware problems
  #SET STA1:SCF = (%CS,'LN':POA'IX')
#BLOCK_END
```

The second example is for a RTU device with one output process object with index 13. This example also has a check mechanism for the station local/remote switch, which is in this example indicated by a double binary process object STA_LR:P10.

```
;reset status
@S = STATUS

;check the state of the station local remote switch
#IF STA_LR:POV10 == 2 #THEN #BLOCK ; remote
  @LR_CS = 0 ; do not report outputs in local state
  ;route the command to the output objects
  ;function code of the command
   #CASE %RA
    ;select
    #WHEN 3 #SET 'LN':PSE13 = 1 ; select
    ;operate
    #WHEN 4 #SET 'LN':POV13 = %OV ; execute
    ;direct operate, direct operate no ack
    #WHEN 5,6 #BLOCK
      #SET 'LN':PSE13 = 1 ; select
      #SET 'LN':POV13 = %OV ; execute
    #BLOCK_END
   #CASE_END
  #BLOCK_END
  #ELSE #BLOCK ; local
    @LR_CS = 128 ; report outputs in local state
  #BLOCK_END

;confirm the command, if required
#IF %RA < 6 #THEN #BLOCK
  #IF STATUS == 0 #THEN @CS = %LR_CS ; report ok status
  #ELSE @CS = %LR_CS + 6 ; report hardware problems
  #SET STA1:SCF = (%CS,'LN':POA'IX')
#BLOCK_END
```

Direct operate - no ack type commands are not confirmed by using the CF attribute.

Analog setpoints

Analog output block messages from the DNP master can be received by AI (Analog Input) process objects.

The unit number (UN attribute) of the input process object must be the same as the STA object number of the corresponding DNP slave station, and the address of the process object must be equal to the address of the command in the DNP master.

The attributes in the following Table 10 of the input process objects receiving the commands are updated.

Table 10: The updated attributes of the input process objects receiving setpoint commands

Attribute	Values	Description
OV	Real or Integer	Value , range depends on the Variation
RA	3,4,5,6	Function code of the command: 3 = select, 4 = operate, 5 = direct operate, 6= direct operate - no ack
RB	0...255	Control status of the command

5.5.2.6 Command procedures for application and system commands

Application and system commands, such as restart commands, can be received by an ANSI bit stream process object. The object's unit number is equal to the station number of the DNP 3.0 slave station. The object's address is equal to the configured value of the CA attribute of the DNP 3.0 slave station. The NET unit sends messages to the process object as transparent data. The messages must be interpreted using SCIL statements.

The following example provides the SCIL code needed to initiate the cold restart and warm restart commands, the function codes 13 and 14 respectively.

```
;unpack the system command
@SYS_CMD = UNPACK_STR(%BS,8)

;application control function code of the command
@FUNC_CODE = %SYS_CMD(2)

;perform the command based on the function code
#CASE %FUNC_CODE

;cold restart
#WHEN 13 #BLOCK

;code needed in cold restart
#BLOCK_END

;warm restart
#WHEN 14 #BLOCK

;code needed in cold restart
#BLOCK_END
#CASE_END
```

The following actions should be taken when one of the restart commands is received. Usually, these commands vary from one application to another. The cold restart command is especially problematic since the base system computer cannot be switched off using SCIL. Listed below are some actions that can be taken with these commands.

- Restart PC-NET by setting LINn:BLT to "NONE" and after a pause back to "INTEGRATED". After this, the lines and stations must be re-created and the NET database re-initialized.
- Reset the message queues in the NET unit by using the RS attribute of the DNP 3.0 slave station.
- Send an interrogation command to all the process devices. This updates the process database, and thus the NET database as well, all the way from the process devices.
- Re-initialize the NET internal database by using the command procedure presented earlier in this document.

5.6 Signal engineering

The term signal engineering here means the engineering needed to establish communication to an NCC using the DNP protocol. The signal engineering in COM500*i* is made using the signal X-reference tool after the system configuration is completed and the process database for the related to process devices is ready. The signal engineering process has been described in detail in COM500*i* Users Guide manual of the SYS600 product. The principal sequence for the signal engineering is:

1. Make a list of all signals that are to be transferred between the master and the slave. Include the data object type, variation and address of the static data object. If an event is to be sent, also include data object type, variation, class and delta of the event.
2. Determine the communication mode,that is, whether polling or unsolicited messages are used for sending data to the master, based on the information about the behavior of the master system.
3. Define the NCCs, communication mode and the cross-reference data using the Signal X-references tool of COM500*i*.
4. Test each signal.

5.7 Status codes

The status codes for the DNP 3.0 Slave protocol are defined in the SYS600 Status Codes manual. Some typical reasons for some of the status codes are also given.

Status codes are sent as system messages which can be received by analog input project objects with a unit number (UN) 0 and an object address (OA) as determined by the MI attribute of the line or station, or alternatively, they are returned as a response to a SCIL command accessing a DNP station object.

5.8 Device profile

5.8.1 Device profile (Subset 2)

Table 11: The device profile that describes the implementation of the DNP 3.0 slave protocol in SYS600, Subset 2: the default subset.

<table>
<tr><td colspan="2">DNP 3.0
DEVICE PROFILE DOCUMENT</td></tr>
<tr><td colspan="2">Vendor Name: Hitachi Power Grids, Grid Automation Products</td></tr>
<tr><td colspan="2">Device Name: SYS600 version 10</td></tr>
<tr><td>Highest DNP Level Supported:
For Requests: Subset Level 2
For Responses: Subset Level 2</td><td>Device Function:
[ ] Master
[x] Slave</td></tr>
<tr><td>Maximum Data Link Frame Size (octets):
Transmitted: 58…292, line attribute ML
Received: (must be 292)</td><td>Maximum Application Fragment Size (octets):
Transmitted: 255…2048, STA attribute ML
Received: 2048</td></tr>
<tr><td>Maximum Data Link Re-tries:
[ ] None
[ ] Fixed at _____
[x] Configurable, range 0 to 10, line attribute EN</td><td>Maximum Application Layer Re-tries:
[x] None
[ ] Configurable</td></tr>
<tr><td colspan="2">Requires Data Link Layer Confirmation:
[ ] Never
[ ] Always
[ ] Sometimes If 'Sometimes', when?_____
[x] Configurable, line attribute LA</td></tr>
<tr><td colspan="2">Requires Application Layer Confirmation:
[ ] Never
[ ] Always (not recommended)
[ ] When reporting Event Data (Slave devices only)
[ ] When sending multi-fragment responses (Slave devices only)
[ ] Sometimes. If 'Sometimes', when?_____
[x] Configurable, station attribute PC</td></tr>
<tr><td colspan="2">Timeouts while waiting for:</td></tr>
<tr><td>Data Link Confirm
[ ] None [ ] Variable [ ] Fixed at __
[x] Configurable, line attribute HT
Complete Appl. Fragment
[ ] None [ ] Variable [ ] Fixed at __
[x] Configurable, station attribute TT
Application Confirm
[ ] None [ ] Variable [ ] Fixed at __
[x] Configurable, station attribute CT
Complete Appl. Response
[x] None [ ] Variable [ ] Fixed at __
[] Configurable</td><td></td></tr>
<tr><td colspan="2">Table continues on next page</td></tr>
</table>

Others: Retransmission after a collision, STA attribute XR (random delay if collision detection used). Complete data link frame: line attribute TI.	
Sends/Executes Control Operations:	
WRITE Binary Outputs [x] Never [ ] Sometimes [ ] Always [ ] Configurable SELECT/OPERATE [ ] Never [ ] Sometimes [x] Always [ ] Configurable DIRECT OPERATE [ ] Never [ ] Sometimes [x] Always [ ] Configurable DIRECT OPERATE - NO ACK [ ] Never [ ] Sometimes [x] Always [ ] Configurable Count > 1 [x] Never [ ] Sometimes [ ] Always [ ] Configurable Pulse On [ ] Never [ ] Sometimes [x] Always [ ] Configurable Pulse Off [ ] Never [ ] Sometimes [x] Always [ ] Configurable Latch On [ ] Never [ ] Sometimes [x] Always [ ] Configurable Latch Off [ ] Never [ ] Sometimes [x] Always [ ] Configurable Queue [x] Never [ ] Sometimes [ ] Always [ ] Configurable Clear Queue [x] Never [ ] Sometimes [ ] Always [ ] Configurable	
FILL OUT THE FOLLOWING ITEMS FOR SLAVE DEVICES ONLY:	
Reports Binary Input Change Events when no specific variation requested: [ ] Never [ ] Only time-tagged [ ] Only non-time-tagged [x] Configurable to send one or the other, depends on data point init	**Reports time-tagged Binary Input Change Events when no specific variation requested:** [ ] Never [ ] Binary Input Change With Time [ ] Binary Input Change With Relative Time [x] Configurable, depends on data point initialization
Sends Unsolicited Responses: [ ] Never [x] Configurable, depends on data point initialization [ ] Only certain objects [ ] Sometimes (attach explanation) [ ] ENABLE/DISABLE UNSOLICITED function codes supported	**Sends Static Data in Unsolicited Responses:** [x] Never [ ] When Device Restarts (depends on data point initialization) [ ] When Status Flags Change (depends on data point initialization)
Default Counter Object/ Variation: [ ] No Counters Reported [x] Configurable, depends on data point initialization [ ] Default Object_____ [ ] Default Variation_____ [ ] Point-by-point list attached	**Counters Roll Over at:** [ ] No Counters Reported [ ] Configurable (attach explanation) [ ] 16 bit [x] 32 Bits, roll-over bits not set [ ] Other Value_____ [ ] Point-by-point list attached
Sends Multi-Fragment Responses:	
[x] Yes	[ ] No

5.8.2 Supported function codes (Subset 2)

Table 12: Supported function codes

CODE	FUNCTION	DESCRIPTION	Supported
Transfer Function Codes			
0	Confirm	Message fragment confirmation No Response	Yes
1	Read	Request objects from outstation Respond with requested objects	Yes
2	Write	Store the specified objects to outstation Respond with status of the operation	Yes
Control Function Codes			
3	Select	Select the output point of outstation Respond with status of control point	Yes
4	Operate	Set the output that has previously been selected Respond with status of control point	Yes
5	Direct operate	Set the output directly Respond with status of control point	Yes
6	Direct operate - no ack	Set the output directly No response	Yes
Freeze Function Codes			
7	Immediate Freeze	Copy the specified objects to freeze buffer Respond with status of operation	Yes
8	Immediate Freeze - no ack	Copy the specified objects to freeze buffer No response	Yes
9	Freeze and Clear	Copy the specified objects to freeze buffer and clear objects Respond with status of operation	Yes
10	Freeze and Clear - no ack	Copy the specified objects to freeze buffer and clear objects No response	Yes
11	Freeze with time	Copy the specified objects to freeze buffer at specified time Respond with status of operation	No
12	Freeze with time - no ack	Copy the specified objects to freeze buffer at specified time No response	No
Application Control Function Codes			
13	Cold Restart	Perform the desired reset sequence Respond with a time object	Yes
14	Warm Restart	Perform the desired partial reset operation Respond with a time object	Yes
15	Initialize Data to Defaults	Initialize the specified data to default Respond with the status of operation	No
16	Initialize Application	Prepare the specified application to run Respond with the status of operation	No

Table continues on next page

CODE	FUNCTION	DESCRIPTION	Supported
17	Start Application	Start the specified application to run Respond with the status of operation	No
18	Stop Application	Stop the specified application to run Respond with the status of operation	No
Configuration Function Codes			
19	Save configuration	Save the configuration Respond with the status of operation	No
20	Enable Unsolicited Messages	Enable Unsolicited Messages Respond with the status of operation	No
21	Disable Unsolicited Messages	Disable Unsolicited Messages Respond with the status of operation	No
22	Assign Class	Assign specified objects to a class Respond with the status of operation	No
Time Synchronization Function Codes			
23	Delay Measurement	Perform propagation delay measurement	Yes
24	Record current time	Used in a network application to allow the Master station and the Out station to record their time at the same instant	Yes
Authentication Request Function Codes			
32	Authentication Request	Authentication Request	Yes
33	Authentication Error	Authentication Error	No
Response Function Codes			
0	Confirm	Message fragment confirmation	Yes
129	Response	Response to requested message	Yes
130	Unsolicited Message	Spontaneous message without request	Yes
131	Authentication Response	Authentication Response	Yes

5.8.3 Supported objects (Subset 2)

Table 13: Supported objects

OBJECT			REQUEST (slave must parse)		RESPONSE (master must parse)	
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**
1	0	Binary Input - All Variations	1	06		
1	1	Binary Input	**1**	**00,01,06**	129, 130	00, 01
1	2	Binary Input with Status	**1**	**00,01,06**	129, 130	00, 01

Table continues on next page

OBJECT			REQUEST (slave must parse)		RESPONSE (master must parse)	
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**
2	0	Binary Input Change - All Variations	1	06,07,08		
2	1	Binary Input Change without Time	1	06,07,08	129, 130	17, 28
2	2	Binary Input Change with Time	1	06,07,08	129, 130	17, 28
2	3	Binary Input Change with Relative Time	1	06,07,08	129, 130	17, 28
3	0	Double bit Input - All Variations	1	06		
3	1	Double bit input	1	00,01,06	129, 130	00, 01
3	2	Double bit input with status	1	00,01,06	129, 130	00, 01
4	0	Double bit input change – All variations	1	06, 07,08		
4	1	Double bit input change event	1	06,07,08	129, 130	17, 28
4	2	Double bit input change event with time	1	06,07,08	129, 130	17, 28
4	3	Double bit input change event with relative time	1	06,07,08	129, 130	17, 28
10	0	Binary Output - All Variations	1	06		
10	1	Binary Output				
10	2	Binary Output Status	**1**	**00,01,06**	129, 130	00, 01
12	0	Control Block - All Variations				
12	1	Control Relay Output Block	3, 4, 5, 6	17, 27,28	129	echo of request + status
12	2	Pattern Control Block				
12	3	Pattern Mask				
20	0	Binary Counter - All Variations	1, 7, 8, 9, 10	06		
20	1	32-Bit Binary Counter	**1**	**00,01,06**	129, 130	00, 01
20	2	16-Bit Binary Counter			129, 130	00, 01
20	3	32-Bit Delta Counter			129, 130	00, 01
20	4	16-Bit Binary Counter			129, 130	00, 01
20	5	32-Bit Binary Counter without Flag	**1**	**00,01,06**	129, 130	00, 01
20	6	16-Bit Binary Counter without Flag			129, 130	00, 01
20	7	32-Bit Delta Counter without Flag			129, 130	00, 01
20	8	16-Bit Delta Counter without Flag			129, 130	00, 01
21	0	Frozen Counter - All Variations	1	06		
21	1	32-Bit Frozen Counter			129, 130	00, 01
21	2	16-Bit Frozen Counter			129, 130	00, 01
21	3	32-Bit Frozen Delta Counter				
21	4	16-Bit Frozen Delta Counter				
21	5	32-Bit Frozen Counter with Time of Freeze				
21	6	16-Bit Frozen Counter with Time of Freeze				
21	7	32-Bit Frozen Delta Counter with Time of Freeze				
21	8	16-Bit Frozen Delta Counter with Time of Freeze				
21	9	32-Bit Frozen Counter without Flag			129, 130	00, 01

Table continues on next page

OBJECT			REQUEST (slave must parse)		RESPONSE (master must parse)	
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)
21	10	16-Bit Frozen Counter without Flag			129, 130	00, 01
21	11	32-Bit Frozen Delta Counter without Flag				
21	12	16-Bit Frozen Delta Counter without Flag				
22	0	Counter Change Event - All Variations	1	06,07,08		
22	1	32-Bit Counter Change Event without Time			129, 130	17, 28
22	2	16-Bit Counter Change Event without Time			129, 130	17, 28
22	3	32-Bit Delta Counter Change Event without Time				
22	4	16-Bit Delta Counter Change Event without Time				
22	5	32-Bit Counter Change Event with Time				
22	6	16-Bit Counter Change Event with Time				
22	7	32-Bit Delta Counter Change Event with Time				
22	8	16-Bit Delta Counter Change Event with Time				
23	0	Frozen Counter Event - All Variations				
23	1	32-Bit Frozen Counter Event without Time				
23	2	16-Bit Frozen Counter Event without Time				
23	3	32-Bit Frozen Delta Counter Event without Time				
23	4	16-Bit Frozen Delta Counter Event without Time				
23	5	32-Bit Frozen Counter Event with Time				
23	6	16-Bit Frozen Counter Event with Time				
23	7	32-Bit Frozen Delta Counter Event with Time				
23	8	16-Bit Frozen Delta Counter Event with Time				
30	0	Analog Input - All Variations	1	06		
30	1	32-Bit Analog Input			129, 130	00, 01
30	2	16-Bit Analog Input	**1**	**00,01,06**	129, 130	00, 01
30	3	32-Bit Analog Input without Flag			129, 130	00, 01
30	4	16-Bit Analog Input without Flag	**1**	**00,01,06**	129, 130	00, 01
31	0	Frozen Analog Input - All Variations				
31	1	32-Bit Frozen Analog Input				
31	2	16-Bit Frozen Analog Input				
31	3	32-Bit Frozen Analog Input with Time of Freeze				
31	4	16-Bit Frozen Analog Input with Time of Freeze				
31	5	32-Bit Frozen Analog Input without Flag				
31	6	16-Bit Frozen Analog Input without Flag				
32	0	Analog Change Event - All Variations	1	06,07,08		
32	1	32-Bit Analog Change Event without Time			129,130	17,28
32	2	16-Bit Analog Change Event without Time			129,130	17,28
32	3	32-Bit Analog Change Event with Time				

Table continues on next page

OBJECT			REQUEST (slave must parse)		RESPONSE (master must parse)	
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**
32	4	16-Bit Analog Change Event with Time				
33	0	Frozen Analog Event - All Variations				
33	1	32-Bit Frozen Analog Event without Time				
33	2	16-Bit Frozen Analog Event without Time				
33	3	32-Bit Frozen Analog Event with Time				
33	4	16-Bit Frozen Analog Event with Time				
40	0	Analog Output Status - All Variations	1	06		
40	1	32-Bit Analog Output Status				
40	2	16-Bit Analog Output Status			129, 130	00, 01
41	0	Analog Output Block - All Variations				
41	1	32-Bit Analog Output Block				
41	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	echo of request + status
50	0	Time and Date - All Variations				
50	1	Time and Date	2	**06** **07** quantity = 1		
50	2	Time and Date with Interval				
51	0	Time and Date CTO - All Variations				
51	1	Time and Date CTO			129, 130	07, quantity=1
51	2	Unsynchronized Time and Date CTO			129, 130	07, quantity=1
52	0	Time Delay - All Variations				
52	1	Time Delay Coarse			129	07, quantity=1
52	2	Time Delay Fine			129	07, quantity=1
60	0					
60	1	Class 0 Data	1	06		
60	2	Class 1 Data	1	06,07,08		
60	3	Class 2 Data	1	06,07,08		
60	4	Class 3 Data	1	06,07,08		
70	1	File Identifier				
80	1	Internal Indications	1 2	All 00,index=7		
81	1	Storage Object				
82	1	Device Profile				
83	1	Private Registration Object				
83	2	Private Registration Object Descriptor				
90	1	Application Identifier				
100	1	Short Floating Point				
100	2	Long Floating Point				
100	3	Extended Floating Point				
101	1	Small Packed Binary-Coded Decimal				

Table continues on next page

OBJECT			REQUEST (slave must parse)		RESPONSE (master must parse)	
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**
101	2	Medium Packed Binary-Coded Decimal				
101	3	Large Packed Binary-Coded Decimal				
		No Object	13			
		No Object	23			

5.8.4 Device profile (Subset 3)

Table 14: The device profile that describes the implementation of the DNP 3.0 slave protocol in SYS600, Subset 3

DNP 3.0 **DEVICE PROFILE DOCUMENT**	
Vendor Name: Hitachi Power Grids, Grid Automation Products	
Device Name: SYS600 version 10	
Highest DNP Level Supported: For Requests: Subset Level 3 For Responses: Subset Level 3	**Device Function:** [ ] Master [x] Slave
Maximum Data Link Frame Size (octets): Transmitted: 58…292, line attribute ML Received: (must be 292)	**Maximum Application Fragment Size (octets):** Transmitted: 255…2048, STA attribute ML Received: 2048
Maximum Data Link Re-tries: [ ] None [ ] Fixed at _____ [x] Configurable, range 0 to 10, line attribute EN	**Maximum Application Layer Re-tries:** [x] None [ ] Configurable
Requires Data Link Layer Confirmation: [ ] Never [ ] Always [ ] Sometimes If 'Sometimes', when?_____ [x] Configurable, line attribute LA	
Requires Application Layer Confirmation: [ ] Never [ ] Always (not recommended) [ ] When reporting Event Data (Slave devices only) [ ] When sending multi-fragment responses (Slave devices only) [ ] Sometimes. If 'Sometimes', when?_____ [x] Configurable, station attribute PC	
Timeouts while waiting for:	
Data Link Confirm [ ] None [ ] Variable [ ] Fixed at __ [x] Configurable, line attribute HT Complete Appl. Fragment [ ] None [ ] Variable [ ] Fixed at __ [x] Configurable, station attribute TT Application Confirm [ ] None [ ] Variable [ ] Fixed at __ [x] Configurable, station attribute CT Complete Appl. Response [x] None [ ] Variable [ ] Fixed at __ [] Configurable	
Others: Retransmission after a collision, STA attribute XR (random delay if collision detection used). Complete data link frame: line attribute TI.	
Table continues on next page	

Sends/Executes Control Operations:	
WRITE Binary Outputs [x] Never [ ] Sometimes [ ] Always [ ] Configurable SELECT/OPERATE [ ] Never [ ] Sometimes [x] Always [ ] Configurable DIRECT OPERATE [ ] Never [ ] Sometimes [x] Always [ ] Configurable DIRECT OPERATE - NO ACK [ ] Never [ ] Sometimes [x] Always [ ] Configurable Count > 1 [] Never [ ] Sometimes [x] Always [ ] Configurable Pulse On [ ] Never [ ] Sometimes [x] Always [ ] Configurable Pulse Off [ ] Never [ ] Sometimes [x] Always [ ] Configurable Latch On [ ] Never [ ] Sometimes [x] Always [ ] Configurable Latch Off [ ] Never [ ] Sometimes [x] Always [ ] Configurable Queue [x] Never [ ] Sometimes [ ] Always [ ] Configurable Clear Queue [x] Never [ ] Sometimes [ ] Always [ ] Configurable	
FILL OUT THE FOLLOWING ITEMS FOR SLAVE DEVICES ONLY:	
Reports Binary Input Change Events when no specific variation requested: [ ] Never [ ] Only time-tagged [ ] Only non-time-tagged [x] Configurable to send one or the other, depends on data point init	**Reports time-tagged Binary Input Change Events when no specific variation requested:** [ ] Never [ ] Binary Input Change With Time [ ] Binary Input Change With Relative Time [x] Configurable, depends on data point initialization
Sends Unsolicited Responses: [ ] Never [x] Configurable, depends on data point initialization [ ] Only certain objects [ ] Sometimes (attach explanation) [ ] ENABLE/DISABLE UNSOLICITED function codes supported	**Sends Static Data in Unsolicited Responses:** [x] Never [ ] When Device Restarts (depends on data point initialization) [ ] When Status Flags Change (depends on data point initialization)
Default Counter Object/ Variation: [ ] No Counters Reported [x] Configurable, depends on data point initialization [ ] Default Object_____ [ ] Default Variation_____ [ ] Point-by-point list attached	**Counters Roll Over at:** [ ] No Counters Reported [ ] Configurable (attach explanation) [ ] 16 bit [x] 32 Bits, roll-over bits not set [ ] Other Value_____ [ ] Point-by-point list attached
Sends Multi-Fragment Responses:	
[x] Yes	[ ] No

5.8.5 Supported function codes (Subset 3)

Table 15: Supported function codes

CODE	FUNCTION	DESCRIPTION	Supported
Transfer Function Codes			
0	Confirm	Message fragment confirmation No Response	Yes
1	Read	Request objects from outstation Respond with requested objects	Yes
2	Write	Store the specified objects to outstation Respond with status of the operation	Yes
Control Function Codes			
3	Select	Select the output point of outstation Respond with status of control point	Yes
4	Operate	Set the output that has previously been selected Respond with status of control point	Yes
5	Direct operate	Set the output directly Respond with status of control point	Yes
6	Direct operate - no ack	Set the output directly No response	Yes
Freeze Function Codes			
7	Immediate Freeze	Copy the specified objects to freeze buffer Respond with status of operation	Yes
8	Immediate Freeze - no ack	Copy the specified objects to freeze buffer No response	Yes
9	Freeze and Clear	Copy the specified objects to freeze buffer and clear objects Respond with status of operation	Yes
10	Freeze and Clear - no ack	Copy the specified objects to freeze buffer and clear objects No response	Yes
11	Freeze with time	Copy the specified objects to freeze buffer at specified time Respond with status of operation	No
12	Freeze with time - no ack	Copy the specified objects to freeze buffer at specified time No response	No
Application Control Function Codes			
13	Cold Restart	Perform the desired reset sequence Respond with a time object	Yes
14	Warm Restart	Perform the desired partial reset operation Respond with a time object	Yes
15	Initialize Data to Defaults	Initialize the specified data to default Respond with the status of operation	No
16	Initialize Application	Prepare the specified application to run Respond with the status of operation	No

Table continues on next page

CODE	FUNCTION	DESCRIPTION	Supported
17	Start Application	Start the specified application to run Respond with the status of operation	No
18	Stop Application	Stop the specified application to run Respond with the status of operation	No
Configuration Function Codes			
19	Save configuration	Save the configuration Respond with the status of operation	No
20	Enable Unsolicited Messages	Enable Unsolicited Messages Respond with the status of operation	Yes
21	Disable Unsolicited Messages	Disable Unsolicited Messages Respond with the status of operation	Yes
22	Assign Class	Assign specified objects to a class Respond with the status of operation	Yes
Time Synchronization Function Codes			
23	Delay Measurement	Perform propagation delay measurement	Yes
24	Record current time	Used in a network application to allow the Master station and the Out station to record their time at the same instant	Yes
Authentication Request Function Codes			
32	Authentication Request	Authentication Request	Yes
33	Authentication Error	Authentication Error	No
Response Function Codes			
0	Confirm	Message fragment confirmation	Yes
129	Response	Response to requested message	Yes
130	Unsolicited Message	Spontaneous message without request	Yes
131	Authentication Response	Authentication Response	Yes

5.8.6 Supported objects (Subset 3)

Table 16: Supported objects

OBJECT	REQUEST (slave must parse)					
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**
1	0	Binary Input - All Variations	1,22	00,01,06		
1	1	Binary Input	1	00,01,06	129, 130	00, 01
1	2	Binary Input with Status	1	00,01,06	129, 130	00, 01

Table continues on next page

OBJECT	REQUEST (slave must parse)					
2	0	Binary Input Change - All Variations	1	06,07,08		
2	1	Binary Input Change without Time	1	06,07,08	129, 130	17, 28
2	2	Binary Input Change with Time	1	06,07,08	129, 130	17, 28
2	3	Binary Input Change with Relative Time	1	06,07,08	129, 130	17, 28
3	0	Double bit Input - All Variations	1,22	00, 01, 06		
3	1	Double bit input	1	00,01,06	129, 130	00, 01
3	2	Double bit input with status	1	00,01,06	129, 130	00, 01
4	0	Double bit input change – All variations	1	06, 07,08		
4	1	Double bit input change event	1	06,07,08	129, 130	17, 28
4	2	Double bit input change event with time	1	06,07,08	129, 130	17, 28
4	3	Double bit input change event with relative time	1	06,07,08	129, 130	17, 28
10	0	Binary Output - All Variations	1	06		
10	1	Binary Output				
10	2	Binary Output Status	**1**	00,01,06	129, 130	00, 01
12	0	Control Block - All Variations				
12	1	Control Relay Output Block	3, 4, 5, 6	17, 27,28	129	echo of request
12	2	Pattern Control Block	5, 6	17, 28	129	echo of request
12	3	Pattern Mask	5, 6	00,01	129	echo of request
20	0	Binary Counter - All Variations	1, 7, 8, 9, 10, 22	00,01,06		
20	1	32-Bit Binary Counter	1	00,01,06	129, 130	00, 01
20	2	16-Bit Binary Counter	1	00,01,06	129, 130	00, 01
20	3	32-Bit Delta Counter	1	00,01,06	129, 130	00, 01
20	4	16-Bit Binary Counter	1	00,01,06	129, 130	00, 01
20	5	32-Bit Binary Counter without Flag	1	00,01,06	129, 130	00, 01
20	6	16-Bit Binary Counter without Flag	1	00,01,06	129, 130	00, 01
20	7	32-Bit Delta Counter without Flag	1	00,01,06	129, 130	00, 01
20	8	16-Bit Delta Counter without Flag	1	00,01,06	129, 130	00, 01
21	0	Frozen Counter - All Variations	1,22	00,01,06		
21	1	32-Bit Frozen Counter	1	00,01,06	129, 130	00, 01
21	2	16-Bit Frozen Counter	1	00,01,06	129, 130	00, 01
21	3	32-Bit Frozen Delta Counter	1	00,01,06	129, 130	00, 01
21	4	16-Bit Frozen Delta Counter	1	00,01,06	129, 130	00, 01
21	5	32-Bit Frozen Counter with Time of Freeze				
21	6	16-Bit Frozen Counter with Time of Freeze				
21	7	32-Bit Frozen Delta Counter with Time of Freeze				
21	8	16-Bit Frozen Delta Counter with Time of Freeze				
21	9	32-Bit Frozen Counter without Flag	1	00,01,06	129, 130	00, 01

Table continues on next page

OBJECT	REQUEST (slave must parse)					
21	10	16-Bit Frozen Counter without Flag	1	00,01,06	129, 130	00, 01
21	11	32-Bit Frozen Delta Counter without Flag				
21	12	16-Bit Frozen Delta Counter without Flag				
22	0	Counter Change Event - All Variations	1	06,07,08		
22	1	32-Bit Counter Change Event without Time	1	06,07,08	129, 130	17, 28
22	2	16-Bit Counter Change Event without Time	1	06,07,08	129, 130	17, 28
22	3	32-Bit Delta Counter Change Event without Time	1	06,07,08	129, 130	17, 28
22	4	16-Bit Delta Counter Change Event without Time	1	06,07,08	129, 130	17, 28
22	5	32-Bit Counter Change Event with Time	1	06,07,08	129, 130	17, 28
22	6	16-Bit Counter Change Event with Time	1	06,07,08	129, 130	17, 28
22	7	32-Bit Delta Counter Change Event with Time				
22	8	16-Bit Delta Counter Change Event with Time				
23	0	Frozen Counter Event - All Variations	1	06,07,08		
23	1	32-Bit Frozen Counter Event without Time	1	06,07,08	129, 130	17, 28
23	2	16-Bit Frozen Counter Event without Time	1	06,07,08	129, 130	17, 28
23	3	32-Bit Frozen Delta Counter Event without Time	1	06,07,08	129, 130	17, 28
23	4	16-Bit Frozen Delta Counter Event without Time	1	06,07,08	129, 130	17, 28
23	5	32-Bit Frozen Counter Event with Time				
23	6	16-Bit Frozen Counter Event with Time				
23	7	32-Bit Frozen Delta Counter Event with Time				
23	8	16-Bit Frozen Delta Counter Event with Time				
30	0	Analog Input - All Variations	1,22	00,01,06		
30	1	32-Bit Analog Input	1	00,01,06	129, 130	00, 01
30	2	16-Bit Analog Input	**1**	**00,01,06**	129, 130	00, 01
30	3	32-Bit Analog Input without Flag	1	00,01,06	129, 130	00, 01
30	4	16-Bit Analog Input without Flag	**1**	**00,01,06**	129, 130	00, 01
30	5	Analog Input - Single-Precision, Floating-Point with Flag	1	00,01,06	129, 130	00, 01
31	0	Frozen Analog Input - All Variations				
31	1	32-Bit Frozen Analog Input				
31	2	16-Bit Frozen Analog Input				
31	3	32-Bit Frozen Analog Input with Time of Freeze				
31	4	16-Bit Frozen Analog Input with Time of Freeze				
31	5	32-Bit Frozen Analog Input without Flag				

Table continues on next page

OBJECT	REQUEST (slave must parse)					
31	6	16-Bit Frozen Analog Input without Flag				
32	0	Analog Change Event - All Variations	1	06,07,08		
32	1	32-Bit Analog Change Event without Time	1	06,07,08	129,130	17,28
32	2	16-Bit Analog Change Event without Time	1	06,07,08	129,130	17,28
32	3	32-Bit Analog Change Event with Time	1	06,07,08	129,130	17,28
32	4	16-Bit Analog Change Event with Time	1	06,07,08	129,130	17,28
32	5	Analog Change Event - Single-Precision, Floating-Point without Time	1	06,07,08	129,130	17,28
32	7	Analog Change Event - Single-Precision, Floating-Point with Time	1	06,07,08	129,130	17,28
33	0	Frozen Analog Event - All Variations				
33	1	32-Bit Frozen Analog Event without Time				
33	2	16-Bit Frozen Analog Event without Time				
33	3	32-Bit Frozen Analog Event with Time				
33	4	16-Bit Frozen Analog Event with Time				
40	0	Analog Output Status - All Variations	1	00,01,06		
40	1	32-Bit Analog Output Status	1	00,01,06	129, 130	00, 01
40	2	16-Bit Analog Output Status	1	00,01,06	129, 130	00, 01
40	3	Analog Output Status - Single-Precision, Floating-Point with Flag	1	00,01,06	129, 130	00, 01
41	1	32-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	echo of request
41	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	echo of request
41	3	Analog output - Single-Precision, Floating-Point	3, 4, 5, 6	17, 28	129	echo of request
50	0	Time and Date - All Variations				
50	1	Time and Date	2 (see 4.14)	07 quantity = 1		
		1	07 quantity = 1	129	07 quantity = 1	
50	2	Time and Date with Interval				
51	0	Time and Date CTO - All Variations				
51	1	Time and Date CTO			129, 130	07, quantity=1
51	2	Unsynchronized Time and Date CTO			129, 130	07, quantity=1
52	0	Time Delay - All Variations				
52	1	Time Delay Coarse			129	07, quantity=1
52	2	Time Delay Fine			129	07, quantity=1
60	0					
60	1	Class 0 Data	1	06		
60	2	Class 1 Data	1 20, 21, 22	06,07,08 06		
60	3	Class 2 Data	1 20, 21, 22	06,07,08 06		

Table continues on next page

OBJECT	REQUEST (slave must parse)					
60	4	Class 3 Data	1 20, 21, 22	06,07,08 06		
70	1	File Identifier				
80	1	Internal Indications	1 2	00,01 00 index = 7		
81	1	Storage Object				
82	1	Device Profile				
83	1	Private Registration Object				
83	2	Private Registration Object Descriptor				
90	1	Application Identifier				
100	1	Short Floating Point				
100	2	Long Floating Point				
100	3	Extended Floating Point				
101	1	Small Packed Binary-Coded Decimal				
101	2	Medium Packed Binary-Coded Decimal				
101	3	Large Packed Binary-Coded Decimal				
		No Object	13			
		No Object	23			

5.8.7 TCP/UDP additions for LAN/WAN network

SYS600 LAN/WAN DNP uses the default IP address provided by the operating system.

A created DNP Slave line reserves a port number

```
2501+linenumber
```

for its internal use.

In TCP/IP mode (connection-oriented), the connection is established to port

```
20000 (default)
```

of the slave device. The local address the line is listening to is defined with the line attribute LD. The port number can be configured using an option in the LD attribute. Only one connection to each master station is established at any one time. The IP address of the master is configured with the IA and AI attributes of the station object.

In UDP/IP mode (connectionless) all data is sent and received through port

```
20000
```

The slave device must receive messages from this port only. All messages that are sent by the slave device must be sent to this port of the master. The IP address of the slave is configured with the IA attribute of the station object. The multidrop configuration requires redefinition of local ports though all the communication goes through port 20000. This configuration has been explained in the description of the IA attribute.

5.8.8 Secure authentication

DNP 3.0 in SYS600 supports secure authentication versions v2 and v5. For secure authentication version v2, the supported message types are of object type 120 (Authentication), variations 1-7 and 9. For secure authentication version v5, the supported message types are of object type 120 (Authentication), variations 1-7 and 9-15. The function codes used with these object types are 32 = Authentication Request and 131 = Authentication Response. The supported qualifiers are fixed in DNP 3.0 standard.

5.8.9 Example topologies:

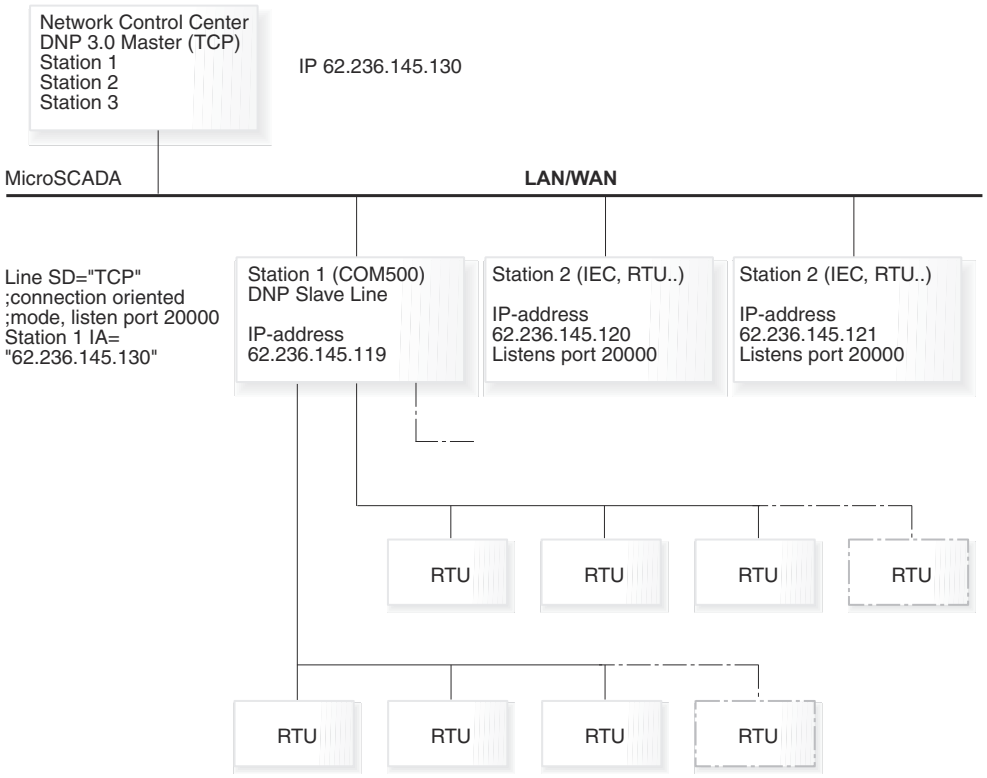


Figure 14: SYS600 (COM500i) as DNP3.0 TCP slave in multidrop environment

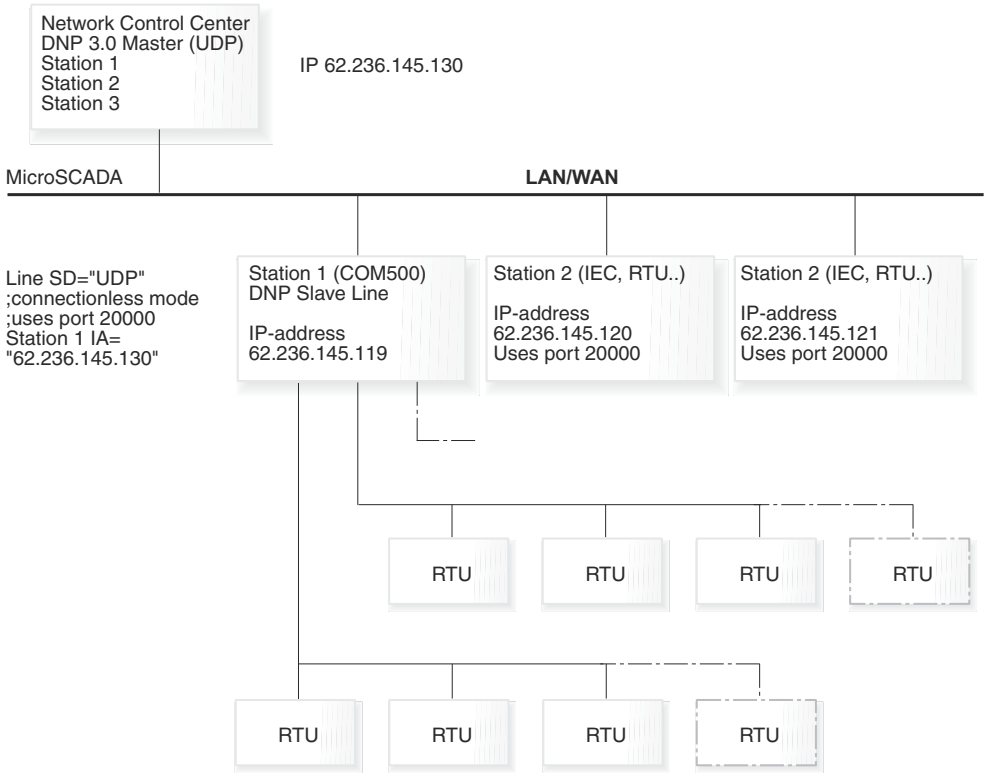


Figure 15: SYS600 (COM500i) as DNP3.0 UDP slave in multidrop environment

Appendix A Configuration examples

1.1 DNP Slave in TCP mode

```
@NET=4
@STA=1
@LINE=1
#SET NET'NET':SPO'LINE'=35
#SET NET'NET':SDV(30)=('STA','LINE')
#SET NET'NET':SSD'LINE'="TCP"
#SET NET'NET':SEN'LINE'=3
#SET NET'NET':SMS'LINE'='NET'
#SET NET'NET':SPD'LINE'=0
#SET NET'NET':SLK'LINE'=15
#SET NET'NET':STI'LINE'=3
#SET NET'NET':SPS'LINE'=50
#SET NET'NET':SLA'LINE'=0
#SET NET'NET':STW'LINE'=0
#SET NET'NET':SIU'LINE'=1
#SET NET'NET':SIU'LINE'=0
#SET STA'STA':SAL=1
#SET STA'STA':SDR=1
#SET STA'STA':SIA="192.168.1.119"
#SET STA'STA':SSA='STA'
#SET STA'STA':SMA=10
#SET STA'STA':STT=25
#SET STA'STA':SCT=10
#SET STA'STA':SRT=30
#SET STA'STA':SPC=1
#SET STA'STA':SRM=4
#SET STA'STA':SIU=1
#SET NET'NET':SIU'LINE'=1
```

1.2 DNP Slave in UDP mode

```
@NET=4
@STA=1
@LINE=1
#SET NET'NET':SPO'LINE'=35
#SET NET'NET':SDV(30)=('STA','LINE')
#SET NET'NET':SSD'LINE'="UDP"
#SET NET'NET':SEN'LINE'=3
#SET NET'NET':SMS'LINE'='NET'
#SET NET'NET':SPD'LINE'=0
#SET NET'NET':SLK'LINE'=15
#SET NET'NET':STI'LINE'=3
#SET NET'NET':SPS'LINE'=50
#SET NET'NET':SLA'LINE'=0
#SET NET'NET':STW'LINE'=0
#SET NET'NET':SIU'LINE'=1
#SET NET'NET':SIU'LINE'=0
#SET STA'STA':SAL=1
#SET STA'STA':SDR=1
#SET STA'STA':SIA="192.168.1.119"
#SET STA'STA':SSA='STA'
#SET STA'STA':SMA=10
#SET STA'STA':STT=25
#SET STA'STA':SCT=10
#SET STA'STA':SRT=30
#SET STA'STA':SPC=1
#SET STA'STA':SRM=4
```

```
#SET STA'STA':SIU=1
#SET NET'NET':SIU'LINE'=1
```

Appendix B Examples of communication system configuration

The following SCIL procedures make the communication system configuration which is related to the base system configuration example presented earlier in this document. The first procedure creates an DNP 3.0 Slave line and one station on this line.

```
;*********************************************************************
; INPUT PARAMETERS
@NET = 3 ; NODE NUMBER OF THE PC-NET
@LINE = 1 ; LINE NUMBER
@STATION = 1; SLAVE ADDRESS
@MASTER_STATION = 1 ; MASTER ADDRESS
@APPLIC = 1
;*********************************************************************
; CREATE A DNP 3.0 LINE TO NET
#IF NET'NET':SPO'LINE'==0 #THEN #BLOCK
 #SET NET'NET':SPO'LINE' = 35 ;DNP 3.0 slave
 #SET NET'NET':SSD'LINE' = "COM1" ;system device name
 #SET NET'NET':SMS'LINE' = %APPLIC ;message application
 #SET NET'NET':SMI'LINE' = %LINE+(6000+(%NET*100)) ;message identifier
 #SET NET'NET':SBR'LINE' = 9600 ;baud rate
 #SET NET'NET':SPY'LINE' = 2 ;parity
 #SET NET'NET':SRD'LINE' = 8 ;receive bit count
 #SET NET'NET':STD'LINE' = 8 ;transmit bit count
 #SET NET'NET':SSB'LINE' = 1 ;stop bit count
 #SET NET'NET':SLK'LINE' = 15 ;no collision detection
 #SET NET'NET':SPS'LINE' = 50 ;buffer pool size
 #SET NET'NET':SML'LINE' = 230 ;maximum message length (bytes)
 #SET NET'NET':SXR'LINE' = 0 ;max. random delay for retransm.(ms)
 #SET NET'NET':SDE'LINE' = 50 ;CTS delay (ms)
 #SET NET'NET':STW'LINE' = 5 ;transmission wait delay (ms)
 #SET NET'NET':SHT'LINE' = 2000 ;header timeout (ms)
 #SET NET'NET':STI'LINE' = 2 ;response timeout (ms)
 #SET NET'NET':SRK'LINE' = 0 ;RTS keep up padding characters
 #SET NET'NET':SRI'LINE' = 0 ;receiver disabling time (ms)
 #SET NET'NET':SPD'LINE' = 5000 ;polling delay (ms)
 #SET NET'NET':SEN'LINE' = 1 ;retransmission limit
 #SET NET'NET':SLA'LINE' = 1 ;link layer conf. enabled
 #SET NET'NET':SOM'LINE' = 0; operating mode
 #SET NET'NET':SRY'LINE' = 1; RTS keepup delay
#BLOCK_END
;*********************************************************************
; CREATE DNP 3.0 STATION TO NET
@STA=%STATION
@MASTER = %MASTER_STATION
 #SET NET'NET':SDV(30) = (%STA,%LINE) ;create station to line
 #SET STA'STA':SIU = 0 ;set station out of use
 #SET STA'STA':SAL = 1 ;allocation
 #SET STA'STA':SAS = %APPLIC ;allocating application
 #SET STA'STA':SMI = 30000+%STA ;message identification
 #SET STA'STA':SMS = %APPLIC ;message application
 #SET STA'STA':SSE = 1 ;system messages enabled
 #SET STA'STA':SSA = %STA ;slave address
 #SET STA'STA':SMA = %MASTER ;master address
 #SET STA'STA':SDI = 0 ;database not intialised
 #SET STA'STA':SIL = 2 ;info address length (bytes)
 #SET STA'STA':STC = 0 ;timesynchronization;(0,1,2,3)
 #SET STA'STA':SCA = 32000 ;command address
 #SET STA'STA':SPC = 1 ;process data confirmation
 #SET STA'STA':SET = 30 ;execute waiting time (s)
```

```
 #SET STA'STA':SST = 5000 ;SYS waiting time (ms)
 #SET STA'STA':SCT = 10 ;confirmation timeout (s)
 #SET STA'STA':STT = 10 ;transport layer timeout (s)
 #SET STA'STA':SDR = 0 ;direction
 #SET STA'STA':SAR = 0 ;appl. message data retries
 #SET STA'STA':SML = 2048 ;max. message length
 #SET STA'STA':SRM = 0 ;running mode
 #SET STA'STA':SIU = 1 ;set station in use
; Set line in use
#SET NET'NET':SIU'LINE' = 1
```

The second example is for a DNP V3.0 slave line with dial-up

```
;*********************************************************************
; INPUT PARAMETERS
@NET = 3 ; NODE NUMBER OF THE PC-NET
@LINE = 1 ; LINE NUMBER
@STATION = 1; SLAVE ADDRESS
@MASTER_STATION = 1 ; MASTER ADDRESS
@APPLIC = 1
;*********************************************************************
; CREATE A DNP 3.0 LINE TO NET
#IF NET'NET':SPO'LINE'==0 #THEN #BLOCK
 #SET NET'NET':SPO'LINE' = 35 ;DNP 3.0 slave
 #SET NET'NET':SSD'LINE' = "COM1" ;system device name
 #SET NET'NET':SMS'LINE' = %APPLIC ;message application
 #SET NET'NET':SMI'LINE' = %LINE+(6000+(%NET*100)) ;message identifier
 #SET NET'NET':SBR'LINE' = 9600 ;baud rate
 #SET NET'NET':SPY'LINE' = 2 ;parity
 #SET NET'NET':SRD'LINE' = 8 ;receive bit count
 #SET NET'NET':STD'LINE' = 8 ;transmit bit count
 #SET NET'NET':SSB'LINE' = 1 ;stop bit count
 #SET NET'NET':SLK'LINE' = 15 ;no collision detection
 #SET NET'NET':SPS'LINE' = 50 ;buffer pool size
 #SET NET'NET':SML'LINE' = 230 ;maximum message length (bytes)
 #SET NET'NET':SXR'LINE' = 0 ;max. random delay for retransm.(ms)
 #SET NET'NET':SDE'LINE' = 50 ;CTS delay (ms)
 #SET NET'NET':STW'LINE' = 5 ;transmission wait delay (ms)
 #SET NET'NET':SHT'LINE' = 2000 ;header timeout (ms)
 #SET NET'NET':STI'LINE' = 2 ;response timeout (ms)
 #SET NET'NET':SRK'LINE' = 0 ;RTS keep up padding characters
 #SET NET'NET':SRI'LINE' = 0 ;receiver disabling time (ms)
 #SET NET'NET':SPD'LINE' = 5000 ;polling delay (ms)
 #SET NET'NET':SEN'LINE' = 1 ;retransmission limit
 #SET NET'NET':SLA'LINE' = 1 ;link layer conf. enabled
 #SET NET'NET':SOM'LINE' = 0; operating mode
 #SET NET'NET':SRY'LINE' = 1; RTS keepup delay
 #SET NET'NET':SAC'LINE' = 1 ;enable autocaller
;*************** Configure Autocaller ********************************
 #SET NET'NET':SCL'LINE' = 0 ;connection time limited
 #SET NET'NET':SCT'LINE' = 0 ;connection time
 #SET NET'NET':SDD'LINE' = 0 ;radio disc. delay
 #SET NET'NET':SPC'LINE' = 0 ;pulse dialing
 #SET NET'NET':SRC'LINE' = 1 ;remote calls enabled
 #SET NET'NET':SRW'LINE' = 0 ;radio connecton wait time
;*********************************************************************
#BLOCK_END
;*********************************************************************
; CREATE DNP 3.0 STATION TO NET
@STA=%STATION
@MASTER = %MASTER_STATION
 #SET NET'NET':SDV(30) = (%STA,%LINE) ;create station to line
 #SET STA'STA':SIU = 0 ;set station out of use
 #SET STA'STA':SAL = 1 ;allocation
 #SET STA'STA':SAS = %APPLIC ;allocating application
 #SET STA'STA':SMI = 30000+%STA ;message identification
 #SET STA'STA':SMS = %APPLIC ;message application
 #SET STA'STA':SSE = 1 ;system messages enabled
 #SET STA'STA':SSA = %STA ;slave address
```

```
 #SET STA'STA':SMA = %MASTER ;master address
 #SET STA'STA':SDI = 0 ;database not intialised
 #SET STA'STA':SIL = 2 ;info address length (bytes)
 #SET STA'STA':STC = 0 ;timesynchronization;(0,1,2,3)
 #SET STA'STA':SCA = 32000 ;command address
 #SET STA'STA':SPC = 1 ;process data confirmation
 #SET STA'STA':SET = 30 ;execute waiting time (s)
 #SET STA'STA':SST = 5000 ;SYS waiting time (ms)
 #SET STA'STA':SCT = 10 ;confirmation timeout (s)
 #SET STA'STA':STT = 10 ;transport layer timeout (s)
 #SET STA'STA':SDR = 0 ;direction
 #SET STA'STA':SAR = 0 ;appl. message data retries
 #SET STA'STA':SML = 2048 ;max. message length
 #SET STA'STA':SRM = 0 ;running mode
 #SET STA'STA':SIU = 1 ;set station in use
; Set line in use
 #SET NET'NET':SIU'LINE' = 1
```

Index

Hitachi ABB Power Grids
Grid Automation Products
PL 688
65101 Vaasa, Finland

https://hitachiabb-powergrids.com/microscadax

Scan this QR code to visit our website

1MRK 511 487-UEN