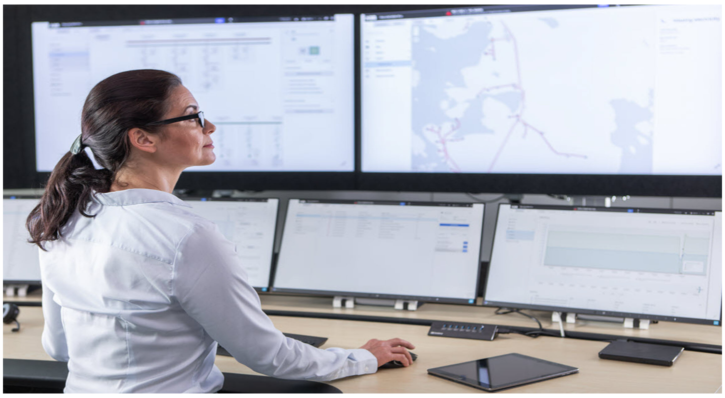
GRID AUTOMATION PRODUCTS

MicroSCADA X SYS600 10.2

System Objects



Document ID: 1MRK 511 482-UEN
Issued: March 2021
Revision: A
Product version: 10.2

Table of contents

Section 1 Copyrights

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 About this manual

2.1 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.

2.2 Intended audience

This manual is intended for installation personnel, administrators and skilled operators to support installation of the software.

2.3 Related documents

Name of the manual	Document ID
SYS600 10.2 Programming Language SCIL	1MRK 511 479-UEN
SYS600 10.2 System Configuration	1MRK 511 481-UEN
SYS600 10.2 Application Objects	1MRK 511 467-UEN
SYS600 10.2 Status Codes	1MRK 511 480-UEN

2.4 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

2.5 Document revisions

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

Section 3 Introduction

3.1 About this section

The purpose of this section is to introduce the role the system objects have in the system. It introduces the system objects and their attributes.

3.2 System objects

System objects are programmable units that specify the system configuration and communication of the SYS600 distributed system.

There are two main types of system objects:

- Base system objects (B), which define the configuration for the base system.
- Communication system objects (S), which define the configuration and communication properties for the process communication system.

The base system objects and the communication system objects, along with the configuration data of the communication modules determine the SYS600 configuration. The communication system objects give the communication units an image of the devices connected to them. The base system objects give the SYS600 main program an image of the devices and the software used by the base system.

When connecting a device to the SYS600 system, some configuration is required either in base system configuration file or in one or more of the NET configuration files or in both places. Connecting a station, for example, requires both a base system and a communication object definition.

3.3 Base system objects

Base system objects define the physical and logical connections and the software and hardware parameters of the base system and its applications. They also define the logical connections to NETs and other base systems and their applications. Base system objects are also used for modification of the base system configuration. For instance, they are used for modification of logical connections to printers, NET communication units and other base systems, momentary connections between different applications and so on.

3.4 Communication system objects

Communication system objects and their attributes specify NET configurations and handle process communication. They give the NET unit an image of the communication lines. For instance, they have the information on which protocol is used on the line, baud rates, way of communication, where to forward information coming from the line and so on. They also give an image of connected devices, such as other communication units, base systems, stations and printers. Each process NET unit has its own configuration file containing a SCIL program.

3.5 Attributes

The properties of a part of a system are described by system object attributes. For instance, the baud rate of a communication line, the station address of a process station and memory allocations in the base system are stored in a corresponding system object attribute. Normally, a system object has several attributes and, thus, contains several types of data. The attributes are identified by two letter names, combined of letters A ... Z.

System object attributes, which specify the properties of a part of the system and affect the system configuration, are called configurable attributes. Other attributes, called dynamic attributes, provide information related to a part of the system or may cause an immediate action in its operation. A communication system object of type SPA (a process station), for instance, has the following configurable attributes:

Station address, SA =	123
Unit type, UT =	3
Reply timeout, RT =	15

The same SPA object can be accessed by the following dynamic attributes:

State, ST =	1
Update points, UP =	1
Send Message, SM =	"message"

3.6 Using system objects in SCIL

In SCIL, the system objects are accessed through their attributes. Using the object notations, which refer to attributes, the system objects can be supervised and controlled with SCIL programs. Changing the value of an attribute may cause an immediate reconfiguration.

In the object notation the attribute is specified with the object name, the object type and the attribute name. For example, the station address of a SPA object defined as SPA3 is accessed by the notation:

SPA3:SSA

where

'SA'	Is the attribute name. The S after the colon indicates that the object is a communication system object.

The object notations are discussed in details in Section 4.

Section 4 System object handling in SCIL

4.1 About this section

This section describes how to handle system objects in SCIL.

4.2 System object notation format

The system object notation has the following format:

name:{application}type{attribute}{index}

where

'name'	is an object name
'application'	is the logical application number
'type'	is the system object type, B or S
'attribute'	is an attribute name
'index'	is a single index or an index range

The parameters within curly brackets may be omitted. No space is allowed between the parameters in the object notation. The parameters are detailed below.

4.3 Name

System objects have predefined names. The name of a system object is composed of three predefined letters, A - Z, and an ordinal number, which is freely chosen. The three letters in the name specify which type the object represents, the number separates the object from others representing the same type. For example, SYS denotes base systems, APL applications, NOD nodes, PRI printers, and STA stations. APL3 denotes application 3, NOD4 node 4, STA5 station 5, etc.

The object number identifies the object within one application where the notation is used. The same physical object can be known under different object numbers within different applications. However, the object numbers of NOD objects must be unique within the entire SYS600 distributed system.

Alternatively, a freely chosen alias name may be given for a system object by specifying its BN attribute, see Section 5.

The predefined base system object names are listed and described in Section 5 and the communication system objects in Section 14.

4.4 Application

Application stands for a logical application number. The logical application number is the application number as known to the present application (according to the application mapping, the AP attribute, see Section 7). A prerequisite for using application number is that the applications recognize each others through the application mapping.

Generally, there is no need to include application number in the system object notation. Mostly, an application number is included when access to base system objects defined in another base system is desired. Application number is also needed when accessing communication system objects from an application, unknown (not defined) to the NET from which the objects are accessed.

4.5 Type

The type is a letter specifying the system object type as follows:

B	Base system objects
S	Communication system objects

4.6 Attribute

The attribute represents the value or feature to be read or written with the object notation. It is named by an attribute name, which is a combination of two letters, A ... Z. The attribute determines the data type (see the Programming Language SCIL manual) and the value of the entire object notation.

A communication system object notation must always contain an attribute. The base system object notation can be used without an attribute only when creating a new object.

4.7 Index

Indices are integer numbers, which are used together with some attributes. As a rule, the indices refer to the elements of an attribute of vector type. The actual attribute determines the data type of the elements.

An index or index range is marked in either of the following manners:

- With an integer number, either a positive integer value or an octal number.
- With an integer type expression embraced by brackets.
- With an interval (i .. j), where 'i' denotes the first index number and 'j' the last. Two points surrounded by brackets, (..), is interpreted as all the indices of the actual object notation. (i ..) means all indices larger than or equal to 'i', and (.. j) all indices less than or equal to 'j'.

In one case (the STAn:SME attribute), the index refers to an address. In this case the index cannot be given by an expression, it must be given as an octal number.

The indexing of the system object attributes is detailed in the attribute descriptions.

4.8 Using system object notations

System object notations can be used in SCIL statements and expressions. When used in expressions, the value of the attribute in the notation replaces the entire object notation. It can, for example, be part of a window definition expression, entailing that object data is shown in the window. It can be included in data object definitions or in conditional expressions, etc. See the Programming Language SCIL manual.

Some examples of system object notations:

STA3:SSA	The station address of the station STA3
NET1:SPO4	The protocol of line 4 of NET1
APL3:BAS	The application state of application

Section 5 Base system objects, overview

5.1 About this section

This section introduces the base system objects:

Section 5.2	Base system object types: An overview of the base system object types.
Section 5.3	Common naming attributes: Attributes common to all base system object types.
Section 5.4	Defining base system objects: The principles for defining base system objects, the start-up configuration, on-line configuration, attribute access, etc.

5.2 Base system object types

5.2.1 Base system object names

The base system object names consist of a predefined three-letter descriptive name and a number (n), which distinguishes units of the same type:

SYS	The base system itself. This object has no number, as there can be only one: the current base system.
APLn	Applications. 'n' = 1 ... 250
MONn	Monitors. 'n' = 1 ...100
PRIn	Printers. 'n' = 1 ... 20
LINn	Links: connections to adjacent nodes. 'n' = 1 ... 20
NODn	Nodes: base systems and communication modules. 'n' = 1 ... 250
STAn	Stations: Remote Terminal Units, PLCs, Protective Equipment, etc. 'n' = 0 ... 50000
STYn	Station type defining objects, 'n' = 1 ... 33

The base system recognizes the individual objects using the number 'n'. Henceforth, the number is referred to as object number. The base system objects and their interconnections are illustrated in Figure 1.

In addition to the naming convention above, a user-defined name may be given to any base system object by specifying its name attribute BN. See Section 5.3.

Below is a brief introduction to each of the object types.

5.2.2 SYS object

The SYS object corresponds to the actual base system. The SYS object attributes specify the base system properties, such as:

- Type of computer and operating system
- Station address and the node number of the base system
- Memory spaces and queue lengths
- Properties and addresses of connected devices

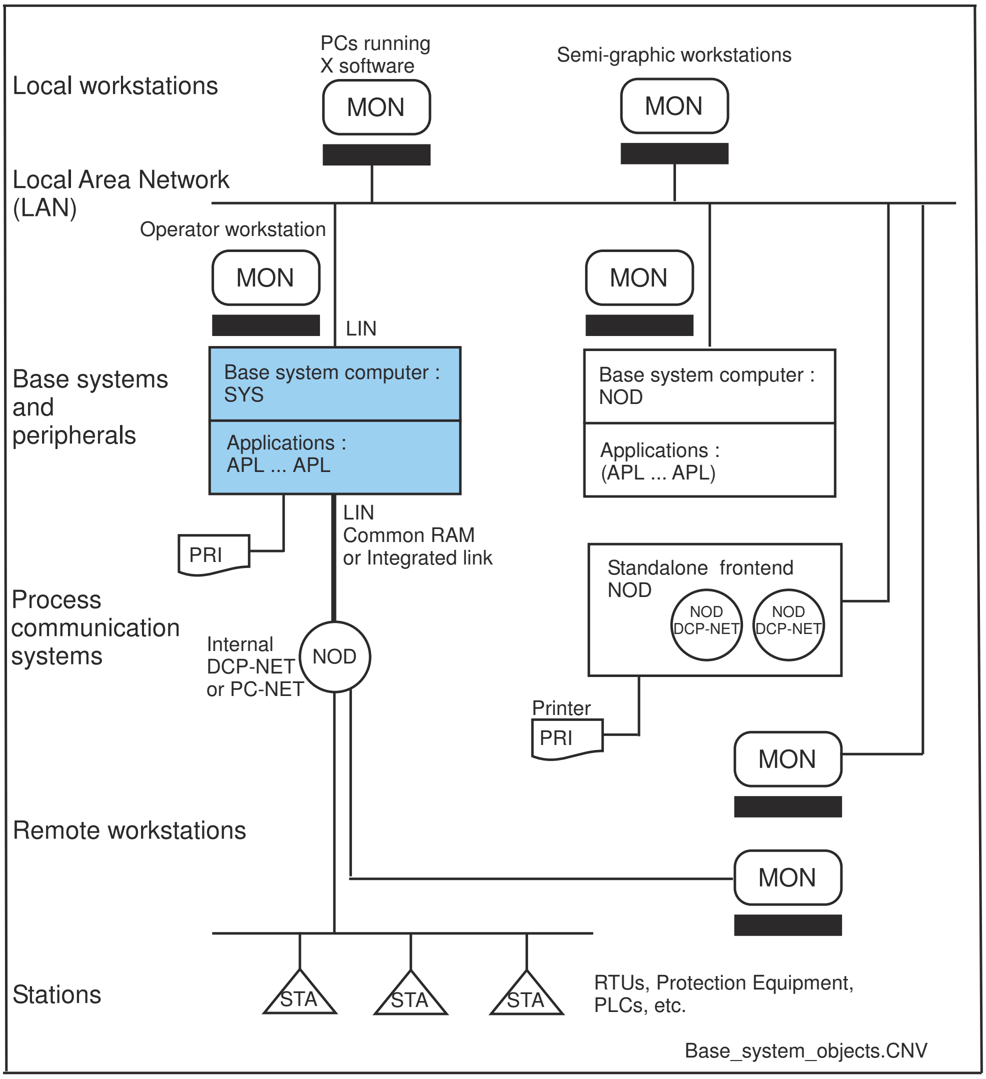


Figure 1: The base system objects defined in the shaded SYS600 base system

5.2.3 APL objects

Each application known to base system must be defined as an APL object. This comprises not only all applications situated within the base system in question, but also all those applications in other base systems that will communicate with the applications in the base system.

The APL object attributes specify the properties of the applications, such as:

- Application type: Local or external (in the same or in another base system)
- Application state: HOT, WARM or COLD
- Logical connections to peripherals and stations (device mapping)
- Logical connections with other applications within the same base system and in other base systems (application mapping)

5.2.4 MON objects

The MON objects correspond to the SYS600 monitors that the operator opens to view an application on a physical monitor screen. When an operator opens a SYS600 monitor, he reserves a MON object as a logical monitor.

5.2.5 PRI objects

Each printer used by the base system must be defined as a PRI object, including printers that are connected to NET units, LAN or other base systems. The PRI attributes specify the properties of the printers, such as:

- Printer type: Colour, black-and-white or transparent
- Connection type: Connected to a NET unit, to a base system or to a LAN
- Lines per page, header texts, etc.

5.2.6 LIN objects

The LIN objects define the links between the base system and adjacent nodes (base systems and communication units). The link object attributes specify the link properties, such as:

- Link type: Type of connection (Integrated Link or LAN)
- Communication properties: Redundancy check, time-out length, etc.
- Diagnostic counters, etc.

5.2.7 NOD objects

Each base system and NET unit known to the base system must be defined as a NOD object, including those which are not directly connected to the base system. The NOD attributes specify the addresses and properties of the nodes. The NOD object numbers are global and must be unique within the entire SYS600 network.

5.2.8 STA objects

The STA objects correspond to the stations known to the base system, when the term 'station' denotes process control units such as Remote Terminal Units (RTUs), Protective Equipment, Central Stations and protocol converters. The STA attributes specify the station properties that are relevant to the base system, for example:

- Type of station: S.P.I.D.E.R. RTUs, stations on ANSI lines, SPA units, P214, etc.
- Node number of the NET unit to which the station is connected and the device number of the station as known by that NET unit (system object number).

Each station that will be known to the base system must be defined as a STA object in the base system, unless it is of the STA default type connected to the default NET unit.

5.2.9 STY objects

By using the STY objects, the system engineer can define station types that are not predefined. A requirement is that the new station type can use an existing station interface and that the protocol has been implemented in NET unit.

5.3 Common naming attributes

The following three attributes are used to name a base system object and query the object type and number of a named base system object. Each base system object, regardless of its type, has these attributes. Consequently, they are described only once here.

5.3.1 BN Base System Object Name

User given name for the object.

Data type:	Text
Value:	Object name, up to 63 characters
Default value:	"" (no name)
Access:	No restrictions

The value of this attribute can be used in SCIL programs as an alias name for the object. The names must be unique, there cannot, for instance, be a STA and a NOD object by the same name. For STA, NOD and PRI objects, this name can also be used as an alias name of the corresponding communication system object.

5.3.2 BT Base System Object Type

The type of the object.

Data type:	Text keyword
Values:	"SYS", "APL", "STA", "STY", "MON", "PRI", "NOD" or "LIN"
Access:	Read-only, configurable

5.3.3 BM Base System Object Number

The object number of the object.

Data type:	Integer
Value:	0 ... 50 000
Access:	Read-only, configurable

Example:

The user wants to create a STA object by number 5 and name "PICCADILLY_STATION". The object can be created in either of the following ways:

```
#CREATE STA5:B = LIST(BN = "PICCADILLY_STATION") #CREATE
PICCADILLY_STATION:B = LIST(BT = "STA", BM = 5)
```

Now the object may be addressed by either STA5:B or PICCADILLY_STATION:B. Also, the corresponding communication system object STA5:S may be addressed by PICCADILLY_STATION:S.

The following statements are true:

```
STA5:BBN == "PICCADILLY_STATION" PICCADILLY_STATION:BBT == "STA"
PICCADILLY_STATION:BBM == 5
```

5.4 Defining base system objects

5.4.1 Principle

The base system objects are defined with SCIL using the SCIL command #CREATE (see the Programming Language SCIL manual).

The base system objects cannot be modified with #MODIFY nor deleted with #DELETE.

5.4.2 SYS_BASCON.COM

During start-up, the base system objects are created and their attributes are given values with the configuration file SYS_BASCON.COM. The main program at each base system start-up automatically executes this text file containing SCIL commands. If the file is missing or erroneous, the system cannot be started.

The base system configuration file SYS_BASCON.COM can be edited with a text editor both off-line and on-line. The modifications in SYS_BASCON.COM are taken into use when the system is started next time.

5.4.3 On-line configuration

With some restrictions, the base system configuration can be modified and extended during operation directly with SCIL. For example, it can be modified with command procedures started by the initial event channels APL_INIT_1 or APL_INIT_2 (see the Application Objects manual), or with the Base System Configuration tool. These online changes are not stored unless they are included in the base system configuration file SYS_BASCON.COM.

5.4.4 Required definitions

The base system object definitions required for various types of system set-ups are detailed in the System Configuration manual.

5.4.5 Attribute access

All attributes can be read with SCIL (the object notation), but all attributes cannot be written. The main access levels are:

- Read-only: The attributes that cannot be written in any circumstances or given values when the object is created with #CREATE.
- Read-only, configurable: The attributes that can be given values with the #CREATE command, but the value cannot be changed later with #SET.
- Not configurable, otherwise no restrictions: The attributes cannot be given values with #CREATE command, otherwise the attribute can be read and written without restrictions.
- Read, conditional write: Attributes that can be given values with #CREATE and which can be changed with #SET provided that the object is in off state (TT = "NONE").
- Write-only: The attribute cannot be read, only written.
- No restrictions: Attributes that can be both read and written without restrictions.

This terminology is used in the attribute descriptions in the subsequent sections.

Section 6 SYS objects for base system

6.1 About this section

This section describes the SYS objects and their attributes:

Section 6.2 General: the definition of SYS objects and the SYS object notation.

Section 6.3 Basic SYS attributes:

- Common naming attributes (BN, BT, BM)
- Basic Configuration Attributes (ND, NN, SA, WA)
- Hardware and Software Information (HW, OM, ON, OS, OV, PR, RD, RE, RP, RS)
- Communication Attributes (DN, DS, ER, TI)
- Time Handling Attributes (CT, TF, TM, TR, TS, TZ)
- Memory Handling Attributes (FS, ME, MF, MP, MS, MU, RC, RU)
- Global Paths and Representation Libraries (PH, RL)
- Security Attributes (HD, ID)
- Application Information (AL, PA)

Section 6.4 Software configuration attributes:

- Shadowing Attributes (SH)
- DDE Server Attributes (DE, DD, DU)
- OPC Server Attributes (OD, OP)

Section 6.5 Device attributes::

- Audio Alarm Device (AA, AD, AW)
- External Clock (CA, CD, CF, CL, CS)
- SPA Device Attributes (SD, SP)

Section 6.6 Operating system event handler attributes

Section 6.7 Miscellaneous attributes

6.2 General

6.2.1 SYS object definition

Each base system requires a corresponding SYS object definition. The SYS object must be defined as the first base system object in the SYS_BASCON.COM file. Otherwise, the base system cannot be started.

6.2.2 SYS object notation

The SYS object attributes are accessed from SCIL with the notation:

SYS:Bat

where

'at'	The attribute name

The SYS object attributes of another base system can be accessed with the notation:

SYS:mBat

where

'm'	Is the number of an application in the other base system as known to the present application (according to the application mapping, see Section 7).

An alternative way of using freely chosen object names is described in Section 5.

6.3 Basic SYS attributes

Common naming attributes BN, BT and BM are described in Section 5.

The SYS specific attributes are described in the following sections.

6.3.1 Basic configuration attributes

These attributes define the node number and address of the base system. ND, NN and SA are obligatory for all base systems.

6.3.1.1 ND Node Number

The node number of the base system computer. The number must be unique among all the base system computers and communication units connected to the network.

Data type:	Integer
Value:	0 ... 250
Default value:	9
Access:	No restrictions

6.3.1.2 NN Node Name

The LAN node name of the computer (host name). NN attribute may, for example, be used to identify the computers in a hot stand-by system.

Data type:	Text
Value:	Node name
Access:	Read-only

6.3.1.3 SA Station Address

The station address of the base system. The station address is used by the SYS600 internal protocol ACP. It is a number, which must be unique among all nodes, for example base systems, and communication units throughout the entire SYS600 network.

Data type:	Integer
Value:	1 ... 255
Default value:	192
Suggested value:	200 + node number
Access:	No restrictions

6.3.1.4 WA Web Address

The web address of the base system. The web address is used by web user interfaces connecting to the SYS600 base system.

Data type:	Text
Value:	Any text, up to 255 characters. Typically, an IP address.
Default value:	""
Access:	No restrictions

6.3.2 Hardware and software information

The base system computer hardware, the operating system or the SYS600 kernel settles these attributes. They are set automatically and cannot be configured or changed.

6.3.2.1 HW Hardware

The type of the base system computer.

Data type:	Text
Value:	"PC/AT"
Access:	Read-only

6.3.2.2 OM Operating System Minor Version

The minor version number of the operating system running in the base system computer.

Data type:	Integer	
Value:	0	Windows 2000, Windows Vista, Windows Server 2008 or Windows 10
	1	Windows XP, Windows 7, Windows 8 or Windows Server 2008 R2
	2	Windows Server 2003 or Windows Server 2012
Access:	Read-only	

See the OV attribute (Operating System Version) for more details of the version numbering.

6.3.2.3 ON Operating System Name

The name of the operating system running in the base system computer.

Data type:	Text
Value:	"Windows 2000"
	"Windows XP"

Table continues on next page

	"Windows Server 2003"	
	"Windows Vista"	
	"Windows Server 2008"	
	"Windows 7"	
	"Windows Server 2008 R2"	
	"Windows 8"	
	"Windows Server 2012"	
	"Windows 10"	
	"Windows Server 2016"	
	"Windows Server 2019"	
Access:	Read-only	

6.3.2.4 OS Operating System

The operating system running in the base system computer.

Data type:	Text	
Value:	"NT"	Windows
Access:	Read-only	

6.3.2.5 OV Operating System Version

The version number of the operating system running in the base system computer.

Data type:	Integer	
Value:	4	Windows 4.x
	5	Windows 2000, Windows XP or Windows Server 2003
	6	Windows Vista, Windows 7, Windows Server 2008, Windows 8 or Windows Server 2012
	10	Windows 10, Windows Server 2016 or Windows Server 2019
Access:	Read-only	

See also the OM attribute (Operating System Minor Version).

The following table summarizes the OV and OM values for current Windows versions that are running SYS600:

Version	OV	OM
NT 4.0	4	0
2000	5	0
XP	5	1
Server 2003	5	2
Vista	6	0
Server 2008	6	0
Windows 7	6	1
Server 2008 R2	6	1
Windows 8	6	2
Table continues on next page		

Version	OV	OM
Server 2012	6	2
Windows 10	10	0
Server 2016	10	0
Server 2019	10	0

To find the combined value of OV and OM as a text string, see the SCIL function OPS_NAME in the Programming Language SCIL manual.

The name of the running operating system is given by attribute ON, see above.

6.3.2.6 PR Product Name

The name of the product can be read from this attribute.

Data type:	Text
Value:	Product name
Access:	Read-only

6.3.2.7 RP Revision of the Product

The revision of the product can be read from this attribute.

Data type:	Text
Value:	Product revision
Access:	Read-only

6.3.2.8 RD Revision Date

Date of the running SYS600 kernel software build (hours, minutes and seconds are all zeroes).

Data type:	Time
Value:	Revision date
Access:	Read-only

6.3.2.9 RE Program Revision

The revision number of the running SYS600 kernel software, for example 9.2.

Data type:	Text
Value:	Program revision
Access:	Read-only

6.3.2.10 RS Revision State

The revision state of the running SYS600 kernel software.

Data type:	Text
Value:	State of running SYS600 kernel software build
	"DEVELOPMENT", "PRODUCTION", ...)
Access:	Read-only

6.3.3 Communication attributes

The following attributes specify the communication properties of the base system.

6.3.3.1 DN Default NET Node Number

The node number of the NET unit that will be regarded as default NET unit. The default NET unit must be directly connected to the base system or connected through LAN. A system object notation where the NET number is not explicitly mentioned, and which has no corresponding base system object, is sent to the default NET unit.

Recommendation: Set the DN attribute to the NET number to which most of the stations will be connected.

Data type:	Integer
Value:	1 ... 250
Default value:	1
Access:	No restrictions

Example:

The system object notation STA2:SIU is routed to the default NET unit, unless the base system contains a STAn:B object referring to the same station according to the device mapping. See the APL attributes in Section 7.

6.3.3.2 DS Default STA Type

The default value for the station type, the attribute STAn:BST, if not explicitly defined (see the STA attributes in Section 11).

Data type:	Text keyword	
Value:	"STA"	Stations using the Allen-Bradley ANSI X3.28 protocol
	"RTU"	S.P.I.D.E.R. RTUs
	"RCT"	PROCOL Station Interface
	"LCU"	Load Control Unit (Load Management)
	"SPA"	SPACOM
	"REX"	REx type relays
	"IEC"	Stations using International Electrotechnical Commission standard protocol
	"DNP"	Stations using Distributed Network Protocol
Default value:	"STA"	
Access:	No restrictions	

6.3.3.3 ER Enable Routing

Enabling and disabling message routing. The base system has the capability to route messages to other nodes, see the example in Figure 2.

If routing is enabled (ER = 1) and the base system receives a message with another destination address than the own station address, the message is routed to the destination node. If the destination address is unknown to the base system, the message is destroyed.

Data type:	Integer	
Value:	0	Routing disabled
	1	Routing enabled
Default value:	0	
Access:	No restrictions	

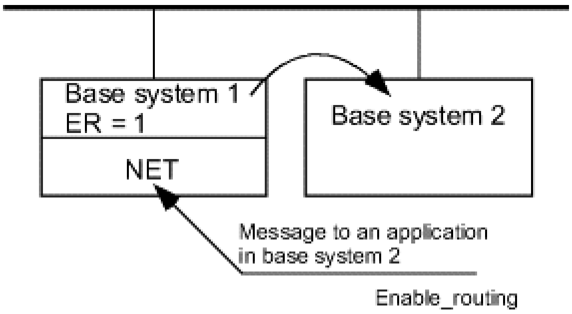


Figure 2: If routing is enabled in base system 1 it can route messages addressed to base system 2

6.3.3.4 TI Timeout Length

The maximum time the base system is waiting for a response after it has sent a message to a connected node. When this time has expired, an error status is produced. This timeout can be temporarily and locally changed by the SCIL function TIMEOUT (see the Programming Language SCIL manual).

Data type:	Integer
Value:	1 ... 65 535
Unit:	Seconds
Default value:	60
Access:	No restrictions

6.3.4 Time handling attributes

6.3.4.1 CT Current Time

Current date and time in one second resolution.

Data type:	Time
Value:	Current date/time in SYS time (defined by SYS:BTR, typically local time)
Access:	Read-only

OPC clients can use this attribute as a heartbeat signal of the base system. It may be subscribed to using update rate 0.

6.3.4.2 TF Time Format

The format of text representation of dates.

Data type:	Integer	
Value:	0	yy-mm-dd
	1	dd-mm-yy
	2	mm-dd-yy
Default value:	0	
Access:	No restrictions	

The TF attribute affects several time functions (such as TIME, TIMES, DATE and TIME_SCAN) and the #SET_TIME command, the Programming Language SCIL manual.

6.3.4.3 TM Time Master

The time master of synchronization of the base system time. The attribute specifies who is responsible for maintaining time zone information required to convert time values read from an external time source to the time reference of the base system.

Data type:	Text keyword
Values:	"APL" or "SYS"
Default value:	"APL" (for compatibility)
Suggested value:	"SYS"
Access:	No restrictions

If TM is set to "APL", the application must set SYS:BTZ (Time Zone) to match the current bias between the time reference of the base system and the time source (PC32 or PC-NET). SYS:BTZ is added to the time received from the external time source to calculate the system time. SYS:BTS (Time Season) is set according to PC31/PC32 clock, if present, otherwise it must be set by the application if used by the application.

If TM is set to "SYS", SYS:BTZ and SYS:BTS are not set nor used by the base system. SCIL function LOCAL_TIME_INFORMATION may be used to obtain time zone related information from the system. The time received from the external time source (if any) is used to set the fraction of semi-hour (minutes 0 ... 29, seconds and milliseconds) of the system clock only.

6.3.4.4 TR Time Reference

The time reference of the base system.

Data type:	Text keyword
Values:	"LOCAL" or "UTC"
Default value:	"LOCAL"
Access:	Read-only, configurable

When the base system runs in local time, all the time attributes of objects and time arguments and return values of functions operate in local time. When it runs in UTC time, they operate in UTC time.

The time reference of the system may be changed (SYS600 must be restarted, however). After the change, all the attributes in the databases are shown in the new time reference.

If TR is set to "UTC" and the base system is synchronized by PC-NET, the TM (Time Master) attribute should be set to "SYS" to enable different time references of the programs.

Before starting to run SYS600 in UTC time, the used application software, including COM 500 and LIBxx applications, must be checked for the support of this feature. Also, if the base system is connected to another SYS600 base system, they should both run in UTC time or the application software used to do APL-APL communication must be checked for the support of different time references of the base systems.

6.3.4.5 TS Time Season

The time season (summer time/winter time) of the PC31/32 clock, if such is used for synchronizing the base system time. The TS attribute is updated from the PC31/PC32 clock each time a valid time (valid according to the status register of the clock board) is read from the clock. The attribute can be set by SCIL if there is no external clock.

Data type:	Text keyword
Values:	"SUMMER", "WINTER" or "UNKNOWN"
Default value:	"UNKNOWN"
Access:	No restrictions

6.3.4.6 TZ Time Zone

Hours to add to the time given by an external time source to get the system time. See attribute TM for details.

Data type:	Real
Value:	-12.0 ... +12.0
Unit:	Hours
Default value:	0
Access:	No restrictions

6.3.5 Memory handling attributes

These attributes allow the programmer to check the free and used memory space, to specify memory flush, and to change the memory space reserved for application pictures and report objects in the main memory.

6.3.5.1 FS File Sync Criterion

Specifies in which situations the base system will force the operating system to flush (write) buffered file modifications out to disk. The purpose of the forced flushing is to guarantee file system integrity in case of a hardware or operating system software failure.

Data type:	Text keyword	
Value:	"NEVER "	No forced flushing. The operating system decides when the memory buffers are written on disk. This value should be used only in very fault tolerant systems.
	"MAINT "	Forced flushing is performed after each object maintenance (#CREATE, #MODIFY, #DELETE). It is also performed if any change of an attribute value causes a change in the structure of the file. For example, if a text attribute of an object grows in length so much that a new data block must be allocated to the file, the forced flushing is performed. This value guarantees (with a very high confidence) the integrity of files. However, the data contents of files may not be exactly up-to-date after a hardware failure (for example a power break).
	"SET "	Forced flushing is performed in the same situations as for "MAINT" and additionally after each #SET command (and DATA_STORE function) that affects the contents of a file.
	"CHECKPOINT "	Forced flushing in the same situations as for "SET" and additionally the report data files are flushed after each completed time channel execution.
	"ALWAYS "	Forced flushing after each data logging. This value may seriously degrade the performance of the report data logging.
Default value:	"CHECKPOINT"	
Access:	No restrictions	

This attribute is obsolete and likely to be removed in some future release. When a modern operating system and modern computer hardware is used, the value "NEVER" should always be used. The other values do not significantly improve fault tolerance, they just degrade the performance.

6.3.5.2 ME Memory Pool Supervision Enabled

Supervision of the global memory pool may be enabled/disabled by SCIL. This attribute is implemented to control the supervision.

Data type:	Integer	
Value:	0	Disabled
	1	Enabled
Default value:	1	
Access:	Read, write, configurable	

Setting ME to 1 always re-enables global memory pool events (even if the old value was also 1).

6.3.5.3 MF Memory Blocks Free

The number of free global memory blocks in each size class. The size of the blocks in each class is defined by the MS attribute, see below.

Data type:	Vector
Value:	Vector of 26 integer elements.
Indexing:	The entire vector is always read without index. Individual elements cannot be accessed.
Access:	Read-only

Example:

The total size of free global memory = SUM(SYS:BMF * SYS:BMS)

6.3.5.4 MP Memory Pool Sizes

The sizes of memory pools. The sizes can be changed in the SYS_CONFIG.PAR. For more information, see the System Configuration manual.

Data type:	List	
Value:	List value with following integer attributes:	
	GLOBAL	Size of the global memory pool
	PRIVATE	Maximum size of private memory pools
	PICO	Size of the local memory pool of monitor processes
	REPR	Size of the local memory pool of repr processes
	PRIN	Size of the local memory pool of prin processes
Unit:	Megabytes	
Access:	Read-only	

The attributes PICO, REPR and PRIN are reported for compatibility. Their value always equal to attribute PRIVATE.

6.3.5.5 MS Memory Block Size

The size of the memory blocks in bytes. The memory blocks are grouped into 26 size classes. This attribute contains the size of the memory blocks in each class.

MS is a vector of 26 elements, where each element represents a certain block size. MF (see above) tells how many blocks of each class are free. MU (see below) tells how many blocks of each class are used.

Data type:	Vector
Value:	Vector of 26 integer elements in the range 8 ... 268 435 456. Each element is the size of the memory blocks in the category.
Unit:	Bytes
Indexing:	The entire vector is always read without index. Individual elements cannot be accessed.
Access:	Read-only

6.3.5.6 MU Memory Blocks Used

The number of global memory blocks used in each size class, see MS above.

Data type:	Vector
Value:	Vector of 26 integer elements.
Indexing:	The entire vector is always read without index. Individual elements cannot be accessed.
Access:	Read-only

Example:

The total size of used global memory = SUM(SYS:BMU * SYS:BMS).

6.3.5.7 RC Report Cache Size

The maximum memory space allowed for the report cache, that is, the report database stored in RAM. When the occupied space in the report cache memory (the RU attribute) rises to this value, the oldest report objects are dropped out. Only report objects that are not running are kept in the report cache.

Data type:	Integer
Value:	0 ... 65 535
Unit:	Kilobytes
Default value:	3072 (3 MB)
Access:	No restrictions

6.3.5.8 RU Report Cache Used

The cache memory space used by report objects in the primary memory. When the RU attribute rises over the RC attribute, the oldest report objects are dropped out from the report cache.

Data type:	Integer
Value:	Integer
Unit:	Kilobytes
Access:	Read-only

6.3.6 Global paths and representation libraries

Base system object attributes PH and RL support system wide and application specific paths and representation libraries.

6.3.6.1 PH Global Paths

Defines the global paths of the system (paths common to all the applications).

Data type:	List
Value:	List of global paths
Access:	Read-only, configurable

The attribute names of the value define the path names and attribute values define the directories included in the path. The attribute value may either be a text value defining one directory or a text vector defining one or more directories.

The attribute can only be set by in SYS_BASCON.COM.

6.3.6.2 RL Representation Libraries

Defines the global representation libraries of the system (libraries that are common to all the applications).

Data type:	List
Value:	List of global representation libraries
Access:	Read-only, configurable

The attribute names of the value define the logical representation library names and attribute values define the library files included in the logical library. The attribute value may be either text value defining one file or a text vector defining one or more files.

The attribute can only be set in SYS_BASCON.COM.

6.3.7 Security attributes

The security attributes are used to configure and display the security settings of the base system.

6.3.7.1 HD System Hardening

Hardening settings of the base system.

Data type:	List
Value:	List value with the following attributes:
	REQUIRE_ENCRYPTED_ACP
	REQUIRE_KNOWN_ACP_CERTIFICATE
	REQUIRE_ACP_IP_WHITELISTING
	REQUIRE_OPC_AUTHENTICATION
	RUN_OPC_GUEST_AS_READ_ONLY
	DENY_EXTERNAL_API_ACCESS
	RUN_EXTERNAL_API_AS_READ_ONLY
Access:	Configurable in SYS_BASCON.COM, otherwise read-only

Attribute **REQUIRE_ENCRYPTED_ACP** takes one of three keyword values. Value "NONE" accepts all unsecure connections, value "NETWORK" (default value) allows unsecure connections from the local host and value "ALL" requires encryption from all connections. Note that even if the value is "NONE", unsecure connection is rejected from the node that has earlier used encrypted communication. When this attribute is set to "NETWORK" or "ALL", neither APL-APL communication (including mirroring) with a pre-9.4 SYS600 installation nor communication with an External Data Access Server running in another computer works. When the attribute is set to "ALL", communication with an External Data Access Client does not work at all. Regardless of the attribute value a secure communication is established between the nodes, if both nodes are capable to do so.

When attribute **REQUIRE_KNOWN_ACP_CERTIFICATE** is FALSE, the first connection from the node is accepted and its certificate is stored(learning mode, default value). When it is TRUE, a connection without a known certificate is not accepted.

When attribute **REQUIRE_ACP_IP_WHITELISTING** is TRUE (default value), incoming external TCP/IP connection requests are accepted only if the source IP address is explicitly listed in the whitelist. If not, the request is rejected and a warning message is written into the system log.

Communication from the base system to a non-whitelisted IP address is blocked as well. When the attribute is FALSE, ACP communication is allowed from and to any IP address. Whitelisting is node-specific and defined by the WL attribute of node objects (NODn:BWL).

When attribute **REQUIRE_OPC_AUTHENTICATION** is FALSE, an OPC client is accepted without authentication (OPC guest). When it is TRUE, authentication is required. If authentication is not required, attribute **RUN_OPC_GUEST_AS_READ_ONLY** defines the access rights of OPC guests. When its value is TRUE (default value), OPC guests do not have write access to system data. When the value is FALSE, OPC guests have full access to system data.

When **DENY_EXTERNAL_API_ACCESS** is TRUE, no external SCIL-API programs (for example SCILC.EXE) are allowed (default value). When it is FALSE, such programs are accepted. SCIL-API programs started by the base system itself are always allowed, however. If external SCIL-API programs are allowed, attribute **RUN_EXTERNAL_API_AS_READ_ONLY** defines the access rights of these programs. When its value is TRUE (default value), the programs do not have write access to system data. When the value is FALSE, full access to system data is granted for the programs.

In SYS_BASCON.COM, it is enough to specify only the exceptions to the default values, for example

```
HD = LIST(REQUIRE_KNOWN_ACP_CERTIFICATE = TRUE)
```

6.3.7.2 ID System Identity

The certificate of the base system in encrypted ACP communication.

Data type:	Byte string
Value:	The certificate
Access:	Read-only

6.3.8 Application information

The following attributes contain information about the configured applications in the system.

6.3.8.1 AL Application List

List of all local applications in the system and their state.

Data type:	Vector of list type elements, one for each local application.	
Value:	Attributes of each element:	
	AN	Application number, 0 ... 250
	NA	Name of the application
	SA	Web address of the shadowing partner computer
	AS	Application state ("COLD", "WARM" or "HOT")
	SS	Shadowing state ("NONE", "RECEIVE", "WARM_SEND" or "HOT_SEND")
	SP	Shadowing phase ("NONE", "TO_WARM_SD", "WARM_SD", "TO_HOT_SD", "HOT_SD", "TO_WARM_RC", "WARM_RC", "TO_HOT_RC" or "HOT_RC")

Table continues on next page

	IS_NEW	Value is TRUE, if the application is brand new, i.e. not yet logged-in by a user. If the application is not new, this attribute is totally omitted from the list.
	ANONYMOUS_LOGIN	Value is TRUE, if anonymous logins are enabled, otherwise FALSE
Access:	Read-only	

6.3.8.2 PA Primary Application

Defines the primary application of the system. The primary application works as the default application of the SCIL API interface and as the default application of the OPC Server name space.

Data type:	Integer
Value:	Application number, 1 ... 250
Default value:	The number of the application that becomes hot first at system start-up
Access:	Read-only, configurable

This attribute is useful especially in Hot Stand-by systems, where the main application is typically not started by SYS_BASCON.COM.

6.4 Software configuration attributes

6.4.1 Shadowing attributes

Shadowing is used in Hot Stand-by base system configurations. It means that all disk and real time database updates of the sending (hot) application are automatically copied to an identical receiving (cold) application. As a rule, the sending and receiving applications are in different base systems (base system computers). Shadowing is possible only between base systems connected via the same LAN. Mainly, shadowing is controlled by the APL base system objects, see the Shadowing attributes in Section 7.

6.4.1.1 SH Shadowing

The state of shadowing: is it in use or not in use.

Data type:	Integer	
Values:	0	No shadowing
	1	Shadowing
Default value:	0	
Access:	Read-only, configurable	

6.4.2 DDE Server attributes

The DDE server functionality, if used, enables external software applications (other than SYS600) to access the SYS600 applications within the base systems. The following attributes specify the DDE server functionality of the base system.

In operating systems before Windows Vista, there is one global DDE server in the base system. In Windows Vista, Windows Server 2008 and later, a DDE server serves one desktop (Windows session) only.

In pre-Vista operating systems, the DDE server attributes are truly global and the server may be started in SYS_BASCON.COM.

In Windows Vista and later, only the DDE server attributes of the own session are seen. The server must be started by executing a SET command within the session (#SET SYS:BDE = 1).

6.4.2.1 DE DDE Server Enabled

Enabling and disabling the DDE Server capability in SYS600. If disabled, applications within the base system cannot be accessed by DDE.

Data type:	Integer	
Value:	0	Disabled
	1	Enabled
Default value:	0	
Access:	Read-only, configurable (pre-Vista) Read-write (Vista and later), only value 1 accepted when written	

6.4.2.2 DD DDE Server Diagnostics

The values of the five diagnostic counters, that count certain events in the DDE server communication.

Data type:	Vector	
Value:	Vector of five integer elements. Each element contains the value of a counter.	
Indexing:	1 ... 5, counter number:	
	1	Number of currently open DDE conversations
	2	Cumulative value of opened conversations
	3	Cumulative value of successful request transactions
	4	Cumulative value of successful poke transactions
	5	Cumulative value of successful execute transactions
Access:	No restrictions	

6.4.2.3 DU DDE Server in Use

The usability of the DDE server.

Data type:	Integer	
Value:	0	Not in use
	1	In use
Access:	Read-only	

In Windows Vista and later, the value is 1 if the DDE server is started in the session.

In earlier operating systems, the value is 1 only if the server is started and there is a logged-in user in the base system computer.

6.4.3 OPC Server attributes

The OPC Server attributes are used to start, diagnose and stop the SYS600 OPC Data Access Server.

6.4.3.1 OD OPC Server Diagnostics

Diagnostic data for the SYS600 OPC Data Access Server.

Data type:	List		
Value:	List value with the following attributes:		
	START_TIME	Time	
	STATE	Text, one of the following keywords:	
		"NONE"	Not started
		"RUNNING"	Started and running
		"STOPPED"	Stopped
	CLIENTS	Vector of lists with the following attributes:	
		NAME	Text, the name given by the client itself
		NUMBER	Integer, the sequence number given by the server
		NETWORK_ADDRESS	Text, the network address of the client reported by RPC
		START_TIME	Time
		GROUP_COUNT	Integer, number of OPC groups in the client
		ITEM_COUNT	Integer, total number of OPC items in the groups
Access:	Read only		

6.4.3.2 OP OPC Server

The state of the OPC Data Access Server.

Data type:	Integer	
Value:	0	OPC Server not running
	1	OPC Server running
Default value:	0	
Access:	No restrictions	

Setting this attribute to 1 starts the SYS600 OPC Data Access Server. If set in SYS_BASCON.COM, the OPC Server is started immediately after the applications have been started.

Setting this attribute to 0 stops the running OPC Server.

The following errors may raise when setting the attribute:

SYST_OPC_SERVER_DISABLED (7292): OPC Server license has not been installed.

OPCS_SERVER_ALREADY_RUNNING (5510): Only one OPC Server is allowed. If this error occurs when opcs.exe is not running, the server has terminated abnormally. The attribute should be set to 0 first and then to 1 again.

OPCS_SERVER_START_TIMEOUT (5509): For an unknown reason, the server did not initialize in a reasonable amount of time.

6.5 Device attributes

These attributes describe the hardware of special equipment within or connected to the base system computer:

- Audio alarm device
- Radio clock
- SPA devices connected to a COM port of the base system computer

The attributes need not be defined in cases where the equipment is not in use or the default values are OK.

6.5.1 Audio alarm device

6.5.1.1 AA Audio Alarm Address

The use of a standard audio-visual alarm unit in PC base systems.

Data type:	Integer
Value:	0 = No audio alarm board
	Any other value = Audio alarm is in use
Default value:	0
Access:	Read-only, configurable

6.5.1.2 AD Audio Alarm Device

Defines the audio alarm card to be used.

Data type:	Text
Value:	"NONE", "FLYTECH FPC-046", "ADVANTECH PCI-1760", "NUDAQ PCI-7250" or "NUDAQ PCI-7256"
	For compatibility, "FLYTECH" is assumed, if AA attribute is set to 1 and AD is not defined.
Default value:	"NONE"
Access:	No restrictions

6.5.1.3 AW Audio Watchdog Cycle

The cycle of the keep-alive output signal to the audio alarm card.

The output is usually connected to a watchdog unit. Setting the value to 0 disables the signal and hence the watchdog functionality.

Data type:	Integer
Value:	0 ... 65 535
Unit:	Seconds
Default value:	3
Access:	No restrictions

6.5.2 External clock

6.5.2.1 CD External Clock Data

Reading and writing of external clock data. This attribute may be read even if the clock type (SYS:BCL) is "PC31", and set only when the clock type is "PC32" or "PCI510".

Data type: List

Value: List value with the following attributes:

TIME	Time
MILLISECONDS	Integer: 0 ... 999
DAY_OF_WEEK	Integer: 1 ... 7 (1 = Monday)
FREE	Integer: 1 = free running, 0 = time from the radio transmitter
DAYLIGHT_SAVING	Integer: 1= yes, 0 = no
SYNC_AFTER_RESET	Integer: 1= synchronized after last reset, 0 = not synchronized
TIME_CHANGE	Integer, time change warning:1 = yes, 0 = no
UTC	Integer, UTC: 1 = yes, 0 = no
LEAP_SECOND	Integer: 1 = yes, 0 = no
SOURCE	Integer, source of the time:0 = radio transmitter, 1 = serial port
INVALID	Integer: 1 = the clock's time is invalid, 0 = valid

For details of the status bit attributes, see the manual of the clock.

Access: Read, conditional write

When setting the clock, attributes missing in the list are left untouched (the clock is first read to preserve them).

The following attributes are ignored, when setting the clock:

MILLISECONDS	Its setting is not supported
DAY_OF_WEEK	It is calculated from TIME
FREE	Read only status bit
SYNC_AFTER_RESET	Read only status bit
UTC	Read only status bit
SOURCE	Read only status bit
INVALID	Read only status bit

This attribute is not returned by FETCH function nor can be set by #CREATE command.

When cyclically reading the clock, the data is considered valid if any of the statements listed apply:

FREE = 0

SYNC_AFTER_RESET = 1

SOURCE = 1

6.5.2.2 CF External Clock Read Frequency

The time period in seconds for periodically recurrent time synchronization of the operating system clock and the physical computer clock against the external clock (see the CL attribute). The attribute determines the time period between automatic clock synchronization. It also works as a momentary synchronization command when the attribute is set to 1.

Data type:	Integer
Value:	0 ... 65 535
	0 = no synchronization
	Setting the attribute to 1 does not affect the synchronization period, but causes an immediate synchronization once.
Unit:	Seconds
Default value:	0
Suggested value:	>= 60
Access:	No restrictions

6.5.2.3 CL External Clock

The type of the external clock, if any. The external clock synchronizes the SYS600 time. See also attributes TM, TZ and TS.

Data type:	Text keyword	
Value:	"NONE"	No external clock
	"PC31"	Radio clock of type PC31. The radio clock is synchronized in accordance with Central European radio time signals.
	"PC32"	PC clock card of type PC32
	"PCI510"	PC clock card of type PCI510
Default value:	"NONE"	
Access:	No restrictions	

6.5.2.4 CS Clock Status

The status register of an external clock. that provides additional clock information when a radio or satellite clock (for example PC31/PC32) is used for synchronizing the base system. Each time the status register of the clock is read by the base system, the value in the register (byte or word) is copied as such to the CS attribute. The meaning of the value is clock type specific.

Data type:	Integer
Value:	0 ... 255
Initial value:	0
Access:	Read-only

Status byte bit meanings:

		'1'	'0'
D0	Clock state	Free running	DCF77 controlled
D1	Daylight saving	Enabled	Disabled
Table continues on next page			

D2	Sync'ed since reset	Yes	No
D3	Dayl.sav. is going to change	Yes	No
D4	UTC time	Yes	No
D5	Leap second announced	Yes	No
D6	Board time from	Serial iface	DCF77
D7	On-board time invalid	Yes	No

6.5.3 SPA device attributes

The following attributes specify the connection of a SPACOM device directly to the base system computer through a COM port.

6.5.3.1 SD SPACOM Driver Name

The operating system device name of the SPACOM communication port when connected directly to the base system computer (not via NET unit). The attribute applies only to base systems with SP = 1, see below.

Data type:	Text
Value:	Max. 10 characters
Default value:	None
Access:	No restrictions

If COM port 10 or higher is used for SPACOM communication, the name must be given in the format "\\.\COM10". For lower port numbers, simple name will do (e.g. "COM5").

6.5.3.2 SP SPACOM Protocol

Specifies whether the direct SPACOM communication is allowed or not.

Data type:	Integer	
Value:	0	Direct connection of SPACOM not allowed
	1	Direct connection of SPACOM allowed
Default value:	0	
Access:	No restrictions	

6.6 Operating system event handler attributes

Operating System Event Handler is a Windows-based program, which passes Windows™ system events to SYS600.

The program is started when SYS600 service is started and it is stopped when SYS600 is stopped (shutdown).

6.6.1 OE Operating System Event Handler Enabled

Enabling the Operating System Event Handler.

Data type:	Integer
Value:	0 or 1
Default value:	0
Access:	Read-only, configurable

6.6.2 OT Operating System Event Handler Filter

Information about which event types are passed to SYS600. Before an application can receive events, attribute EE of the base system application object must be set to 1.

Data type:	Vector	
Value:	Vector(3), integer element values 0 ... 31	
	First element filters the application log information.	
	Second element filters the system log information	
	Third element filters the security log information	
	Values of the vector elements are bit masked the following way, thus (combined) values can be used masking on or off certain types of events:	
	ERROR	00000001 (1)
	WARNING	00000010 (2)
	INFORMATION	00000100 (4)
	AUDIT_SUCCESS	00001000 (8)
	AUDIT_FAILURE	00010000 (16)
Default value:	(0, 0, 0)	
Access:	No restrictions	

Example:

```
#SET SYS:BOT=(1, 3, 7)
;Errors are reported from the application log
;Errors and warnings are reported from the system log
;Errors, warnings and information are reported from
;the security log
```

6.7 Miscellaneous attributes

6.7.1 CX Comment Text

A freely chosen comment text.

Data type:	Text
Value:	Any Unicode text, up to 255 characters
Access:	No restrictions

6.7.2 SV System Variables

The attribute can be used as global variable. It is reserved for Hitachi Power Grids and should not be used in application programs.

Data type:	Vector
Value:	Defined by Hitachi Power Grids
Default value:	A vector of 10 integer 0’s
Access:	No restrictions

6.7.3 UV User Variables

The attribute can be used as global variables in application programs.

Data type:	Vector
Value:	Defined by the application
Default value:	A vector of 10 integer 0’s
Access:	No restrictions

Example:

The attribute works as a counter:

```
#SET SYS:BUV1 = SYS:BUV1 + 1
```

Section 7 APL objects for base system

7.1 About this section

This section describes the base system APL objects and their attributes:

Section 7.2	General: APL Objects and APL Object Notation.
Section 7.3	Basic APL attributes: Common naming attributes (BN, BT, BM), Basic Configuration (AN, AS, NA, TT), External and Alias Applications (ND, TN), Proxy Applications (HA, HS), Supervision Configuration (ME, QE), Paths, Representation Libraries and Text Databases (PH, RL, TD).
Section 7.4	Mapping attributes: Application Mapping (AP) and Device Mapping Attributes (MO, PR, ST).
Section 7.5	Shadowing attributes: SA, SC, SD, SF, SI, SL, SN, SP, SQ, SR, SS, SW, SY.
Section 7.6	Resource handling attributes: AA, AU, EM, EU, HB, PM, PQ, PU, RO, QD, QL, QM, QO, QP, QU.
Section 7.7	Application control attributes: Functional Definitions (AQ, CE, EE, HP, LS, OE, OI, OP, PF, PN, PP, SM, UA, WS), User Interface definitions (IT, LA, MS), Operation control (AC, AT, HT, LT, UC), SCIL Program Control Attributes (PS, RS), Revision Compatibility (RC).
Section 7.8	Application diagnostic attributes: DI, DT, DS.
Section 7.9	Mirroring attributes: Configuration attributes (EP, HE, IE, IS), Diagnostic attributes (HD, ID).
Section 7.10	Miscellaneous APL attributes: Self-diagnostics (AD), Comment (CX), Global Variables (SV, UV).

7.2 General

7.2.1 APL objects

Each application known to the base system must be defined as an APL object. This concerns not only all applications situated in the current base system (local applications), but also applications in other base systems (external applications) which are communicating with the local applications. An external application is defined with a reference to the application in the other base system.

At least one local application must be created with an application name (see the NA attribute) and set to "HOT" (see the AS attribute) in the SYS_BASCON.COM file.

7.2.2 APL object notation

The APL attributes are accessed from SCIL with the object notation:

APLn:Bat

where

'n'	The application number, 1 ... 250. If 'n' is omitted or n = 0, the object notation refers to the application where the notation is used.
'at'	Attribute name

The APL attributes of applications in another base system are accessed with the following object notation:

APLn:mBat

where

'm'	The logical application number (according to the application mapping, the AP attribute) of an external application (see the TT attribute).
'n'	The number of the application object in the other base system

An alternative way of using freely chosen object names is described in Section 5.

7.3 Basic APL attributes

Common naming attributes BN, BT and BM are described in Section 5.

The APL specific attributes are described in the following sections.

7.3.1 Basic configuration

The following attributes are set when defining the APL objects. When defining APL objects corresponding to local applications, all attributes are relevant. When defining APL objects corresponding to external applications, the AN and TT attributes are relevant.

7.3.1.1 AN Application Number

The base system object number (the 'n' in the object notation above) of the application. The number is defined when the APL object is created. This attribute is not included in FETCH (see the Programming Language SCIL manual).

Data type:	Integer
Value:	1 ... 250
Access:	Read-only, configurable

Example:

The object number of the current application is shown:

```
!SHOW AN APL:BAN
```

7.3.1.2 AS Application State

The state of a local application (an application within the same base system). The attribute determines whether the application is running (HOT), passive but available (WARM) or passive, not available (COLD). Setting an application to COLD performs a clean shutdown of the application.

A base system can contain several HOT applications.

Data type:	Text keyword	
Value:	"HOT"	The application is running. Its databases are stored in the primary memory.
	"WARM"	The application is not running, but the databases are loaded and accessible.

Table continues on next page

	"COLD"	The application is not running and not accessible, but it may receive file shadow input from another application.
Default value:	"COLD"	
Access:	No restrictions	

Example:

```
#SET APL4:BAS = "HOT"
```

When an application is set to "HOT", the event channels APL_INIT_1 and APL_INIT_2 are activated. When it is set to "COLD", the event channel APL_CLOSE is executed before the shutdown (see Application Objects manual).

7.3.1.3 NA Name

The name of the local application. The application name is the same as the name of the application directory branch used in the directory tree in the APL directory. The directory tree must exist when the application is created. The name is obligatory for all local applications.

Data type:	Text
Value:	Up to 10 ASCII letters or underscores
Access:	Read, conditional write

7.3.1.4 TT Translation Type

Determines how the base system regards the application and where the application is found, see example in Figure 3.

Data type:	Text keyword	
Value:	"NONE"	Off, the application is out of use
	"LOCAL"	Normal operating state
	"ALIAS"	The APL object number is translated to another APL object determined by the TN attribute
	"EXTERNAL"	The application is situated in another base system computer. The application is defined by the ND and TN attributes.
	"PROXY"	The application is a proxy application. See Section 5.
Default value:	"NONE"	
Access:	No restrictions	

Changing TT from normal operating state LOCAL to any other state will cause the application state to be set COLD.

7.3.2 External and alias applications

The following attributes must be defined for external applications, that is, applications located in another base system (TT = "EXTERNAL"). The attribute TN must be defined for alias applications (TT = "ALIAS").

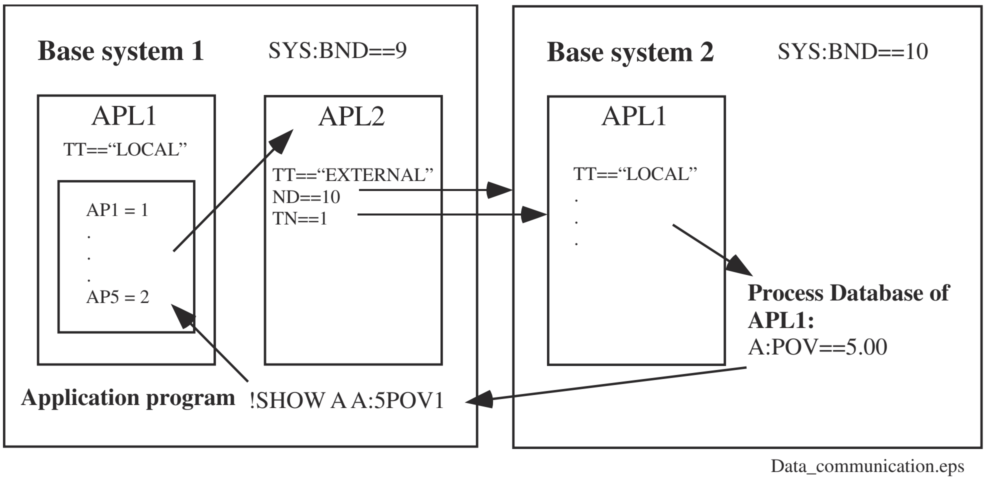


Figure 3: Communication between applications that are in different base systems

7.3.2.1 ND Node Number

When the application is located in another base system (TT = "EXTERNAL"), this attribute is the node number (the NOD object number) of the base system where it is found. The attribute is not applied for local applications (applications located in the same base system).

Data type:	Integer
Value:	0 ... 250
Default value:	No
Access:	Read, conditional write

Example:

The application that APL2 refers to is located in the base system whose node number is 10. See also Figure 3 and the TN attribute below.

```
APL2:BTT ==  "EXTERNAL", and
APL2:BND ==  10
```

7.3.2.2 TN Translated Object Number

Concerns external applications (TT = "EXTERNAL") and applications defined as an alias (TT = "ALIAS"). If the application is external, the attribute is the object number of the corresponding application in the other base system (defined by the ND attribute). If the application is an alias, the attribute is the translated object number.

Data type:	Integer
Value:	0 ... 250
Default value:	No
Access:	Read, conditional write

Examples:

1. Creating an alias application:

```
#CREATE APL5:B = LIST(TN = 2, TT = "ALIAS")
```

APL2 in the current base system refers to APL1 in the base system with node number 10. By using APL2 in application mapping (Section 7.4), communication is obtained with application 1 in the other base system.

2. Creating an external application:

```
#CREATE APL2:B = LIST(ND = 10, TN = 1, TT = "EXTERNAL")
```

7.3.3 Proxy application attributes

A proxy application is an application that represents an HSB application pair for other applications. An APL-APL communication request to a proxy application is routed to the hot application of the HSB pair.

When a request fails because of lost communication or non-hot application state, the base system starts polling the two applications. If a newly hot application is found within the communication timeout of the request, the currently hot application (attribute HA) is updated and the switch-over (or a short communication break) goes unnoticed by the issuer of the request. Otherwise, a SCIL_APL_APL_COMMUNICATION_TIMEOUT error is returned.

A proxy application is created by an application object of translation type "PROXY" (attribute TT) and defining the HSB application pair (attribute HS).

7.3.3.1 HA Hot Application

The number of currently hot application of the proxy.

Data type:	Integer
Value:	0 ... 250, application number 0 if neither is hot
Access:	Read-only

7.3.3.2 HS HSB Application Pair

The two true applications represented by the proxy.

Data type:	Vector of length 2
Value:	Two integers 0 ... 250, application numbers constituting an HSB pair
Access:	No restrictions

If the second application number is 0, the proxy represents only one application. This kind of configuration may be used when HSB is temporarily out of use or, in a non-redundant system, to take advantage of the recovery from short communication breaks.

Examples:

1. Creating a proxy application:

```
#CREATE APL10:B = LIST(TT = "PROXY", HS = (11, 12))
```

2. Reading data via the proxy

```
HOT_NODE_NAME = SYS:10BNN
SHADOWING_PHASE = APL:10BSP POSITION = BREAKERNAME:10POV10
```

7.3.4 Supervision configuration attributes

7.3.4.1 ME Memory Pool Supervision Enabled

Supervision of the local pools may be enabled/disabled by SCIL. This attribute is implemented to control the supervision.

Data type:	Integer	
Value:	0	Disabled
	1	Enabled
Default value:	1	
Access:	Read, write, configurable	

Setting ME to 1 always re-enables local memory pool events (even if the old value was also 1).

7.3.4.2 QE Queue Supervision Enabled

Supervision of application queues may be enabled/disabled by SCIL. This attribute is implemented to control the supervision.

Data type:	Integer	
Value:	0	Disabled
	1	Enabled
Default value:	1	
Access:	Read, write, configurable	

An application may disable the queue supervision temporarily when it causes an event burst by itself, for example at application start-up. Setting QE to 1 always re-enables queue overflow events (even if the old value was also 1).

7.3.5 Paths, representation libraries and text databases

Application object attributes PH, RL and TD define the application specific paths, representation libraries and text databases.

7.3.5.1 PH Paths

Defines the application specific paths.

Data type:	List
Value:	List of application specific paths
Access:	Read, conditional write

The attribute names of the value define the path names and attribute values define the directories included in the path. The attribute value may be either text value defining one directory or a text vector defining one or more directories.

The attribute can be set by the "#CREATE APLn:B" command (typically in SYS_BASCON.COM) or the "#SET APLn:BPH" command. However, the #SET command is not allowed after the initialization of the application has completed. It is allowed in command procedures executed by the event channels APL_INIT_1 and APL_INIT_H.

Example:

```
;The following line of code may be used in APL_INIT_1 to add path "XXXX"
to the APL:BPH #SET APL:BPH = MERGE_ATTRIBUTES(APL:BPH, LIST(XXXX = "/APL/
APLNAME/XXXX"))
```

7.3.5.2 RL Representation Libraries

Defines the application specific representation libraries.

Data type:	List
Value:	List of application specific representation libraries
Access:	Read, conditional write

The attribute names of the value define the logical representation library names and attribute values define the library files included in the logical library. The attribute value may be either text value defining one file or a text vector defining one or more files.

The attribute can be set by the "#CREATE APLn:B" command (typically in SYS_BASCON.COM) or the "#SET APLn:BRL" command. However, the #SET command is not allowed after the initialization of the application has completed. It is allowed in command procedures executed by the event channels APL_INIT_1 and APL_INIT_H.

Example:

```
#CREATE APL2:B=LIST(-
    RL = LIST(DEFAULT =  "c:\own_pirs\my_pir",-
                      "APL_/APL_STAND",-
                      "LAN_/LAN_STAND")),-
    ...  )
```

7.3.5.3 TD Text Databases

Defines the application specific text databases.

Data type:	Text vector of any length
Value:	Names of application specific text database files
Default value:	Empty vector
Access:	No restrictions The attribute must be written as a whole, individual elements may not be set one by one.

The database file names are listed in their logical search order, site specific files first and product specific files last.

The attribute can be set by the "#CREATE APL:B" command or the "#SET APLn:BTD" command regardless of the application state.

Example:

```
#CREATE APL2:B=LIST(-
    TD = VECTOR("APL_/SPECIAL_TEXTS.SDB",-
               "/LIB6/TEXTS/LIB666_TEXTS.SDB"),-
     ...  )
```

7.4 Mapping attributes

7.4.1 Application mapping

Application mapping is required if the application communicates with other SYS600 applications, in the same or other base systems.

7.4.1.1 AP Application Mapping

Enabling communication between different applications in the same or in the different base systems.

Data type:	Vector
Value:	A vector of 250 integers in the range 0 ... 250. Physical numbers of the connected applications. 0 = undefined.
Indexing:	1 ... 250. Logical application number
Default value:	All elements = 0
Access:	Read, conditional write

An application recognizes other applications by their logical application numbers. The AP attribute is a translation table between the logical application numbers and the corresponding physical application numbers (base system object numbers). The logical numbers are used as indices and the physical numbers are the value of the attribute.

The application gets access to the data bases of all applications that are known to it by a logical application number. The logical application numbers are used in the object notations after the colon when referring to an object in another application, see Figure 4.

If no intercommunication between different applications is needed, the attribute need not be set. The attribute is not needed for file shadowing.

Example:

The application knows application 2 (APL2) as application 5 (logical number). For example, the notation BREAKER:5POV refers to the process object BREAKER in application 2. See the Figure 4.

```
APL:BAP5 = = 2
```

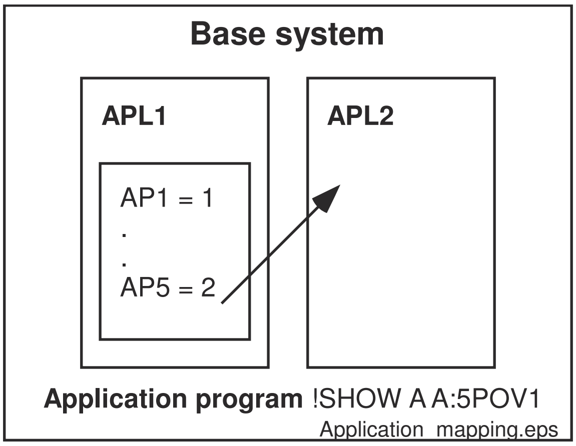


Figure 4: Application mapping between two applications

7.4.2 Device mapping attributes

The device mapping attributes define the mapping between the logical and physical device numbers for the MON, PRI and STA objects.

Logical device numbers are numbers used within an application in various tools and in SCIL object notation. For instance, the unit number given in the process object definition is a logical station number. Likewise, when accessing STA communication system objects (STA:S objects), the 'n' in the object notation is the logical number of the station.

The physical device number is the number of the corresponding base system object.

Using device (and application) mapping, complete standard applications may be designed for re-use in different base systems. The application may use fixed logical device numbers that are later mapped to the physical reality of the hosting base system.

The device mapping attributes are translation tables between logical and physical device numbers. See the example in Figure 5. The attributes are vectors where the logical device numbers work as indices and the element values give the corresponding physical device numbers.

For PRI and STA objects, it is recommended to use one to one device number mapping for simplicity, if there is no good reason to do otherwise. This is also the default for the ST and PR attributes.

For MON objects, value -1 is used instead of the physical device number, indicating that any physical monitor number will do.

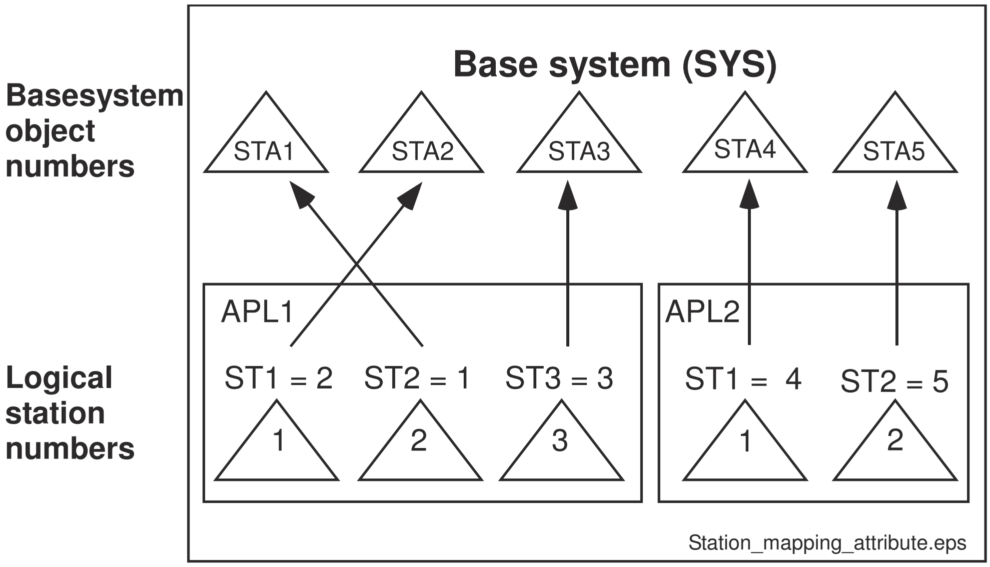


Figure 5: An illustration of the station mapping attribute (ST). The PR attribute works in a similar way.

7.4.2.1 MO Monitor Mapping

Mapping of logical monitor numbers to physical monitor numbers.

Data type:	Vector	
Value:	Vector of up to 100 integers in the range -1 ... 100	
	0	Not mapped
	-1	Any physical monitor number will do
	1 ... 100	Physical monitor numbers of monitors currently used
Indexing:	Logical monitor number, 1 ... 100	
Default value:	All elements = 0 (that is no monitor available for the application)	
Access:	Read, conditional write	

The attribute defines how many SYS600 monitors will be available for the application, and which are the logical numbers of monitors that the application may use. For each currently open monitor, it tells the mapped physical monitor number.

When a monitor is opened, any free physical monitor number is allocated for it. Physical monitor number -1 is configured in MO. During an open monitor session, the actual physical monitor number is shown in MO.

An open monitor may be unmapped by setting its MO index value to -1. This may be used (with care, of course) to close a monitor from another monitor.

Logical monitor numbers are used in

- process object attribute PD (Picture Devices)
- SCIL programs to distinguish and communicate between monitors

Example:

Application APL1 reserves 5 logical monitors:

```
#CREATE APL1:B = LIST(-
 .........
 MO = (-1,-1,-1,-1,-1), -
 .........)
```

7.4.2.2 PR Printer Mapping

Mapping of logical printer numbers to physical printer numbers.

Data type:	Vector
Value:	Up to 20 integers in the range 0 ... 20. Physical printer numbers.
Indexing:	Logical printer number, 1 ... 20
Default value:	Each element = the index (one-to-one mapping)
Access:	Read, conditional write

Logical printer numbers are used in

- printout commands #PRINT and #LIST
- process object attribute LD (List Devices)
- communication system object notation PRIn:S.

Because one-to-one mapping is the default, PR attribute does not always have to be specified.

7.4.2.3 ST Station Mapping

Mapping of logical station numbers to physical station numbers.

Data type:	Vector
Value:	Up to 50 001 integers in the range 0 ... 50 000. Physical station numbers
Indexing:	Logical station number, 0 ... 50 000
Default value:	Each element = the index (one-to-one mapping)
Access:	Read, conditional write

Logical station numbers are used

- as process object attribute UN (Unit)
- in communication system object notation STAn:S.

The translation of a STAn:S object notation is illustrated in Figure 6.

Because one-to-one mapping is the default, ST attribute does not always have to be specified.

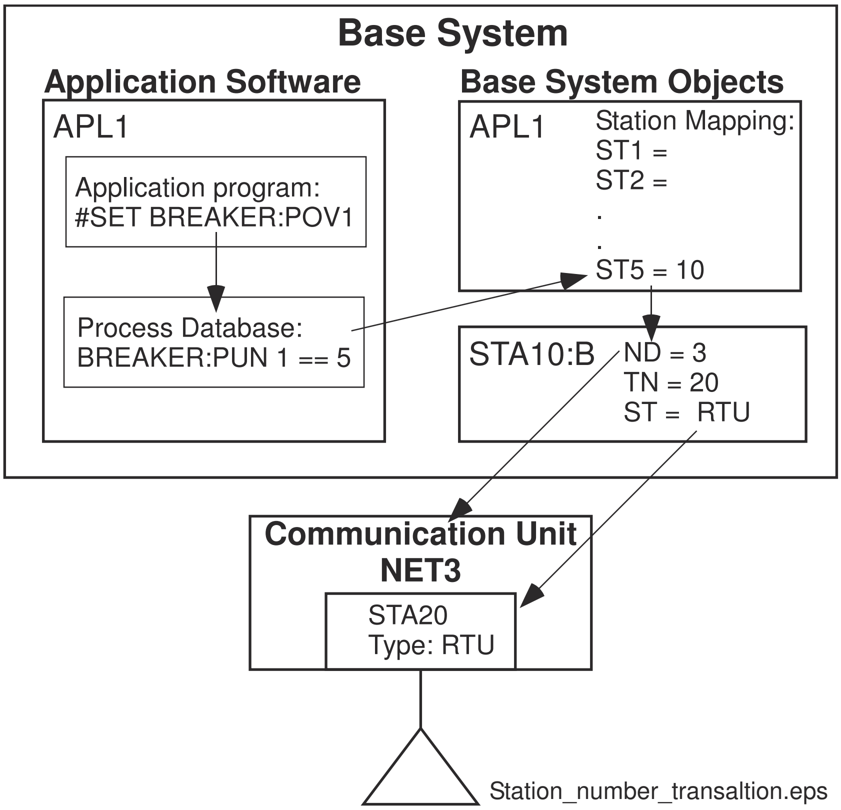


Figure 6: An example of the translation of station numbers

7.5 Shadowing attributes

Shadowing means that data from the hot application is copied to the stand-by application. Data is shadowed on event basis, which means that during run-time, only changed data items are shadowed.

7.5.1 SA Shadowing Partner Web Address

The web address of the SYS600 computer of the partner application.

Data type:	Text
Value:	Any text up to 255 characters, typically an IP address.
Access:	No restrictions

7.5.2 SC Shadowing Connection Time

The maximum waiting time that the primary application tries to get connection with the stand-by application after shadowing has been activated.

Data type:	Integer
Value:	0 ... 65 535
Unit:	Seconds
Default:	120 seconds
Access:	No restrictions

The connection for shadowing is established either by setting the APLn:BSS attribute to "HOT_SEND" state (provided that shadowing has been enabled using the SYS:BSH attribute), or by enabling shadowing with the SYS:BSH attribute (when APLn:BSS already is "HOT_SEND"). If no response is received within the time specified by the SC attribute, an error code is generated and no more trials are done.

7.5.3 SD Shadowing Diagnostic Counters

The value of the diagnostic counters on the connection between the hot and the stand-by base systems.

Data type:	Vector
Value:	Vector of 16 integer elements (elements 11 ... 16 not in use) Counter values
Indices:	Counter number, 1 ... 16. The attribute without index refers to the entire vector (all indices). An index range cannot be used when setting the attribute.
Access:	Read-only, the values can be resettable

There are 10 counters:

1. TRANSMITTED MESSAGES
2. TRANSMITTED COMMANDS
3. TRANSMITTED TRANSACTIONS
4. TRANSMITTED KILOBYTES
5. RECEIVED MESSAGES
6. RECEIVED COMMANDS
7. RECEIVED TRANSACTIONS
8. RECEIVED KILOBYTES
9. RAM DUMP TIME
10. FILE DUMP TIME

7.5.4 SF Shadowing Flush Time

The maximum time a message is buffered before it is flushed to the stand-by application.

Data type:	Integer
Value:	0 ... 65 535
Unit:	Milliseconds
Default value:	100 ms
Access:	No restrictions

The shadowing handling process copies all updates in files stored on disk under the application directory and all updates on RAM to the stand-by application. It tries to pack as much as possible in one copy transaction and meanwhile keeps the messages to be transferred in a buffer. A flush is performed when the message length arises over 60 kb or when the oldest message in the buffer is as old as specified by the SF attribute.

7.5.5 SI Shadowing Diagnostic Interval

The time between the diagnostic commands which the running (HOT and SEND) application sends to the stand-by application (if no transaction messages are sent).

Data type:	Integer
Value:	0 ... 65 535
Unit:	Milliseconds
Default:	The value of the SF attribute
Access:	No restrictions

7.5.6 SL Shadow Dump Slowdown

Shadowing dump phase control. In some applications, the file dump phase of shadowing start-up might degrade the system disk I/O performance unacceptably. Particularly, this might happen if shadowing was used within one system to build a file database copy for backup purposes. Using the SL attribute the file dump data transfer can be slowed down. SL specifies the slow down time in milliseconds per kilobytes of transferred data.

Data type:	Integer
Value:	0 ... 65 535
	The slow down time in milliseconds per kilobytes of transferred data
Default:	0
Access:	No restrictions

Setting this attribute to a non-zero value slows down the dump phase drastically. The attribute is more or less obsolete with today's operating systems and hardware.

7.5.7 SN Shadowing Partner Number

The logical application number of the shadowing partner application (usually an external application).

Data type:	Integer
Value:	0 ... 250
	Logical application number
Access:	Read-only, configurable
Example:	See the example of the SS attribute below

When the application is the sending part in a shadowing relation (see the SS attribute below), the SN attribute is the number of the receiving application. When the application is the receiving part, the SN attribute is the number of the sending application.

In a mirroring configuration, the attribute is used as follows:

If a host application is an HSB application, both partners are defined as external applications in the image system and their SN attributes are set to point to each other. This enables the mirroring software to automatically switch communication to the host that becomes hot.

If an image application is an HSB application, both partners are defined as external applications in the host system and their SN attributes are set to point to each other. This enables the mirroring software to listen to the two image applications and communicate with the one that is currently hot.

7.5.8 SP Shadowing Phase

The shadowing phase in progress.

Data type:	Text keyword	
Value:	"NONE"	No shadowing
	"HOT_RC"	Normal operation of the stand-by application in Hot Stand-by systems
	"HOT_SD"	Normal operation of the primary application in hot stand-by systems
	"TO_HOT_SD"	On-going file dump or RAM dump (primary application)
	"TO_HOT_RC"	On-going file dump or RAM dump (stand-by application)
	"WARM_RC"	Normal operation of the stand-by application in warm stand-by systems (file shadowing)
	"WARM_SD"	Normal operation of the primary application in warm stand-by systems
	"TO_WARM_RC"	On-going file dump or RAM dump (stand-by application) in warm stand-by systems
	"TO_WARM_SD"	On-going file dump or RAM dump (primary application) in warm stand by systems
Access:	Read-only	

7.5.9 SQ Shadow the Event Channel Queue

Enabling and disabling the shadowing of the event channel queue in hot stand-by applications.

Data type:	Integer	
Value:	0	Event channel shadowing disabled
	1	Event channel shadowing enabled
Default:	0	
Access:	No restrictions	

When a process object is updated, it may activate an event channel for post-processing the event by command procedures and data objects. The event channel activation requests are queued and executed in turn. In a hot stand-by environment, it is possible that a take-over occurs after a process object has been updated but before the activated event channel has been executed. If the event channel queue has not been shadowed, the post-processing of the event is lost.

If event channel shadowing is enabled (SQ = 1), the mechanism is as follows:

During shadowing, the event channel activation requests caused by process object updates are transferred to the receiving application. Likewise are indications of completed event channels transferred. (An event channel is completed when the connected data object, command procedure or time channel has been executed). After a take-over, the event channel queue is reconstructed in the running application and the event channels are executed after APL_INIT_H. The events are queued in the original order. If there were several pending activations for the same object, only the last one is re-queued. The snapshot variables of the process object are not shadowed but reconstructed from the current values in the running application. Event channel activations generated by SCIL are not shadowed.

The event channel shadowing guarantees that each event channel activation is executed in at least one of the HSB application pairs. However, there is a slight risk that an event channel is executed twice (in both applications).

7.5.10 SR Shadowing Receive Timeout

The timeout of the Hot Stand-By connection.

Data type:	Integer
Value:	0 ... 65 535
Unit:	Seconds
Default value:	5 seconds
Access:	No restrictions

The hot stand-by connection is considered broken if no response is received when this time has elapsed since a message (or a diagnostic command) has been sent from the primary system to the stand-by system. Likewise, the connection is considered broken if the stand-by application does not receive any diagnostic command when the diagnostic interval (the SI attribute) in addition the time specified by the SR attribute has elapsed.

7.5.11 SS Shadowing State

The role of the application in the shadowing relation.

Data type:	Text keyword
Value:	"NONE"
	"RECEIVE"

Table continues on next page

	"HOT_SEND"	File + RAM shadowing
	"WARM_SEND"	File shadowing
Default value:	"NONE"	
Access:	Read and write, not configurable	

This attribute determines whether the application acts as the sending or receiving application in a shadowing relation. The shadowing pair is defined with the APLn:BSN attribute. During operation the sending application should be in "HOT" state and the receiving one in "COLD" state.

Example:

The current application APL1 sends file and RAM shadowing to application defined by the APL:BSN attribute:

```
APL:BSS  ==  "HOT_SEND"
APL:BSN  ==  10
```

7.5.12 SW Shadowing Watchdog

The logical application number of the watchdog application.

Data type:	Integer
Value:	0 ... 250
Access:	Read-only, configurable

7.5.13 SY Time Synchronization Interval

The time between time synchronization signals sent from the primary application to the stand-by application.

Data type:	Integer
Value:	0 ... 65 535
Unit:	Seconds
Access:	No restrictions

7.6 Resource handling attributes

7.6.1 AA APL-APL Server Count

The number of APL-APL communication servers that handles the APL-APL communication.

Data type:	Integer
Value:	0 ... 10
	0 = no processes for APL-APL communication
Default:	1
Access:	Read-only, configurable

If there is only one process per application to serve incoming APL-APL communication requests and if a request takes a long time to satisfy, other requests will have to wait. A request can take a long time to satisfy if it contains process communication, which might time out. The situation might lead to timeouts and/or slow communication. Hence, in applications that receive APL-APL communication requests from more than one base system, the APL:BAA attribute should be given a larger value for better throughput.

Attribute AU below may be used to analyse the communication and help finding an adequate value for AA attribute.

Setting APL:BAA to 0 disables APL-APL communication with the application. This may be used for security reasons.

7.6.2 AU APL-APL Server Queue Used

The count of pending incoming APL-APL communication requests.

Data type:	Integer
Access:	Read-only

This attribute may be used to supervise the APL-APL communication and analyse its problems. In a normal situation, this value should be zero indicating that all the incoming requests are served immediately. If it shows positive values, the value of AA attribute (see above) may be too small. If it shows growing values or stays positive for a long time, the external SCIL application(s) sending the requests should be checked.

7.6.3 EM Event Queue Length Maximum

The maximum number of process events that will be queued for event channel activation or for mirroring communication.

Data type:	Integer
Value:	1 ... 65 535
Default value:	500
Access:	No restrictions

If the application is local, the attribute specifies the maximum length of the event channel input queue. While the queue is full, the base system does not accept process messages from the process communication units (for example PC-NET's, mirroring clients and External OPC Clients). The behavior is specified for not losing events in any circumstances (However, if the event buffers in those units or process stations behind them do overflow, events will be eventually lost).

If the application is external, the attribute specifies the maximum length of the mirroring event queue. The behavior in an overflow situation is specified by attribute EP, see Section 7.9.

7.6.4 EU Event Queue Used

The current used length of the event channel input queue (if a local application) or the mirroring event queue (if an external application). See the EM attribute.

Data type:	Integer
Value:	0 ... the current value of the EM attribute
Access:	Read-only

7.6.5 HB History Buffer Length

The size of the history buffer, that is maximum number of process object history registrations.

Data type:	Integer
Value:	0 ... 65 535
Default value:	0
Access:	No restrictions

7.6.6 PM Printer Spooler Queue Length Maximum

The maximum number of printout commands in the printout queues.

Data type:	Vector	
Value:	Vector of two integers in the range 1 ... 65 535	
Indexing:	1	Process printouts
	2	Report printouts
Default value:	(500, 500)	
Access:	No restrictions	

When this number is full, no new print-out commands are handled. The function of the system is delayed. The limit protects the system from being overloaded. SYS600 printer despooler (PRNC) also supervises the length of the print queue of the connected printer and waits while the length is more than 100 jobs.

7.6.7 PQ Parallel Queues

The maximal number of parallel report queues that the application can use.

Data type:	Integer
Value:	0 ... 30
Default value:	0
Access:	Read, conditional write, only in "COLD" state
Example:	See QP

7.6.8 PU Printer Spooler Queue Length Used

The present number of printout commands waiting for execution, see the PM attribute.

Data type:	Vector	
Value:	Vector of two integers	
Indexing:	1	Process printouts
	2	Report printouts
Access:	Read-only	

7.6.9 QD Queue Dedication

Defines whether parallel queues are dedicated or not.

Data type:	Vector	
Value:	Vector of up to 30 integer values	
	0	Not dedicated
	1	Dedicated
Indexing:	Parallel queue number	
Default:	0	
Access:	Read-only, configurable	
Example:	See QP	

A dedicated parallel queue is a queue that does not execute report objects whose PQ attribute is 0.

If all parallel queues are dedicated, a report object with PQ=0 (and PE=1) is run in event channel queue.

7.6.10 QL Process Query Length

Maximum length of process database query performed with the function PROD_QUERY, see the Programming Language SCIL manual.

Data type:	Integer
Value:	0 ... 2 000 000. Number of process objects or events.
Default value:	1 000
Access:	No restrictions

7.6.11 QM Queue Maximum

The maximum length of queues.

Data type:	Vector	
Value:	A vector of four integers in the range 1 ... 65 535	
Indexing:	1	The time channel queue
	2	The event channel queue
	3	The parallel queues
	4	The delayed execution queue (#EXEC_AFTER)
Default value:	(1000,1000,1000,1000)	
Access:	No restrictions	

This attribute is designed to protect the SYS600 system against erroneous SCIL applications. For example, a command procedure MYSELF that only requeues itself twice (two #EXEC MYSELF:C commands) would very quickly eat all the system resources and freeze or crash the whole program. When the limit specified by the attribute is reached, #EXEC commands will fail by error code REPF_EXECUTION_QUEUE_FULL (1118).

However, the length of the queues 2 and 3 may exceed the limit specified by the attribute because of process events. The reasoning behind this is as follows:

The rate of incoming process events is supervised by the EU and EM attributes (as well as PU and PM) attributes of the application.

When a process event is accepted into the process database (i.e. EU < EM and PU < PM), the attribute EU is incremented, the value from the process is stored in the database and, among other things, its event channel is activated. The event channel may contain one or more command procedure and data object activations. These objects are queued for execution in the event channel queue (queue 2), or in one of the parallel queues. In this situation, the queue length maximums QM(2) and QM(3) are not honoured, because otherwise events would be lost. When all the objects of the event channel have been executed, the EU attribute is decremented to make the process database accept a new process event.

So, as an example, if each process message activates 5 parallel command procedures (the primary object of the event channel plus 4 secondaries) and the value of the EM attribute is 500, the length of parallel queues (QU(3) is likely to be about 2500 in a rush situation, regardless of the value of QM(3).

7.6.12 QO Queued Objects

The names of objects in various system queues queued for execution, but not yet started (c.f. attribute RO).

Data type:	Vector	
Value:	Vector of 34 elements	
Indexing:	1	The time channel queue.
	2	The event channel queue.
	3 to 17	The parallel queues 1 to 15.
	18	The objects queued for parallel execution, but not yet assigned to any particular parallel queue.
	19	The objects to be executed in future, i.e. activated by #EXEC_AFTER.
	20 to 34	The parallel queues 16 to 30.
Access:	Read only	

If a queue is empty, the corresponding element of the attribute is an empty vector, otherwise it is a text vector listing the queued objects. An object is identified by a text string starting with character 'D' (for data object), 'C' (for command procedure), 'T' (for time channel object), 'A' (for event channel object) or 'E' (for event object) followed by the name of object. In case of an event object, the index of the object enclosed in parentheses follows.

This attribute is used for application debugging.

7.6.13 QP Queue Priority

The priority of the parallel queue.

Data type:	Vector
Value:	Vector of up to 30 text keyword values ("NORMAL", "LOW" or "HIGH")
Indexing:	Parallel queue number
Default:	"NORMAL"
Access:	Read-only, configurable

The priority of the queue defines how the operating system is to allocate processor time for the queue compared to the time channel queue, event channel queue and other parallel queues. There are three priority classes: low, normal and high.

The priority of time channel and event channel queue is fixed (always 'normal').

Example:

```
CREATE APL1:B = LIST(NA = "NAME", -
... -
PQ = 3, -
QP = ("NORMAL" , "NORMAL" , "HIGH"), -
QD = ( 0 , 0 , 1))
```

An application is created with 3 parallel queues. The queues 1 and 2 run with normal priority and serve report objects with PQ = 0. The queue 3 runs with a higher priority and only accepts report objects, whose PQ attribute is 3.

7.6.14 QU Queue Used

The number of waiting executions in queues.

Data type:	Vector	
Value:	A vector of four integers	
Indexing:	1	The time channel queue
	2	The event queue
	3	The parallel queues
	4	The delayed execution queue (#EXEC_AFTER)
Access:	Read-only	

7.6.15 RO Running Objects

The names of the data objects or command procedures currently under execution in REPR queues.

Data type:	Vector	
Value:	Vector of max. 32 text elements	
Indexing:	1	The time channel queue
	2	The event channel queue and
	3 to 32	The parallel queues 1 to 30
Access:	Read only	

If a queue is empty, the corresponding element of the attribute is an empty text string, otherwise it is a text string starting with character 'D' (for data object) or 'C' (for command procedure) followed by the name of object.

This attribute is used for application debugging.

7.7 Application control attributes

7.7.1 Functional definitions

The following attributes specify some functions of the application.

7.7.1.1 AQ Alarm Picture Queue Handling

Specifies the alarm picture queue handling, and how pictures are removed from the monitor alarm picture queues.

Data type:	Text keyword	
Value:	"BY_MON"	The queue is maintained on SYS600 monitor basis. An alarm picture is not removed from the queue until it has been displayed on the monitor. This is the default.
	"BY_APL"	The queue is maintained on application basis. When an alarm picture is displayed on a SYS600 monitor, it is removed from the alarm picture queues of all monitors belonging to the application.
	"BY_OBJ"	The same as "BY_APL", but additionally the alarm picture is removed from all alarm picture queues when all alarming objects in the picture have been acknowledged (regardless of how).
Default value:	"BY_MON"	
Access:	No restrictions	

7.7.1.2 CE CAM Enabled

Tells whether CAM (Centralized Account Management) is used for user authentication or not.

Data type:	Integer	
Value:	0	CAM is not used. The local user account management (UAM) is used to authenticate users.
	1	CAM is used to authenticate users.
	2	CAM is used to authenticate users with custom role integration.
Default:	0	
Access:	Can be set only when the application is cold, otherwise no restrictions	

Area of Responsibility (AoR) cannot be used when CAM is in use. SYS600 Base System automatically disables AoR if also CAM is enabled. Warning message about such configuration is logged to SYS600 Notify.

When SYS600 application attribute CE is set to value 2, custom roles defined in SDM600 can be used in SYS600. Each custom role definition has a unique integer identifier in both systems. Typically, these identifiers are negative integer values. To match custom role defined in SDM600 and SYS600 role, these identifiers must be identical in both systems.

Reserved role identifiers which cannot be used in SYS600:

-100 Administrator

-101 OPC Guest

-102 Anonymous

Example: A custom role "110kV Operator" is created in SDM600 with integer identifier -1. The same custom role must be created in SYS600 user management tool using the same integer identifier. Custom roles are matched using only integer identifiers. The role names may differ.

7.7.1.3 EE System Event Enabled

Specifies whether system events (event channel SYS_EVENT) are received by this application or not.

Data type:	Integer	
Value:	0	System events are not received.
	1	System events are received.
Default:	0	
Access:	No restrictions	

There may be several applications in the system with EE = 1. When a system event occurs, the event channel is activated in all these applications. The applications should be programmed to co-ordinate the actions taken as a consequence of the event.

7.7.1.4 HP History Logging Policy

Determines storing the history of events. The alternatives are history database, event logging and history buffer, or neither of them. For more information on how to configure storing the event history, see the System Configuration manual.

Data type:	Text keyword	
Value:	"DATABASE"	Selects the new database scheme
	"EVENT_LOG"	Selects the old event log scheme
	"NONE"	No event logging is done
Default value:	"EVENT_LOG"	
Suggested value:	"DATABASE" for new applications	
Access:	No restrictions	

HP may be set by "#CREATE APLn:B" command and by #SET command when the application is cold and its shadowing state (APL:BSS) is "NONE".

If HP = "DATABASE", attribute HB (History Buffer length) does not have any meaning.

When HP = "DATABASE", the application attribute HT is incremented every time an event is written into the history database.

7.7.1.5 LS Languages Supported

The languages supported by the application.

Data type:	Text vector
Value:	Two-letter ISO 639 standard language names of the supported languages
Default:	Empty vector
Access:	No restrictions

7.7.1.6 OE OPC A&E Server Enabled

Informs whether the application has an OPC Alarms and Events Server.

Data type:	Integer	
Value:	0	OPC A&E Server is not running
	1	OPC A&E Server is running
Default:	0	
Access:	No restrictions	

Setting the attribute to 1 starts the server, setting to 0 stops it.

7.7.1.7 OI Object Identifier Definition

Defines the Object Identifier (OI) structure used by the process objects of the application.

Data type:	List		
Value:	The following attributes:		
	DEPTH	Number of levels in the OI structure, 0 ... 5.	
	LEVELS	Vector of length DEPTH. The elements are lists of the following attributes:	
		WIDTH	Width of the corresponding OI field
		NAME	Name of the level, an identifier
		TITLE	Description of the level (localizable text)
Default:	See below		
Access:	Read-only, configurable		

The sum of WIDTHs may not exceed 63.

If the attribute is not defined in SYS_BASCON.COM, it is, for compatibility reasons, automatically constructed from APL:BSV(15) when the application is started. If APL:BSV(15) is not defined, the default value is used.

Example:

The default value of the attribute is

```
LIST(DEPTH = 3,-
  LEVELS = VECTOR(LIST(WIDTH = 10, NAME = "STA", TITLE = "Substation"),-
          LIST(WIDTH = 15, NAME = "BAY", TITLE = "Bay"),-
          LIST(WIDTH =  5, NAME = "DEV", TITLE = "Device"))
```

7.7.1.8 OP OPC A&E Server Data

Configuration data for the OPC Alarms and Events Server of the application.

Data type:	List	
Value:	The following attribute:	
	DE	Text, the delimiter character(s) used in hierarchical event source names. If DE is not given, default delimiters ". / \ :" are applied.
Default:	Empty list	
Access:	No restrictions	

Setting this attribute has no immediate effect on the running OPC A&E Server. To apply the changes, restart the server (by setting the OE attribute to 0 and back to 1).

7.7.1.9 PF Post-processing Policy for Object Status 1

This attribute specifies how the incoming status 1 from stations is treated in the process database.

The status may be stored in the database as a FAULTY status 1, or it may be changed to 2 (OBSOLETE_STATUS, or INVALID).

Data type:	Text keyword	
Value:	"DEFAULT"	The status is stored as 1.
	"REGARD_AS_2"	The status is changed to 2.
Default value:	"DEFAULT"	
Access:	No restrictions	

This attribute has been implemented to prevent excessive events and alarms from gateway applications that report some communication problems by status 1.

When an incoming status 1 is received from a station, the policy of the corresponding station object (attribute STA:BPF, see Section 11) is first applied. Only if STA:BPF = "DEFAULT", or the unit number (UN) of the object is 0, the policy defined by this APL:BPF attribute is applied.

7.7.1.10 PN External Database Port Numbers

Defines the port numbers used by external sql databases. These values are not used by SYS600 base system.

This attribute is useful if the default port (5432) is already reserved in the local machine or in the Hot Stand-by shadowing partner machine.

Data type:	List	
Value:	The following attributes:	
	LOCAL	Port number range 0-65535.
	REMOTE	Port number range 0-65535.
Default:	See below	
Access:	Read-only, configurable	

If the attribute is not defined in SYS_BASCON.COM, both list attributes default to value 5432.

Attribute **LOCAL** defines the port number on the local computer. Attribute **REMOTE** defines the port number on the shadowing partner computer.

7.7.1.11 PP Post-processing Policy for Object Status 2

The post-processing policy applied when the status of a process object of the application changes from 0 (OK) to 2 (OBSOLETE) or vice versa.

This attribute specifies how the activation criteria (attributes PA, AA and HA) are interpreted in case the status of the process object changes from 0 to 2 or vice versa while the object value (OV) remains unchanged. The keyword informs when the status change is considered to fulfill the criterion NEW VALUE.

Data type:	Text keyword	
Value:	"NEVER"	Never
	"WHEN_SET_TO_2"	When OS is set to 2 (but not vice versa)
	"ACTION_WHEN_SET"	AA fulfilled when OS is set from 0 to 2 or vice versa, PA and HA not fulfilled
	"WHEN_SET"	When OS is set from 0 to 2 or vice versa
	"ALWAYS"	When OS changes from 0 to 2 or vice versa, even if caused by lost connection to the station (suspension)
	"DEFAULT"	Same as "WHEN_SET_TO_2", for compatibility
Default value:	"DEFAULT"	
Access:	No restrictions	

The phrase "when OS is set" here means that the status is explicitly set by SCIL, an OPC client or a process (ACP) message, as opposed to the implicit status change to 2 caused by lost connection to the station.

When the status of a process changes from 0 to 2 or vice versa, the policy of the corresponding station object (attribute STA:BPP, see Section 11) is first applied. Only if STA:BPP = "DEFAULT", or the unit number (UN) of the object is 0, the policy defined by this APL:BPP attribute is applied.

7.7.1.12 SM Topological State Mapping

Defines the default encoding of topological states in the process database of the application. The encoding is applied to determine the TS (Topological State) attribute value of DB (Double Binary) process objects as well as to determine the return value of the SCIL API function SCIL_Get_Switch_State.

Data type:	List	
Value:	An empty list or a list of four integer-valued attributes. Each attribute has a value in range 0 ... 3. No two attributes may have the same value. In power process (network topology schema POWER), the attributes are defined as follows:	
	"OPEN "	The DB or AI value that represents the OPEN position
	"CLOSED"	The DB or AI value that represents the CLOSED position
	"MIDDLE"	The DB or AI value that represents the MIDDLE position
	"FAULTY"	The DB or AI value that represents the FAULTY position
Default value:	Empty list	
Access:	No restrictions	

If the value of the attribute is an empty list (the default value), the system-wide default, which corresponds to the SM value LIST(OPEN = 2, CLOSED = 1, MIDDLE = 0, FAULTY = 3), is applied.

This attribute may be overridden station-wise by the similar SM attribute of the station object, see Section 11.2.2.

7.7.1.13 UA UAL Configuration

Defines the UAL (User Account Logging) configuration of the application.

Data type:	List			
Value:	The following attributes:			
	SOURCE	Optional text used as the contents of the Source field of the UAL messages. If omitted, the name of the application is used.		
	SERVERS	Vector of up to 6 list elements. Each element describes an external Syslog server to receive UAL events. A Syslog server is configured by the following three attributes:		
		TYPE	Type of the server and/or used communication protocol:	
			"ARCSIGHT"	ArcSight server. This value refers to the HP ArcSight Logger developed by Hewlett Packard Co.
			"UDP"	Other Syslog server using UDP communication
			"TCP"	Other Syslog server using TCP communication
		IP_ADDRESS	Text, the IP address of the server	
		PORT	The UDP or TCP port number used by the server	
Default:	Empty list (only local UAL logging is done)			
Access:	No restrictions			

7.7.1.14 WS Windows Single Sign-on enabled

Informs whether the application uses Windows Single Sign-on functionality.

Windows Single Sign-on and its requirement are described in the Application Design manual.

Data type:	Integer	
Value:	0	Windows Single Sign-on not used
	1	Windows Single Sign-on in use
Default:	0	
Access:	No restrictions	

7.7.2 User interface related attributes

The following attributes affect the user interface of the application.

7.7.2.1 IT Input Timeout

Length of timeout for waiting for user input

Data type:	Integer
Value:	0 ... 65535, seconds to wait
Default value:	0 (no timing)
Access:	No restrictions

The input timeout affects the semi-graphic input commands !INPUT_VAR, !INPUT_KEY and !INPUT_POS. The execution of the SCIL program containing the input command is interrupted by error PICO_INPUT_TIMEOUT (961), if the operator does not respond in the specified time. The timer is reset by each character typed.

The timeout is applied in all semi-graphic pictures shown by the monitors of the application.

7.7.2.2 LA Language

The language of the application.

Data type:	Text keyword
Value:	Language according to the standard ISO 639
Default value:	"EN" (English)
Access:	No restrictions

The chosen language can be overridden by the MONn:BLA attribute and by the SCIL function SET_LANGUAGE.

7.7.2.3 MS Monitor Alarm Signal Size

The size of the monitor alarm signal in the upper right corner of the screen.

Data type:	Integer
Value:	1 ... 48
Default value:	1
Access:	No restrictions

The alarm signal is always square formed and this attribute specifies the number of semi-graphic character positions on one side of the square.

7.7.3 Operation control attributes

The following attributes keep track of events in the application.

7.7.3.1 AC Alarm Count

The number of active alarms.

Data type:	Vector
Value:	Vector of 7 integers
Indexing:	Indices 1 ... 7 refer to the alarm classes. Index 0 returns the total number. An attribute notation without an index refers to the whole vector.
Access:	Read-only

7.7.3.2 AT Alarm Tag

A tag number that is updated each time the alarm queue is updated (a new alarm occurs, an alarm is acknowledged, or an alarm is cleared).

Data type:	Integer
Value:	0 … 65 535
Access:	Read-only

The attribute can be used, for example, in the alarm list so that a change of the attribute causes an updating of the display.

7.7.3.3 HT History Tag

A tag number that is updated each time the history buffer is updated.

Data type:	Integer
Value:	0 … 65 535
Access:	Read-only

The attribute can be used, for example, in the event list so that a change of the attribute causes an updating of the display.

7.7.3.4 LT Blocking Tag

A tag number that is updated each time a blocking attribute (AB, HB, PB, UB or XB) of a process object is set or cleared.

Data type:	Integer
Value:	0 … 65 535
Access:	Read-only

The attribute can be used, for example, in the blocking list so that a change of the attribute causes an updating of the display.

7.7.3.5 UC Unacknowledged Alarm Count

The number of unacknowledged active and inactive alarms.

Data type:	Vector
Value:	Vector of seven integers
Indexing:	Indices 1 ... 7 refer to the alarm classes. Index 0 returns the total number. An attribute notation without an index refer to the whole vector.
Access:	Read-only

7.7.4 SCIL program control attributes

Setting one of the following attributes externally terminates the execution of the SCIL program in the specified SYS600 process. For example, this can be used as an emergency stop of an erroneous SCIL program that has entered an eternal loop.

7.7.4.1 PS Printer Spool Stop

Terminates the SCIL program currently run by a printer spooler process (PRIN)..

Data type:	Vector	
Value:	A vector of two integers (0 or 1)	
Indexing:	1	Process printout
	2	Report printout
Access:	Read, write	

7.7.4.2 RS Report Task Stop

Terminates the SCIL program currently run by a report process (REPR).

Data type:	Vector	
Value:	Vector of 32 integers (0 or 1)	
Indexing:	1	The time channel queue
	2	The event channel queue and
	3 to 32	The parallel queues 1 to 30
Access:	Read, write	

7.7.5 Revision compatibility

7.7.5.1 RC Revision Compatibility

Allowing the application engineer to choose the behavior of SCIL language elements when upgrading to a new SYS600 version that works differently from the version used during the application engineering. The mechanism may be useful, for example, in the following cases:

- A bug has been corrected in the new revision, but an application has taken advantage of the old bug and relies on it.
- Some limit or restriction of the program has been removed, but an application may be coded to rely on the restriction.

The compatibility with the old revision is defined by compatibility issues. The mechanism is handled by the RC attribute and a SCIL function. With the RC attribute, all the SCIL programs executed within the application may be forced to behave in the old way regarding one or more compatibility issues. The SCIL function REVISION_COMPATIBILITY enables the programmer to temporarily (in a picture, dialog system or command procedure) override the revision compatibility defined by the RC attribute.

Data type:	Text or a text vector
Value:	The name(s) of the enabled compatibility issues. Enabling a compatibility issue means that it behaves like in the previous SYS600 version. The following compatibility issues are available:
	"ON_COMMAND_EXPANSION"
	"FILE_FUNCTIONS_CREATE_DIRECTORIES"
	"SETTING_LA_AND_AG_DOES_NOT_ALARM"
	"NO_QUALITY_ATTRIBUTE_SEMANTICS"
	"NO_ALIAS_CHECKING"
	"NO_ALARM_BY_OR_AND_OF"
	"DONT_RECALCULATE_AL_AFTER_ALARM_BLOCKING"
	"CREATE_VERSION_1_FILES"

Table continues on next page

	"KEEP_FILE_VERSION_1_DATABASE_FILES"
	"DEFAULT_DAYLIGHT_POLICY_IS_CALENDAR"
	"844_COMPATIBLE_MIRRORING"
	"CREATE_VERSION_2_SCIL_DATABASES"
	"DO_NOT_SYNCHRONIZE_PICTURE_UPDATE"
	"COUPLE_AUDIO_ALARMS_AND_PRINTOUTS"
	"ALLOW_CONFLICTING_F_ATTRIBUTE_NAMES"
	"DONT_CAUSE_TAKEOVER_ON_FATAL_ERRORS"
Access:	Read-only, configurable

Example:

The application APL1 uses the compatibility issue ON_COMMAND_EXPANSION:

```
#CREATE APL1:B = LIST(...
RC = "ON_COMMAND_EXPANSION",-
...)
```

Compatibility issues:

7.7.5.2 ON_COMMAND_EXPANSION

In MicroSCADA revision 8.1 and older, the macros of each SCIL command line were expanded before the line was interpreted. This lead to an incorrect behavior in case of a single line #ON command, for example:

```
@A = "XYZ"
#ON EVENT:E1 #EXEC 'A':E2
```

When event EVENT:E1 occurred, command "#EXEC XYZ:E2" was executed regardless of the current value of A. Variable expansion is a run-time operation, which should use the current values of variables. The following worked correctly:

```
#ON EVENT:E1 #BLOCK
    #EXEC 'A':E2
#BLOCK_END
```

7.7.5.3 FILE_FUNCTIONS_CREATE_DIRECTORIES

If the file name argument of READ_TEXT or other file handling functions was given in operating system dependent format, such as \DIR\FILE.TXT, the directory was created if it did not exist. This bug has been fixed. FILE_DIRECTORY_DOES_NOT_EXIST status is now returned.

If the compatibility issue FILE_FUNCTIONS_CREATE_DIRECTORY is set (by application attribute RC or SCIL function REVISION_COMPATIBILITY), the following SCIL functions and commands create the directory, when needed, instead of returning the error status:

```
    WRITE_TEXT
    WRITE_BYTES
    WRITE_COLUMNS
    #CREATE_FILE
```

Other file handling functions (e.g. READ_TEXT) never create the directory.

7.7.5.4 SETTING_LA_AND_AG_DOES_NOT_ALARM

In rev. 8.4.0 and earlier, setting AG or LA attribute of a process object did not affect the alarm state of the object and no post-processing was done. In 8.4.1, the alarm state is updated according to the new value and normal post-processing is done. Due to the change, some old applications generate unwanted alarms and printouts when run under 8.4.1. To prevent this, this revision compatibility value was implemented.

The value can be used only as the value of the application attribute RC. It cannot be used as an argument of SCIL function REVISION_COMPATIBILITY, because event handling is done by the process database.

7.7.5.5 NO_QUALITY_ATTRIBUTE_SEMANTICS

In MicroSCADA revision 8.4.2 and earlier, the quality attributes SB (Substituted), BL (Blocked), OR (Out of Range) and OF (Overflow) have been information-only attributes, i.e. they have been stored in the process object to be available for SCIL but their values have not affected the behavior of the process object in any way.

In newer revisions the following rules apply:

- A change of a quality attribute generates an event if EE = 1.
- A change of a quality attribute activates an event channel, a printout and/or history logging if the activation is enabled (AE == 1, LD <> 0 or HE == 1) and the activation criterion (AA, PA or HA) is "NEW VALUE" or "UPDATE".
- In such an activation, the changed attribute is reported as the value of CA pseudo-attribute. If more than one attribute is changed at the same time, each change will be reported separately in any order. For example, if OV changes from 0 to 1 and SB from 1 to 0, two activations occur, one with CA == "BI", BI == 1 and SB == 0, the other with CA == "SB", BI == 1 and SB == 0.
- When the switch state (SS) or the substitution state (SU) of the object is changed, the quality attributes are set to 0.

When "NO_QUALITY_ATTRIBUTE_SEMANTICS" is specified, the quality attributes behave as in MicroSCADA rev. 8.4.2 and earlier.

7.7.5.6 NO_ALIAS_CHECKING

Since revision 8.4.2 of MicroSCADA global variables are guarded against alias references. The revision compatibility switch NO_ALIAS_CHECKING is implemented for compatibility. Status SCIL_VARIABLE_ALIASING_ERROR is generated when aliasing rules are violated.

If this switch is given (either by the RC attribute of the application or by REVISION_COMPATIBILITY function), alias checking is not done. Turning on the switch by the RC attribute disables the checking of alias referencing in the whole application. To disable the alias reference checking locally in a program, use the REVISION_COMPATIBILITY function. The REVISION_COMPATIBILITY function is described in the Programming Language SCIL manual.

The arguments of method calls, as well as all the arguments of SCIL functions except for the last one, are passed by copy instead of reference. This degrades performance, when text, bit string, byte string, vector and list arguments are used. See the Installation manual.

If the MicroSCADA base system revision 8.4.2 will be used together with applications created with earlier revisions of the base system, e.g. using LIB 4.0.1, the revision compatibility switch NO_ALIAS_CHECKING should be turned on.

7.7.5.7 NO_ALARM_BY_OR_AND_OF

Since revision 8.4.4 of MicroSCADA, the protocol specific attribute OR (Out of Range) and OF (Overflow) value 1 generate an alarm (c.f OS value 1 or FAULTY).

"NO_ALARM_BY_OR_AND_OF" may be set if the application, for a reason or another, does not like this new behavior.

7.7.5.8 DONT_RECALCULATE_AL_AFTER_ALARM_BLOCKING

Since revision 8.4.4 of MicroSCADA, the alarm state is recalculated when AB is set back to 0. However, no alarm printouts nor event channels are activated (they are not activated when AB is set to 1, neither).

"DONT_RECALCULATE_AL_AFTER_ALARM_BLOCKING" may be set if the application, for a reason or another, does not like this new behavior.

7.7.5.9 CREATE_VERSION_1_FILES

In revision 8.4.4 of MicroSCADA, the implementation of keyed files was enhanced to give better performance and to support files of any size (earlier implementation had the size limit of 32 MB). Process and report database files, pictures, representation libraries and files created by SCIL using #CREATE_FILE command are implemented as keyed files.

The files created by the new version 2 implementation cannot be read by earlier MicroSCADA revisions. However, they can be converted to version 1 format by SCIL function KEYED_FILE_MANAGER, see the Programming Language SCIL manual.

When "CREATE_VERSION_1_FILES" is set, all the files are created in the old version 1 format. The use of this value is not recommended, but it may be useful in cases where pictures are engineered in 8.4.4 environment but used in 8.4.3 or earlier.

7.7.5.10 KEEP_FILE_VERSION_1_DATABASE_FILES

In revision 8.4.4 of MicroSCADA, the implementation of keyed files was enhanced to give better performance and to support files of any size (earlier implementation had the size limit of 32 MB).

When an application is started up the first time using MicroSCADA revision 8.4.4, the process database file APL_PROCES.PRD and the report database files APL_REPORT.nnn are automatically converted to the new version 2 format. The old files are renamed by appending a postfix "_V1" to the name. If the conversion fails, for example because of disk space shortage, the old file is used and the conversion is tried again during the next start-up.

If, for some reason, the conversion is not wanted, compatibility issue "KEEP_FILE_VERSION_1_DATABASE_FILES" may be set before starting up the application. If it is later removed, the conversion takes place during the next start-up.

7.7.5.11 DEFAULT_DAYLIGHT_POLICY_IS_CALENDAR

In revision 8.4.4 of MicroSCADA, the implementation of time handling was comprehensively rewritten. Both local and UTC time as well as daylight saving time are fully supported. There is a slight incompatibility between the new and old implementation of the scheduling of time channels: The default behavior of time channels at daylight saving time/standard time switches has been changed.

Prior to revision 8.4.4, the scheduling of time channels was synchronized to the local time of the system. When the local time was moved backwards at daylight saving to standard time switch, the time channels stopped for an hour. Correspondingly, at standard to daylight saving time switch, the time channels were excessively scheduled.

In revision 8.4.4, the default behavior is that the time channels are scheduled evenly (synchronized to UTC time) when the local time changes due to daylight saving and there is a new attribute DP (Daylight Switch Policy) to specify the behavior, see the Application Objects manual.

When "DEFAULT_DAYLIGHT_POLICY_IS_CALENDAR" is set, the time channels created with earlier program revisions keep behaving as before. Even scheduling is the default behavior of new time channels, however.

7.7.5.12 844_COMPATIBLE_MIRRORING

Mirroring between MicroSCADA 8.4.4 and MicroSCADA 8.4.5 does not work when default settings are used. When upgrading from 8.4.4 to 8.4.5, both systems have to be upgraded to make mirroring work again.

If the SYS600 mirroring network is large, upgrading may cause unacceptably long breaks in the operation of the network. The compatibility issue, "844_COMPATIBLE_MIRRORING", has been implemented to ease upgrading.

When "844_COMPATIBLE_MIRRORING" is set (in MicroSCADA 8.4.5, or later), the mirroring works with an 8.4.4 application, and also with another 8.4.5 application with "844_COMPATIBLE_MIRRORING". However, it does not work with an 8.4.5 application without "844_COMPATIBLE_MIRRORING".

It is recommended to build a new mirroring network without "844_COMPATIBLE_MIRRORING". In addition, if an operation break is acceptable, an upgrade to 8.4.5 should be done without "844_COMPATIBLE_MIRRORING", because then it is easier to add new nodes in the network afterwards.

When "844_COMPATIBLE_MIRRORING" is set, the upgrading can be done system by system without disturbing the operation of the network. In this case, each new node that is added to the network later must set "844_COMPATIBLE_MIRRORING" as well.

The setting of "844_COMPATIBLE_MIRRORING" does not affect the functionality of the program or cause any decrease in performance. All the new features introduced in MicroSCADA 8.4.5 (such as hierarchical mirroring) work as specified.

7.7.5.13 CREATE_VERSION_2_SCIL_DATABASES

The internal implementation of SCIL databases (SDB) has been optimised for faster access in SYS600 9.0. An SDB created in the new (version 3) format cannot be read by MicroSCADA 8.4.5, which uses version 2 format.

When "CREATE_VERSION_2_SCIL_DATABASES" is set (in SYS600 9.0 or later), SDB’s are created in the old version 2 format. This compatibility issue may be used for convenience in cases where SDB files created in a 9.0 system are frequently moved to an 8.4.5 system.

7.7.5.14 DO_NOT_SYNCHRONIZE_PICTURE_UPDATE

The timing of update programs of pictures is synchronized to the system clock (See the Programming Language SCIL manual, command !UPDATE). In the revision 8.2 (or older), such a synchronization was not done. When an old application that relies on the old behavior is upgraded, this setting may be used to avoid recoding of the pictures.

When "DO_NOT_SYNCHRONIZE_PICTURE_UPDATE" is set, the executions of update programs are not synchronized.

This setting does not affect the cyclic methods of Visual SCIL objects.

7.7.5.15 COUPLE_AUDIO_ALARMS_AND_PRINTOUTS

Generation of audio alarms has been changed in SYS600 9.1 and in MicroSCADA 8.4.5 SP2. Audio alarms and alarm printouts are now generated independently of each other. In earlier revisions, an audio alarm was generated only when an alarm row was printed on the event printer.

When "COUPLE_AUDIO_ALARMS_AND_PRINTOUTS" is set, the audio alarms of the application are generated as in earlier program revisions.

7.7.5.16 ALLOW_CONFLICTING_F_ATTRIBUTE_NAMES

When an old (Rev. 8.4.3 or older) application was upgraded to 8.4.5 or later, the creation of F (Free Type) objects fails by PROF_FREE_ATTRIBUTE_NAME_ALREADY_EXISTS (2212), if the F object defines attribute names implemented as common process attributes in the base system in MicroSCADA revisions up to 8.4.4.

Examples of such conflicting attributes are RB, TI, TY, OI, BL, RB, OR and CT.

When "ALLOW_CONFLICTING_F_ATTRIBUTE_NAMES" is set, such conflicting attribute names are accepted when an F object is created.

This switch should be used only when an old application is upgraded, because the new base system functionality implemented by conflicting attributes will be lost when the name is overloaded. In addition, some common SCIL tools (such as the Object Navigator) and other SCIL software may be confused when the data type and meaning of some common attributes are not that expected.

7.7.5.17 DONT_CAUSE_TAKEOVER_ON_FATAL_ERRORS

Since Rev. 9.3, a deliberate HSB takeover is initiated in case of an occurrence of a fatal error (such as global memory pool overflow or lack of operating system resources), if the primary application of the system is ready for a takeover (its shadowing phase is HOT_SD).

When "DONT_CAUSE_TAKEOVER_ON_FATAL_ERRORS" is set, the system tries to continue, possibly resulting in a halt and/or takeover later on.

7.8 Application diagnostic attributes

For a hot application, application diagnostics provide an automated means of monitoring the state of other applications. The diagnostics is based on the APL-APL communication, that is why it is also called APL-APL diagnostics. Both external and local applications may be monitored.

The attributes DI (Application Diagnostic Interval) and DT (Application Diagnostic Timeout) of the monitoring application define the applications to be monitored. The status of the APL-APL connection and the state of the external application may be read from the DS (Application Diagnostic Status) attribute.

When the status of the APL-APL connection changes from bad to good or vice versa, or the state of the supervised application changes, the predefined event channel APL_EVENT is activated to allow application specific actions. For details about the event channel, see the Application Objects manual.

The diagnostics of application 'n' is started, when both DI(n) and DT(n) are set to a non-zero value. If they are non-zero at the application start-up, the diagnostics is started after the event channel APL_INIT_1 (or APL_INIT_H) has been executed.

Correspondingly, the diagnostics of application 'n' is stopped, when either DI(n) or DT(n) is set to zero.

7.8.1 DI Application Diagnostic Interval

The interval between diagnostic messages.

Data type:	Vector of 250 elements
Indexing:	The logical application number of the supervised application
Element type:	Integer 0 ... 65 535
Value:	The interval between two successive diagnostic messages
Unit:	Seconds
Default value:	0
Access:	No restrictions

7.8.2 DS Application Diagnostic Status

The status of the APL-APL connection and the state of the supervised application.

Data type:	Vector of 250 elements	
Indexing:	The logical application number of the supervised application	
Element type:	List	
Value:	Attributes of each element:	
	STATUS	Integer 0 ... 65 535
		The status of the latest diagnostic message
	AS	Text keyword, the AS (Application State) attribute of the supervised application
	SS	Text keyword, the SS (Shadowing State) attribute of the supervised application
	SP	Text keyword, the SP (Shadowing Phase) attribute of the supervised application
	The attributes AS, SS and SP are shown only when STATUS = 0, otherwise they are unknown.	
Default value:	Each element defaults to LIST(STATUS = 10).	
Access:	Read-only	

7.8.3 DT Application Diagnostic Timeout

The timeout of each diagnostic message.

Data type:	Vector of 250 elements
Indexing:	The logical application number of the supervised application
Element type:	Integer 0 ... 65 535
Value:	The timeout length of each diagnostic message
Unit:	Seconds
Default value:	0
Access:	No restrictions

Example:

```
#SET APL:BAP(5) = 5        ;Map external application 5 as logical application 5
#SET APL:BDT(5) = 10       ;Timeout 10 seconds
#SET APL:BDI(5) = 60       ;Start once-a-minute diagnostics
```

7.9 Mirroring attributes

The attributes relevant only to mirroring systems are described below. For further information about the mirroring concept, see the System Configuration manual.

7.9.1 Mirroring configuration

7.9.1.1 EP Event Queue Overflow Policy

The policy to be obeyed in the host system when the mirroring event queue is about to overflow (EU >= EM).

Data type:	Text keyword	
Value:	One of the following keywords:	
	"DISCARD"	Discard the queue and quit communication with this image application.
	"KEEP"	Prevent losing events in the image application by not accepting process messages from NET units until EU < EM.
	"NONE"	Works as "DISCARD".
Default value:	"NONE"	
Access:	No restrictions	

When the policy is "DISCARD", the host will dispose of all the events in the queue and send an overflow message to the image application. The image will then re-establish the connection and do a new subscription.

When the policy is "KEEP", the host application will do what it can to avoid losing events. Specifically, the host application works exactly in the same way as when its own event channel queue is about to overflow: it does not accept process messages from the NET unit while the queue is full. The "KEEP" policy is obeyed only while the connection to the image application is established. During a connection break, "DISCARD" policy is applied to prevent shortage of system resources.

7.9.1.2 HE Host Enabled

Enables and disables mirroring communication with this (external) host application.

Data type:	Integer	
Value:	0	Disabled:
	1	Enabled
Default value:	1	
Access:	No restrictions	

The attribute is used in the image system to temporarily block the incoming events from a host application. During the blocking, the host buffers the events and sends them when the communication is re-enabled.

7.9.1.3 IE Image Enabled

Enables and disables mirroring communication with this (external) image application.

Data type:	Integer	
Value:	0	Disabled:
	1	Enabled
Default value:	1	
Access:	No restrictions	

The attribute is used in the host system to temporarily block sending events to an image application. During the blocking, the host buffers the events and sends them when the communication is enabled again.

7.9.1.4 IS Image Stations for System Messages

The locations of the image stations that are to receive system messages from this host application. The system messages are recognized by their object address, the unit number (attribute UN) of the process object is 0.

Data type:	Vector of 10 list type elements	
Value:	Attributes of each element:	
	APL	The number of an (external) image application
	UN	The unit number within the image application.
Default value:	Each element defaults to LIST(APL = 0, UN = 0)	
Access:	No restrictions	

Note: The attribute is read by the application only at application start-up. Consequently, setting the attribute while the application is running has no immediate effect.

7.9.2 Mirroring diagnostics

7.9.2.1 HD Host Diagnostics

Mirroring diagnostics for an external host application in the image system.

Data type:	Vector of 10 integer elements	
Value:	Counters for the following:	
Indexing:	1	Received events
	2	Received event messages
	3	Transmitted process object commands
	4	Transmitted STA:S commands
	5	Established connections
	6	Connection breaks
	7	Successful reconnections after a break
	8	Stations currently mirrored
	9	Addresses currently subscribed to
	10	Missing addresses
Access:	No restrictions	

7.9.2.2 ID Image Diagnostics

Mirroring diagnostics for an external image application in a host system.

Data type:	Vector of 10 integer elements	
Value:	Counters for the following:	
	1	Transmitted events
	2	Transmitted event messages
	3	Received process object commands
	4	Not used, always 0
	5	Established connections
	6	Connection breaks
	7	Successful reconnections after a break
	8	Stations currently mirrored
	9	Addresses currently subscribed to
	10	Missing addresses
Access:	No restrictions	

7.10 Miscellaneous APL attributes

7.10.1 AD Application Self-Diagnostics

Self-diagnostics to be displayed by various tools.

Data type:	Vector	
Value:	A vector of list elements with the following attributes:	
	CATEGORY	Text, the type or category of the diagnostic message. The only value currently implemented: "LOAD" Load time diagnostics
	AREA	Text, the diagnostic area. The implemented area: "PROCESS DATABASE"
	OTYPE	Text, the object type concerned: "P" Process objects "" Not specific to any object type

Table continues on next page

	SOURCE	Text, additional info about the source of the diagnostic message: "UNIT" Process unit "" No additional info
	NUMBER	Integer, the specific number of the source: Unit number 0 (no specific source to be reported)
	STATUS	SCIL status code of the problem
	TIME	Time value, the time the diagnostics was done
	DESCRIPTION	Text vector, description of the problem
	ACTION	Text vector, description of actions to be taken by the user to fix the problem
Access:	Read-only	

In future, additional categories, areas, object types and sources may be implemented.

7.10.2 CX Comment Text

A freely chosen comment text.

Data type:	Text
Value:	Any Unicode text, up to 255 characters
Access:	No restrictions

7.10.3 EY Emergency Password

Password to be entered at emergency login.

Data type:	Integer
Value:	Positive integer
Access:	Configurable in SYS_BASCON.COM, otherwise read-only

Emergency password can be used only once. When consequent emergency login has to be done, an integer larger than the previously used has to be assigned to the EY attribute.

Instructions:

1. Stop MicroSCADA
2. Open sys_bascon.com file and insert EY attribute, see listing below. Note to use correct application.
3. Start MicroSCADA
4. Use user name EMERGENCY when logging in to the application
5. Password is same as the value of EY attribute
6. Change administrator password from the user management
7. You can leave or comment EY attribute definition in sys_bascon.com file so it is easier to remember what integer value was used last time.

```
#else #block ; *** Single System *** #loop_with j = 1 ..
length(Apl_Names) Apl_Nb = Apl_Numbers(j) #if
length(OPC_AE_Server_Enabled) < j #then Apl_OE = 0 #else_if
OPC_AE_Server_Enabled(j) #then Apl_OE = 1 #else Apl_OE = 0 #if
length(Apl_Image_Stations) < j #then Apl_IS = vector() #else APL_IS =
Apl_Image_Stations(j) Apl_Permanent = list(- NA =
Apl_Names(j),- ;Application name AS = "HOT",- ;Application state OE =
Apl_OE,- ;OPC A&E Server PH = Global_Paths, -;Global paths SV =
Apl_SV,- ;System variables (reserved) IS = Apl_IS,- ;Image Stations for
System Messages MO = Monitor_Mapping,- ;Monitor mapping PR =
Printer_Mapping,-;Printer mapping EY = 1,- ;Incremental password
```

7.10.4 SV System Variables

The attribute can be used as global application related variables. It is reserved for Hitachi Power Grids and should not be used in application programs.

Data type:	Vector
Value:	Defined by Hitachi Power Grids
Default value:	Vector of 10 integer 0’s
Access:	No restrictions

7.10.5 UV User Variables

The attribute can be used as global application related variables in application programs.

Data type:	Vector
Value:	Defined by the application
Default value:	Vector of 10 integer 0’s
Access:	No restrictions

Section 8 MON objects for base system

8.1 About this section

This section describes the MON objects and their attributes:

Section 8.2	General: The link types and the MON object notation.
Section 8.3	MON attributes: • Basic monitor definition attributes (BN, BT, BM, DT, TT) • Informative monitor attributes (AN, AP, LI, SD, SG, SZ) • Monitor control attributes (BP, ED, IL, LA, MS, PC, WC) • Miscellaneous monitor attributes (CX, SV, UV)

8.2 General

8.2.1 MON objects

The MON objects correspond to the SYS600 monitors opened on screens either by the operator or automatically. A screen, the base system screen or a workstation, can contain one or more SYS600 monitors connected to the same or different applications in the same or in different base systems.

Each SYS600 monitor that will be used by the base system and its applications must be defined as a MON object. The monitors are reserved by an application with the monitor mapping attribute (APLn:BMO, Section 8.).

8.2.2 MON object notation

The MON object attributes are accessed from SCIL with the following object notation:

MONn:Bat

where

'n'	The object number for the SYS600 monitor. The 'n' may be omitted from the object notation, whereby the notation refers to the monitor where the notation is used.
'at'	The attribute name

The MON attributes of monitors defined in another base system are accessed with the following object notation:

MONn:mBat

where

'm'	The logical application number of an external application (see the TT attribute, Section 7)
'n'	The object number of the SYS600 monitor in the other base system

An alternative way of using freely chosen object names is described in Section 5.

8.3 MON attributes

Common naming attributes BN, BT and BM are described in Section 5.

The MON specific attributes are described in the following sections.

8.3.1 Basic monitor definition attributes

Common naming attributes BN, BT and BM are described in Section 5.

The MON specific attributes are described in this and next sections.

The following attributes must be defined for all MON objects:

8.3.1.1 DT Device Type

The type of the SYS600 monitor given as a text. The type of monitor affects the user interface.

Data type:	Text keyword	
Value:	"VS"	SYS600 monitor supporting the display of Visual SCIL dialogs.
	"NONE"	Monitor is not defined
Default value:	"VS"	
Access:	Read, conditional write	

8.3.1.2 TT Translation Type

Determines the operating state of the monitor. If TT = "LOCAL", it must not be set to "NONE".

Data type:	Text keyword	
Value:	"NONE"	Off, out of operation
	"LOCAL"	Normal operating state
Default value:	"NONE"	
Access:	Read, conditional write	

8.3.2 Informative MON attributes

8.3.2.1 AN Application Monitor Number

The logical number of the monitor as seen from the controlling application.

Data type:	Integer
Value:	1 ... 100
Access:	Read-only

8.3.2.2 AP Application Number

The number of the application that controls the monitor according to the monitor mapping (the attribute APLn:BMO, Section 7).

Data type:	Integer	
Value:	0 ... 250	
Access:	Read-only	

8.3.2.3 LI Logged In

Informs whether a user has logged in on the monitor. When a user logs in or out, an event channel named MON_EVENT (if it exists) is started in the application. See the Application Objects manual.

Data type:	Integer	
Value:	0	No login
	1	Login
Access:	Read-only	

8.3.2.4 SD System Device Name

The identifier of the system device. This attribute is set automatically when the operator opens a monitor.

Data type:	Text
Value:	""
Access:	Read-only

The value is always an empty string. This attribute is preserved for compatibility reasons.

8.3.2.5 SG Semigraphic

The graphic mode identifier of the monitor. This attribute is automatically set to 1 by the base system software if the monitor is not capable of displaying primitive graphics.

Data type:	Integer	
Value:	0	Full-graphic monitor
Access:	Read-only	

The value is always 0. This attribute is preserved for compatibility reasons.

8.3.2.6 SZ Screen Size

The size of screen in pixels.

Data type:	Vector
Value:	Vector of two integers > 0: (x-size, y-size). The first element (x-size) is the horizontal size, the second element (y-size) the vertical size.
Access:	Read-only

8.3.3 Monitor control attributes

8.3.3.1 BP Blink Policy

Specifies how to handle the blink behavior in situations when the picture handler is busy, for example with a demanding SCIL command. Blinking in pictures is realized as a shift between background and foreground display.

Data type:	Text keyword	
Value:	"NONE"	No special actions taken (default)
	"FOREGROUND"	All blinking areas on screen are displayed as non-blinking
	"BACKGROUND"	The blinking areas are displayed as in blink-on state (in background color)
Access:	No restrictions	

8.3.3.2 ED Enter Key Disabled

In SYS600 pictures that are not in an input state, the enter key has the same function as the mouse click. This functionality may be disabled by this attribute.

Data type:	Integer	
Value:	0	Enter key works like a mouse click
	1	Enter key is disabled
Default value:	0	
Access:	No restrictions	

When a monitor session is closed, the value of this attribute is restored to the value it had at the beginning of the session.

8.3.3.3 IL Input Locked

Prevents the input to a SYS600 monitor from keyboard or mouse. This attribute affects semigraphic monitors only, not Visual SCIL dialogs.

Data type:	Integer	
Value:	0	Input not locked (default)
	1	Input locked: No digitizer, mouse nor keyboard input is obeyed
Default value:	0	
Access:	No restrictions	

8.3.3.4 LA Language

The default language used in dialogs and pictures.

Data type:	Text keyword
Value:	Language name according to the ISO standard 639
Default value:	The value of APL:BLA
Access:	No restrictions

The chosen language can be overridden by the SCIL function SET_LANGUAGE.

8.3.3.5 MS Monitor Stop

Terminates the SCIL program currently run by the SYS600 monitor.

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

8.3.3.6 PC Picture Containers

The maximum number of picture containers in a Visual SCIL monitor.

Data type:	Integer
Value:	0 ... 10
Default:	1
Access:	No restrictions

It is not recommended to have more than 1 picture container in a dialog, because the SCIL programs of all the pictures within a dialog are executed sequentially. Therefore, a picture doing a lengthy operation (such as !INPUT_VAR, #PAUSE etc.) blocks all the other pictures in the dialog.

8.3.3.7 WC Window Color

The color of the background behind a window in a picture and behind the blinking alarm signal.

Data type:	Text or vector
Value:	Color specification given in either of the four manners described in the Programming Language SCIL manual.
Default value:	None = Black color
Access:	No restrictions

When a window is shown on a graphics screen, this color will be shown in the window locations for a moment until the window is drawn. Likewise, when a window is erased from screen, this color is shown until the background is redrawn.

Example:

```
#SET MON1:BWC = ("M",1)
```

8.3.4 Miscellaneous MON attributes

8.3.4.1 CX Comment Text

A freely chosen comment text.

Data type:	Text
Value:	Any Unicode text, up to 255 characters
Access:	No restrictions

8.3.4.2 SV System Variables

The attribute can be used as global monitor related variables. The system variables are reserved for Hitachi Power Grids and should not be used in application programs.

Data type:	Vector
Value:	Defined by Hitachi Power Grids
Default value:	A vector of 10 integer 0's
Access:	No restrictions

8.3.4.3 UV User Variables

The attribute can be used as global variables in application programs.

Data type:	Vector
Value:	Defined by the application
Default value:	A vector of 10 integer 0's
Access:	No restrictions

Section 9 LIN objects for base system

9.1 About this section

This section describes the LIN objects and their attributes:

Section 9.2	General: The link types and the LIN object notation.
Section 9.3	LIN attributes: • Common naming attributes (BN, BT, BM) • Basic LIN Definition Attributes (LT, TR, SC) • Integrated Link Attributes (NA, TI) • Diagnostic Counters (DC) • Miscellaneous LIN attributes (CX)

9.2 General

9.2.1 LIN objects

The LIN objects describe the links and connections to adjacent nodes - base systems and communication units. All node connections must be defined as LIN objects, but several nodes can be connected to the same link and use the same LIN object definition. The object number 'n' of the LIN objects can be freely chosen in the range 1 ... 20.

A base system may use the following links:

- A LAN link to external base systems and communication modules
- One or more integrated links to PC-NET units

9.2.2 LIN object notation

From SCIL the LIN object attributes are accessed with the following object notation:

LINn:Bat

where

'n'	The object number and 'at' the attribute name. The 'n' must not be omitted from the object notation.

The LIN attributes of links defined in another base system are accessed with the following object notation:

LINn:mBat

where

'm'	The logical application number of an external application (see the TT attribute, Section 15)
'n'	The object number of the link in the other base system

An alternative way of using freely chosen object names is described in Section 5.

9.3 LIN attributes

Common naming attributes BN, BT and BM are described in Section 5.

The LIN specific attributes are described in the following sections.

9.3.1 Basic LIN definition attributes

9.3.1.1 LT Link Type

The type of the link. This attribute must be given for all types of links.

Data type:	Text keyword	
Value:	"NONE"	Undefined
	"LAN"	LAN (TCP/IP)
	"INTEGRATED"	A PC-NET unit
Default value:	"NONE"	
Access:	Read, conditional write	

When an INTEGRATED link is created, the integrated PC-NET program is started, provided that the SC attribute has been given correctly, see below.

When an INTEGRATED link is deleted (by setting the LT attribute to "NONE"), the PC-NET program is stopped and all the process objects located in the PC-NET are marked old, i.e. their status is set to 2 (OBSOLETE_STATUS).

9.3.1.2 TR Transport

The type of LAN protocol. This attribute is defined only for LAN links (LT = "LAN"). The only possibility is TCP/IP, which is also the default value.

Data type:	Text keyword
Value:	"TCPIP" or "NONE"
Default values:	"TCPIP"
Access:	Read, conditional write

9.3.1.3 SC Start Command

The location and name of the executable program of the PC-NET unit and the location and name of the configuration file.

Data type:	Text
Value:	The name of the program including path if not in the \sc\prog\exec directory and the name of the configuration file, if not \sc\sys\active\sys_\pc_net.cf1. These two names are separated with a blank space.
Default:	Empty
Access:	Read, conditional write

Examples:

```
;both the program and configuration file given
#SET LIN3:BSC="\sc\prog\exec\pc_nets.exe \sc\sys\active\sys_
\pc_net_conf.txt"
#SET LIN3:BSC="\sms\prog\exec\pc_nets.exe \sms\sys\active\sys_\pc_net.cf1"
 ;default configuration file \sc\sys\active\sys_\pc_net.cf1 used
#SET LIN3:BSC="\sc\prog\exec\pc_nets.exe"
```

9.3.2 Integrated link attributes

The following attributes apply to PC-NET connections (LT = "INTEGRATED").

9.3.2.1 NA NAK Limit

The maximum number of NAKs (negative acknowledgements) accepted by the base system. When this number is reached, the base system regards the message transmission as failed and an error status is produced (16105).

Data type:	Integer
Value:	0 ... 65 535
Default value:	3
Suggested value:	3
Access:	No restrictions

9.3.2.2 TI Timeout Length

The time limit applied when the nodes are waiting for response to a message or polling packet.

Data type:	Integer
Value:	0 ... 65 535
Unit:	Seconds
Default value:	2
Suggested value:	Depending on the Baud Rate
Access:	No restrictions

9.3.3 Diagnostic counters

9.3.3.1 DC Diagnostic Counters

A counter of all major events and error situations of the communication link.

Data type:	Vector
Value:	A vector of 16 non-negative integer values.
Indexing:	Counter number
Access:	No limitation

Each line has 16 diagnostic counters numbered 1 ... 16 and with the following meanings:

1. TRANSMITTED MESSAGES
A counter that is incremented whenever a message is transmitted successfully. A successful transmission includes the reception of a positive acknowledgment (ACK).
2. FAILED TRANSMISSIONS
A counter that is incremented when a message transmission fails. The transmission has failed if no positive acknowledgment (ACK) is received in spite of retrials. The counter is also incremented if the states of the modem signals CTS and DCD prevent transmission.
3. TRANSMIT TIMEOUTS
A counter that is incremented each time a time-out occurs during response waiting. If, for example, 3 timeouts occur at the transmission of a message (with retrials), the counter is incremented 3 times. When the retry limit is reached, the counter 2 is also incremented once.
4. TRANSMITTED ACKS
A counter that is incremented each time when a positive acknowledgment (ACK) is transmitted.
5. TRANSMITTED NAKS
A counter that is incremented each time when a negative acknowledgment (NAK) is transmitted.
6. TRANSMITTED ENQS
A counter that is incremented each time an enquiry (ENQ) is transmitted.
7. RECEIVED ACKS
A counter that is incremented each time a positive acknowledgment is received from the line.
8. RECEIVED NAKS
A counter that is incremented each time a negative acknowledgment (NAK) is received from the line.
9. RECEIVED ENQS
A counter that is incremented each time an enquiry (ENQ) is transmitted.
10. Not in use.
11. RECEIVED MESSAGES
A counter that is incremented each time a message has been received from the line without errors.
12. Not in use.
13. Not in use.
14. Not in use.
15. Not in use.
16. Not in use.

9.3.4 Miscellaneous LIN attributes

9.3.4.1 CX Comment Text

A freely chosen comment text.

Data type:	Text
Value:	Any Unicode text, up to 255 characters
Access:	No restrictions

Section 10 NOD objects for base system

10.1 About this section

This section details the NOD objects and their attributes. It contains two sections:

Section 10.2	General: The meaning of the NOD objects and the NOD object notation.
Section 10.3	NOD attributes: Common naming attributes (BN, BT, BM), Basic node definition attributes (LI, NN, NT, RN, SA), Node diagnostic attributes (DF, DI, DT), Node communication attributes (LT, RT), Security attributes (ID, PI, AI, SC, WL) and Miscellaneous NOD attributes (CX, OP).

10.2 General

10.2.1 NOD objects

The NOD objects represent nodes in the SYS600 system. The following devices and communication programs must be defined as NOD objects in a base system:

- Other base systems, which will communicate with the base system in question.
- NET units (PC-NET) which will communicate with the base system in question, directly or indirectly.
- Gateways (protocol converters using CPI-library) of various types.
- OPC Data Access Servers that are used as a data source for the process database.
- OPC Alarm and Event Servers that are used as a data source for the process database.

The NOD object numbers are global and must coincide throughout the entire network. They must also coincide with the system object node numbers (NET object numbers) defined in the communication units (see Section 15).

10.2.2 NOD object notation

From SCIL the NOD object attributes are accessed with the following object notation:

NODn:Bat

where

'n'	NOD object number 'n' can be 1 ... 250. 'n' may not be omitted from the object notation.
'at'	Attribute name

The NOD attributes of nodes defined in another base system are accessed with the following object notation:

NODn:mBat

where

'm'	The logical application number of an external application.
'n'	The object number of the node in the other base system. See the TT attribute, Section 7

An alternative way of using freely chosen object names is described in Section 5

10.3 NOD attributes

The common naming attributes BN, BT and BM are described in Section 5.

The NOD specific attributes are described in the following sections.

10.3.1 Basic node definition attributes

10.3.1.1 LI Link Number

The number of the LIN object that defines the link along which the node is reached. See Section 9.

Data type:	Integer
Value:	0 ... 20
Default value:	None
Access:	No restrictions

This attribute is irrelevant for OPC nodes. The OPC servers are addressed by OP and RN attributes.

10.3.1.2 NN LAN Node Name

The LAN node name of the node (host name) or the TCP/IP internet address of the node.

Data type:	Text
Value:	0 ... 255 characters
Default value:	No
Access:	No restrictions

The attribute applies to nodes connected to LAN. For other nodes, such as OPC nodes, this attribute may be used as a comment like description.

10.3.1.3 NT Node Type

The type of the node.

Data type:	Text keyword	
Value:	"SYS"	Base system
	"NET"	PC-NET unit
	"GATEWAY"	Gateway
	"OPC_DA"	OPC Data Access Server
	"OPC_AE"	OPC Alarm and Event Server
	"UNKNOWN"	Unknown
Default value:	"UNKNOWN"	
Access:	No restrictions	

For other than OPC server nodes, this attribute does not have to be set. If node diagnostics is enabled, it will set the attribute according to the reply from the remote node. The value of the attribute is purely informative.

For OPC server nodes, the attribute must be set to "OPC_DA" or "OPC_AE" to start the communication with the server. The computer where the server is running is specified by the RN attribute, see below. The communication may be stopped by setting NT to "UNKNOWN", and restarted by setting it back to "OPC_DA" or "OPC_AE".

10.3.1.4 CE Communication Engine

The type of the communication engine behind the node. This attribute is used by the System Configuration Tool in its internal processing.

Data type:	Text keyword	
Value:	""	Reserved for Base system nodes
	"PC-NET"	PC-NET unit
	"External OPC DA Client"	External OPC DA Client
	"Modbus Slave"	Modbus Slave
	"ICCP"	ICCP
	"IEC 61850 Server"	IEC 61850 Server
	"NET-CCT"	NET-CCT (Protocol)
Default value:	""	
Access:	Read-only, configurable	

Attribute is purely informative, and it must be set when the node is created. Reading is allowed.

10.3.1.5 RN Routing Node

Specifies another node as the router for this node.

Data type:	Integer
Value:	0 ... 250. The node number (NOD object number) of the routing node. 0 = No routing.
Default value:	0
Access:	No restrictions

All the messages sent to this node are re-routed to the node specified by the RN attribute. A routing node is useful when the node defined by the NOD object is not directly connected to the base system.

When the node type is "OPC_DA" or "OPC_AE", the RN attribute specifies the physical node where the OPC server is running (0 = the current SYS node).

For nodes of other types, the current SYS node cannot be specified as the routing node.

Only one-step routing is allowed, the node specified by the RN attribute of another node may not be re-routed. For example, if the RN attribute of node 1 is 2, the RN attribute of node 2 must be 0.

Example:

Base system A is connected via LAN to another base system B (with nd = 1), which contains an internal NET unit (with nd = 10). Use the following definition in the base system A:

```
#SET NOD10:BRN = 1
```

10.3.1.6 SA Station Address

The station address of the node used by the SYS600 internal protocol ACP.

Data type:	Integer
Value:	0 ... 255
Default value:	0
Access:	No restrictions

The station address is a number, which must be unique among all nodes (base systems and communication units) in the entire SYS600 system. When assigning station addresses, also the station addresses of the stations using the ANSI protocol must be regarded as these use the same numbering. For base systems, the station address is the value of the SYS:BSA attribute. For communication units it is NETn:SSA.

Nodes that do not communicate with the ACP protocol, such as OPC server nodes, need no station address.

10.3.2 Node diagnostic attributes

10.3.2.1 DF Diagnostic Event from First Found

This attribute specifies whether a system event (a "FOUND" type SYS_EVENT) is generated when the connection to the node has been established for the first time after the system start-up.

Data type:	Integer	
Value:	0	System event is not generated
	1	System event is generated
Default value:	0 (for compatibility)	
Access:	No restrictions	

10.3.2.2 DI Diagnostic Interval

The time in seconds between diagnostic messages from the base system to the node.

Data type:	Integer
Value:	0 ... 65 535
	0 = No diagnostics
Unit:	Seconds
Default value:	0
Access:	No restrictions

10.3.2.3 DT Diagnostic Timeout

The timeout of diagnostic messages, that is, the time the base system waits for a reply to a diagnostic message sent to the node.

Data type:	Integer
Value:	0 ... 65 535
Unit:	Seconds
Default value:	60 seconds
Access:	No restrictions

When the node type is "OPC_DA" or "OPC_AE", this attribute has no meaning. The diagnostics is enabled by the DI attribute alone.

10.3.3 Node communication attributes

10.3.3.1 LT Last Transaction

The transaction number of the last transaction (RP570 or SPA protocol) that the NET unit has sent to the base system. The attribute is updated only in the stand-by base system and applies to the NET nodes. See the LT attribute in Section 15.

Data type:	Integer
Value:	Integer
Access:	Read-only

Example:

```
#SET NET1:SLT = NOD1:BLT
```

10.3.3.2 RT Registration Time

The moment of time when the node last sent a message to the base system.

Data type:	Time
Value:	Time of last received message
Access:	Read-only

The attribute can be used for supervising NET units. Note that the time is not updated with messages that are replies, for example to diagnostic messages.

10.3.4 Security Attributes

10.3.4.1 AI Activate Identity

Activation (acceptance) of the pending identity.

Data type:	Integer
Value:	1
Access:	Write-only

Setting AI to 1 accepts the pending identity of the node, copies the PI attribute to ID and clears the PI. AI can be set using the Security tab of the node in the Base System Object Navigator tool.

10.3.4.2 ID Node Identity

The certificate of the node.

Data type:	Byte string
Value:	The fingerprint of the X.509 certificate of the node
Access:	No restrictions

10.3.4.3 PI Pending Identity

Pending certificate of the node.

Data type:	Byte string
Value:	The fingerprint of pending X.509 certificate of the node.
Access:	No restrictions

When the base system receives from a node a certificate that differs from the known certificate (attribute ID, see above), it stores the new certificate in this attribute to be accepted by the user. A warning message like "OSSL_CERTIFICATE_MISMATCH. SA: 110 Address: 172.31.2.153" is written into the system log. The pending certificate is accepted by setting attribute AI, see above.

10.3.4.4 SC Secure Communication

Tells whether the connection between the base system and the node is secure (encrypted) or not.

Data type:	Integer	
Value:	0	The communication is unsecure (or no communication established)
	1	The communication is secure
Default value:	0	
Access:	Read-only	

10.3.4.5 WL IP Whitelist

The IP addresses that are allowed to communicate with the base system via this node.

Data type:	Text or text vector
Value:	IP address(es) in the standard notation
Default value:	""
Access:	Only a super-user may set this attribute, for others it is read-only.

If an external node (typically another SYS600 or a node running External Data Access Client) attempts to connect to SYS and its IP address is not in the IP Whitelist, the connection request is rejected and a warning message is written into the system log. Outgoing connection requests to a non-whitelisted IP addresses are blocked as well.

When an IP address is dropped out of the whitelist, the active TCP/IP connection(s) to that IP address are aborted, if any.

This attribute is meaningful only for nodes connected to the LAN link using ACP protocol. For example, it has no effect on OPC_DA and OPC_AE type nodes.

This attribute has no effect when the system hardening (SYS:BHD) attribute REQUIRE_ACP_IP_WHITELISTING is FALSE.

10.3.5 Miscellaneous NOD attributes

10.3.5.1 CX Comment Text

A freely chosen comment text.

Data type:	Text
Value:	Any Unicode text, up to 255 characters
Access:	No restrictions

10.3.5.2 OP OPC Server

OPC server configuration.

Data type:	List		
Value:	Various attributes described below.		
	The following two attributes are meaningful for any node object that describes a connection to a remote computer:		
	US	Unicode text, the user name (optional) This attribute is used to connect an OPC client to the server as a user other than MicroSCADA. The name may be prefixed by the domain name, for example "DOMAIN\USER".	
	PW	Unicode text, the password of the user specified by US (optional) This attribute is hidden when evaluating the OP attribute in SCIL.	
	The following three attributes are valid fo OPC_DA and OPC_AE type nodes:		
	CI	Text, the class id (CLSID) of the OPC server in the standard Windows format, e.g. "{CE0322A9-65A9-4268-84D5-DD7A17E94C56}".	
	SN	Text, the name of the server (optional, for SCIL use only). Used by the Namespace Browser of the Object Navigator, as an example.	
	SK	Text, the server kind. This value is used to take vendor specific actions in the server - client communication.	
		If SK is not given (or is unknown to the base system), a generic server with no vendor specific features is assumed.	
		The currently recognized values are:	
		"AC 800"	ABB AC 800 series
			The A&E client performs an automatic refresh when it receives the (vendor-specific) NeedsRefresh event from the server.
	The following attribute is valid for OPC_DA type nodes:		

Table continues on next page

GR — List vector, specifies the item groups to be used by the SYS600 OPC DA client(s) of this server. The default value is an empty vector. Each element of the vector defines one item group by the following attributes:

IG	Unicode text, the name of the group. The process objects whose IG (Item Group) attribute matches this name are put into this group. The name is 1 to 255 characters long, it may contain any characters and is case-sensitive.
UR	Integer, the update rate of the group (milliseconds), default 0. Value -1 specifies that no spontaneous updates are sent by the server. The values of the items in the group are read by demand using the SCIL function OPC_DA_REFRESH, see the Programming Language SCIL manual.
PD	Real or integer, the percent deadband of the group, default 0.0.

The following attribute is valid for OPC_AE type nodes:

AA — Auto acknowledge, integer 0 or 1.If AA is set to 1, acknowledging the SYS600 alarm (via the AR attribute of the process object) automatically acknowledges the corresponding A&E condition, if any. If AA is set to 0 or is missing, the A&E conditions may be acknowledged only by using the SCIL function OPC_AE_ACKNOWLEDGE.

Default:	Empty list
Access:	No restrictions

The OPC server is located by the RN attribute, which identifies the network node where the server is running, and the CI field of the OP attribute, which identifies the server within the node.

The user account for launching an OPC server is defined by the US and PW attributes of the node describing the server. If not given, the user account is defined by the referenced network node object (RN). If not defined there either, the MicroSCADA user account is applied.

Every OPC_DA node (OPC Data Access Server) has two predefined item groups, which are used when the IG attribute of a process object has been left undefined (IG = ""):

- DefaultGroupIn is the default group for input objects. Its UR and PD are both zeroes.
- DefaultGroupOut is the default group for output objects (process object types BO, DO and AO). This group is not subscribed to.

The "DefaultGroupIn" may be overridden by the OP attribute, for example to specify a non-zero update rate.

Setting this attribute does not affect an existing connection to the OPC server. To apply the changes, restart the communication (by setting the NT attribute to "UNKNOWN" and back to "OPC_DA" or "OPC_AE").

Section 11 STA and STY objects for base system

11.1 About this section

This section describes the base system objects related to stations (process units):

Section 11.2.1	General
Section 11.2.2.1	Basic attributes (AP, BN, BT, BM, CS, CX, ND, PF, PP, SM, ST, TN, TB, TR, TT, UN).
Section 11.2.2.2	Mirroring attributes (AE, HS, IS, LP, MR).
Section 11.2.2.3	Redundancy (AS, PR, PS, RR, RS, SS).
Section 11.3	STY objects and their attributes (BN, BT, BM, CT, CX, DB, LP, NA).

11.2 STA objects

11.2.1 General

11.2.1.1 STA objects

Each station, that is communicating with the base system and is not of the default STA type (the attribute SYS:BDS) or connected to a communication unit other than the default NET unit (the attribute SYS:BDN), must be defined as a STA object. Stations of the default STA type connected to the default NET unit need not be individually defined. Devices that are not communicating with base system, such as star couplers in LONWORKS® [1] networks need not be defined as STA objects in the base systems.

11.2.1.2 STA object notation

From SCIL the STA object attributes are accessed with the following object notation:

STAn:Bat

where

'n'	0 ... 50 000. 'n' must not be omitted from the object notation

The STA attributes of stations defined in another base system are accessed with the following object notation:

STAn:mBat

where

'm'	The logical application number of an external application (see the TT attribute in Section 7.
'n'	The object number of the station in the other base system

An alternative way of using freely chosen object names is described in Section 5.

[1] LONWORKS is a trademark of Echelon Corporation registered in the United States and other countries.

11.2.2 STA attributes

Common naming attributes BN, BT and BM are described in Section 5.

The STA specific attributes are described below.

11.2.2.1 Basic attributes

AP Application Number
The number of the application that is connected to the STA object.

Data type:	Integer
Value:	0 ... 250
Default value:	0
Access:	Read-only

The attribute is set when the first AUTO state process object connects to the station.

CS Communication State
The state of communication between the station and the base system.

Data type:	Text	
Value:	"NONE"	No communication
	"RUNNING"	Communication established This state is entered when a RUNNING system message or an update message from the station is received.
	"SUSPENDED"	Communication lost This state is entered when a SUSPENDED system message is received or the connection to the station's node is lost.
Default value:	"NONE"	
Access:	Read-only	

The attribute is set when the communication between a process database unit and the station is established. This happens when the first process object of the unit is created.

The implementation is based on the RUNNING/SUSPENDED system messages of stations.
In order to receive these messages, the NET attribute SE must be 3 or 4 for the PC-NET communication unit. The recommended value is 4. See the NET attribute SE (Section 15.3.4) for more information.
For External DA client communication unit, NET attribute SE=4 must also be used. This setting can be defined using External DA Client configuration tool, from the command line when External DA client is started or from configuration file of the External DA client. For more information, see the External OPC Data Access Client User's Guide.

CX Comment Text
A freely chosen comment text.

Data type:	Text
Value:	Any Unicode text, up to 255 characters
Access:	No restrictions

ND Node Number

The node number of the communication unit to which the station is connected.

Data type:	Integer
Value:	0 ... 250
Default value:	SYS:BDN
Access:	No restrictions (see below)

The node number of the NET unit is the same as the NET system object number.

If the station type of the station is "OPC" or "OAE", the ND attribute is the node number of the node object that specifies the OPC server of the station (see Section 10).

For proxy stations (see Section 11.2.2.3), this is a read-only attribute that reflects the ND attribute of the active station. When no connection to the process has been established, its value is 0.

This attribute has no functional purpose for the base system if the station is an image station (MR = "IMAGE"). However, it may be set for documentation or application purposes.

PF Post-processing Policy for Object Status 1

This attribute specifies how the incoming status 1 from this station is treated in the process database.

The status may be stored in the database as a FAULTY status 1 (default behavior), or it may be changed to 2 (OBSOLETE_STATUS, or INVALID).

Data type:	Text keyword	
Value:	"DEFAULT"	No station specific policy has been defined, the policy of the application (APL:BPF) is applied.
	"REGARD_AS_2"	The status is changed to 2.
Default value:	"DEFAULT"	
Access:	No restrictions	

This attribute has been implemented to prevent excessive events and alarms from gateway applications that report some communication problems by status 1.

PP Post-processing Policy for Object Status 2

The post-processing policy applied when the status of a process object of this station changes from 0 (OK) to 2 (OBSOLETE), or vice versa.

This attribute specifies how the activation criteria (attributes PA, AA and HA) are interpreted in case the status of the process object changes from 0 to 2 or vice versa while the object value (OV) remains unchanged. The keyword specifies when the status change is considered to fulfill the criterion NEW VALUE.

Data type:	Text keyword	
Value:	"NEVER"	Never
	"WHEN_SET_TO_2"	When OS is set to 2 (but not vice versa)
	"ACTION_WHEN_SET"	AA fulfilled when OS is set from 0 to 2 or vice versa, PA and HA not fulfilled
	"WHEN_SET"	When OS is set from 0 to 2 or vice versa
	"ALWAYS"	When OS changes from 0 to 2 or vice versa, even if caused by lost connection to the station (suspension)

Table continues on next page

	"DEFAULT"	No station specific policy has been defined, the policy of the application (APL:BPP) is applied.
Default value:	"DEFAULT"	
Access:	No restrictions	

The phrase "when OS is set" means here that the status is explicitly set by SCIL, an OPC client or a process (ACP) message, as opposed to the implicit status change to 2 caused by lost connection to the station.

If the policy is "DEFAULT" and the policy of the application (APL:BPP, see Section 7) is "DEFAULT" as well, the "WHEN_SET_TO_2" policy is applied for compatibility.

SM Topological State Mapping

Defines the encoding of topological states of the process objects within the station. The encoding is applied to determine the TS (Topological State) attribute value of DB (Double Binary) process objects as well as to determine the return value of the SCIL API function SCIL_Get_Switch_State.

Data type:	List	
Value:	An empty list or a list of four integer-valued attributes. Each attribute has a value in range 0 ... 3. No two attributes may have the same value. In power process (network topology schema POWER), the attributes are defined as follows:	
	"OPEN"	The DB or AI value that represents the OPEN position
	"CLOSED"	The DB or AI value that represents the CLOSED position
	"MIDDLE"	The DB or AI value that represents the MIDDLE position
	"FAULTY"	The DB or AI value that represents the FAULTY position
Default value:	Empty list	
Access:	No restrictions	

If the value of the attribute is an empty list (the default value), the application default specified by the similar SM attribute of the application object is applied, see Section 7.7.1.

ST Station Type

The type of the station. Some of the station types are predefined. Other types can be defined with the STY objects described in the next section.

Data type:	Text keyword	
Value:	Station type name:	
	"NONE"	Undefined
	"STA"	Stations using the ANSI X3.28 protocol: Allen-Bradley PLC, Westronic D20 and M4000,SRIO, SELMA and SCPmicro
	"SPA"	SPACOM connected to a NET unit via the SPAprotocol or via a LONWORKS line and an LSG device
	"REX"	REx relays (REF, RED, REC, etc.) connected via a LONWORKS line
	"RTU"	S.P.I.D.E.R. RTU
	"PCL"	Procontrol 214
	"RCT"	PROCOL Station Interface
	"LCU"	Local Control Unit (Load Management)

Table continues on next page

"IEC"	International Electrotechnical Commission standard protocol. IEC 60870-5-101 and IEC 60870-5-104 for upper (NCC) level and process level communication, IEC 60870-5-103 for process level communication. IEC 61850 for upper level communication and IEC 60870-6 (ICCP) for both client and server communication.
"DNP"	Distributed Network Protocol for upper level and process level communication
"OPC"	Any field device exposing its data via an OPC Data Access Server
"OAE"	Any field device exposing its data via an OPC Alarm and Event Server

Additional station type names defined by STY objects, see Section 11.3.

Default value:	SYS:BDS
Access:	No restrictions

TB Time Bias

Time bias, as minutes, between the local time of the base system and the local time of the station. The local time of the station is calculated by adding TB to the local time of the base system. This attribute has no meaning if the value of TR attribute is not "STA".

Data type:	Integer
Value:	-1440 ... 1440 (minutes)
Default value:	0
Access:	No restrictions

TN Translated Object Number

The station (or device) number within the communication unit to which the station is directly connected, or the base system object number of an ALIAS station.

Data type:	Integer
Value:	0 ... 2047 (Communication unit limitations may exist)
	TN = 0 for an EXTERNAL NET-connected station refers to the broadcast "station". See Section 17.
Default value:	The same as the base system object number
Access:	No restrictions (see below)

For stations connected to a NET unit, TN is the STA system object number in the NET. The communication unit may have own limitations for the value range of the attribute TN and the amount of STA objects created to the instance of the unit. These limitations may depend on the used protocol. For the PC-NET communication unit, see description of the NET Node attribute DV in Section 15.4.4. For other communication units, see corresponding manual. If TN is modified on-line, the application logic must ensure that a valid basesystem STA object is configured for each STA object configured to the NET unit all the time. The swapping of the TN values between two STA objects is usually used in project specific solutions of back-up connections.

For "OPC" type stations, TN is the device number within the OPC server, or 0, if the OPC server does not support numbered devices.

For proxy stations (see Section 11.2.2.3), this is a read-only attribute that reflects the TN attribute of the active station. When no connection to the process has been established, its value is 0.

This attribute has no functional purpose for the base system if the station is an image station (MR = "IMAGE"). However, It may be set for documentation or application purposes.

TR Time Reference
The time reference of the station. Specifies how the time stamps sent by the station are to be interpreted.

Data type:	Text keyword	
Value:	"LOCAL"	Local time of the base system
	"UTC"	UTC time
	"STA"	Local time different from the base system
Default value:	"LOCAL"	
Access:	No restrictions	

If value "STA", i.e. local time different from the base system's local time, is specified, attribute TB must be set to tell the bias.

TT Translation Type
The operating state and location of the station.

Data type:	Text keyword	
Value:	"NONE"	Off, out of operation
	"ALIAS"	The STA object number refers to another STA object
	"EXTERNAL"	Connected to a communication unit (the normal case)
Default value:	"NONE"	
Access:	No restrictions	

UN Unit Number
The number of the unit (in the application specified by attribute AP) that is connected to the STA object.

Data type:	Integer
Value:	0 ... 50000
Default value:	0
Access:	Read-only

The attribute is set when the first AUTO state process object connects to the station.

11.2.2.2 Mirroring attributes

For further information about the mirroring concept, see the System Configuration manual.

AE Analog Events
The addresses of the analog input objects that are considered as event objects in mirroring and redundant communication.

Data type:	Integer or text vector of any length
Value:	Each element defines an object address of an analog input object that is to be considered as an event object. The object address is given as a text (IN, Item Name) for "OPC" type stations, or as an integer (OA, Object Address) for other stations.
Default value:	Empty vector
Access:	No restrictions

An 'event object' here means that every update of the object value is significant and may not be sacrificed for communication throughput.

Notes:

1. The value may be set only as a whole. Single vector elements may not be set.
2. In mirroring, the value of this attribute is in use only when the mirroring role (MR) of the station is "HOST" or "BOTH".
3. In mirroring, the value of this attribute is checked only when a subscription of process events is received from an image application. If, during the mirroring communication, an immediate effect after a change is required, the mirroring of the station must be stopped and then restarted (for example by setting MR to "NONE" and then back to "HOST").

HS Host Station

The location of the host station of this (image) station.

Data type:	List	
Value:	Attributes of the list:	
	APL	The number of the (external) host application
	UN	The unit number within the host application.
		UN value 0 refers to the system messages of the host application.
Default value:	LIST(APL = 0, UN = 0)	
Access:	May be set only when MR is "IMAGE" or "BOTH", otherwise no restrictions	

This attribute is cleared (APL and UN set to 0) when attribute MR is set to "NONE" or "HOST".

IS Image Stations

The locations of the image stations of this (host) station.

Data type:	Vector of 10 list type elements	
Value:	Attributes of each element:	
	APL	The number of an (external) image application
	UN	The unit number within the image application.
Default value:	Each element defaults to LIST(APL = 0, UN = 0)	
Access:	May be set only when MR is "HOST" or "BOTH", otherwise no restrictions	

This attribute is cleared (APL and UN of each vector element set to 0) when attribute MR is set to "NONE" or "IMAGE".

LP Load Control Policy

The load control policy applied to the analog input objects of the station.

Data type:	Text keyword	
Value:	"DEFAULT"	See below.
	"KEEP_NO_ANALOGS"	All the analog objects are subject to load control.
	"KEEP_TIME_STAMPED_ ANALOGS"	The analog objects that are not time-stamped by the station are subject the load control.
	"KEEP_ALL_ANALOGS"	No analog object events are subject to load control.
Default value:	"DEFAULT"	
Access:	No restrictions	

The "DEFAULT" behavior depends on the station type. The policy is defined by the LP attribute of the corresponding STY object.

In mirroring, the value of this attribute has a meaning only when the mirroring role (MR) of the station is "HOST" or "BOTH".

For proxy stations (see Section 11.2.2.3), this attribute defines the buffering scheme of analog process events.

The value of this attribute is checked only when the process database is set up. If an immediate effect after a change is required, the mirroring of the station must be stopped and then restarted (for example by setting MR to "NONE" and then back to "HOST").

MR Mirroring Role
The role of the station in the mirroring environment.

Data type:	Text keyword	
Value:	"NONE"	Not participating in mirroring
	"HOST"	Host station (sending process data to one or more image stations in external applications)
	"IMAGE"	Image station (receiving process data from the host station in an external application)
	"BOTH"	Acts both as an image and as a host station in a hierarchical mirroring system
Default value:	"NONE"	
Access:	No restrictions (see notes below)	

Notes:

1. Setting MR to "NONE" clears HS and IS (APL and UN are set to 0).
2. Setting MR from "BOTH" to "HOST" clears HS.
3. Setting MR from "BOTH" to "IMAGE" clears IS.

11.2.2.3 Redundancy

Overview
A redundant station configuration consists of

- one IED with redundant communication with the base system, or
- two redundant IEDs and their independent communication with the base system.

In the first case, the events received by the base system via the two communication links are identical. In the second case, time stamps of the events may differ slightly and polled values (typically analog values) are received at mutually independent intervals.

The implementation of station redundancy at the base system level is described below. It guarantees that no events are lost nor duplicated when one of the two communication links between the IED and the base system fails.

The implementation is based on the RUNNING/SUSPENDED system messages of stations.
In order to receive these messages, the NET attribute SE must be 3 or 4 for the PC-NET communication unit. The recommended value is 4, see the NET attribute SE (Section 15.3.4) for more information.
When the station redundancy is used with STA objects created to PC-NET unit, the values of the station attributes AS (Allocating Application) and MS (Message Application) must be equal.
For External DA client communication unit, NET attribute SE=4 must also be used. This setting can be defined using External DA Client configuration tool, from the command line when External DA client is started or from configuration file the for External DA client. For more information, see the External OPC Data Access Client User's Guide.

Redundancy roles

A redundant station is described by three interconnected STA objects:

1. The proxy STA object represents the redundant station for the application. It is not directly connected to the IED.
2. The primary STA object represents the primary communication route between the base system and the IED.
3. The secondary STA object represents the secondary communication route between the base system and the IED.

The role of each STA object in the redundancy is defined by their RR (Redundancy Role) attribute. Each proxy STA object contains references to its primary and secondary STA object (attributes PS and SS), and each primary and secondary STA object contains a reference to its proxy STA object (attribute RS).

Active and stand-by routing

In a normal situation, events from the IED are received via both communication routes (that is, by the primary and the secondary STA object), and the primary STA is the active STA object of the proxy. The events received by the active STA object are immediately written to the process database, while the events received by the stand-by STA object (normally the secondary STA) are buffered by the base system.

Commands (process object SET’s) and requests (access to STA:S attributes) for the proxy STA object are sent via its active STA object. Primary and secondary STA objects may not be explicitly accessed by the process database, no unit (UN) in the database can refer to a primary or a secondary STA. On the other hand, the primary and the secondary STA may be explicitly accessed via STA:S attributes.

The currently active station may be read from the AS (Active Station) attribute of the proxy STA object. By setting the same attribute, the active station is forced and automatic routing by the base system is disabled.

Switch-over

When the connection to the IED via the currently active STA object has been lost, the base system automatically switches to use the stand-by STA object, if it is in RUNNING state. The buffered events of the stand-by STA are browsed and new events (the events not received via the lost connection) are fed into the process database. Communication between SYS600 and the IED then continues normally via the new route. No events are lost nor duplicated.

When both connections to the IED have been established (or re-established after a connection break), the PR (Prefer Primary) attribute of the proxy STA object is examined by the base system. If it is set to 1 and the secondary STA is currently active, a switch-over to the primary takes place.

An application event (APL_EVENT) is generated for each change of active station, see the Application Objects manual.

Buffering scheme

An event received via the stand-by route is buffered if

- it is newer than the value stored in the process database
- it is as old as the value in the database but the values are different

Here, the event time stamps are considered equal if they differ less than 10 milliseconds. This allows for a small synchronization bias between redundant IED's.

Not all analog values are kept, they are buffered according to the scheme used in mirroring. The policy is configured by the LP (Load Control Policy) and the AE (Analog Events) attribute of the proxy STA object, see Section 11.2.2.2.

Up to 10 000 events per station are buffered.

HSB systems

When redundant (hot stand-by) SYS600 systems are used, another type of redundancy may be used: The IED (or a redundant IED pair) may send its events to both HSB base systems, even if the target application is hot only in one of them.

The events sent to the cold application that is in HOT_RC shadowing phase are buffered by the base system exactly like the events received via the stand-by route in the redundant station scheme described above. The situation may be seen as an analog of a redundant station, where the events received by shadowing come from the ’active station’ and the events from the communication unit come from the 'stand-by station'.

When the receiving application becomes hot at HSB switch-over, the buffered new events that have not been received via shadowing are fed into the process database. Again, no events are lost nor duplicated.

The HSB functionality described here is totally automatic, no configuration is needed.

Note that the behavior at HSB switch-over is independent of station redundancy. It works in the same way for redundant and non-redundant stations. As a consequence, External OPC DA Clients do not have to do their own buffering to cope with HSB switch-overs. Buffering in the cold application is not available with PC-NET communication units. This means that the communication lines of the PC-NET unit should be kept out of use when the main application is cold. For more information, see the from System Configuration manual.

Attributes

The attributes related to redundant stations are described below. In addition, attributes AE (Analog Events) and LP (Load Control Policy) are applied to redundant stations, see Section 11.2.2.2.

AS Active Station

The active station of a proxy station.

Data type:	Integer	
Value:	0	No active station (or no redundancy)
	1	Primary station
	2	Secondary station
Default value:	0	
Access:	No restrictions	

This attribute is relevant to PROXY objects only. In other STA objects, its value must be 0.

When read, the attribute tells which one of the two routings currently feeds the process database. By default, the routing is automatically selected by the base system.

When set to 1 or 2 (or the STA object is created with AS value 1 or 2), the routing is forced to the given value. Automatic routing is disabled.

When set back to 0, automatic routing is re-enabled.

PR Prefer Primary
Defines whether the base system should prefer the primary to the secondary station in its automatic routing.

Data type:	Integer	
Value:	0	Do not prefer the primary station (or no redundancy)
	1	Prefer the primary station
Default:	0	
Access:	No restrictions	

This attribute is effective only when the secondary station is the active station and the primary station becomes RUNNING. If PR is set to 1, the base system in this situation switches to use the primary routing.

PS Primary Station
The station number of the primary station of a proxy station.

Data type:	Integer
Value:	0 ... 50000
Default value:	0
Access:	Settable only when TT = "NONE", otherwise no restrictions

This attribute is relevant to PROXY objects only. In other STA objects, its value must be 0.

RR Redundancy Role
The redundancy role of the STA object.

Data type:	Text keyword	
Value:	"NONE"	No redundancy
	"PROXY"	The STA object defines a redundant (logical) station.
	"PRIMARY"	The STA object defines the primary connection to a redundant station.
	"SECONDARY"	The STA object defines the secondary connection to a redundant station.
Default value:	"NONE"	
Access:	Settable only when TT = "NONE", otherwise no restrictions	

RS Proxy Station
The station number of the proxy station.

Data type:	Integer
Value:	0 ... 50000
Default value:	0
Access:	Settable only when TT = "NONE", otherwise no restrictions

This attribute is relevant to PRIMARY and SECONDARY objects only. In other STA objects, its value must be 0.

SS Secondary Station

The station number of the secondary station of a proxy station.

Data type:	Integer
Value:	0 ... 50000
Default value:	0
Access:	Settable only when TT = "NONE", otherwise no restrictions

This attribute is relevant to PROXY objects only. In other STA objects, its value must be 0.

Configuration example

The following piece of SCIL code configures a redundant station.

The primary and secondary STA object are first created but not started, because the proxy STA does not yet exist. After the proxy has been created, the primary and the secondary are ready to start. The first example below shows the configuration when the primary and secondary stations are in different nodes. The value of TN defines the used object number in the communication module and same TN value cannot appear twice within the same node. This example is applicable to especially to External OPC DA client nodes where the same configuration file is used as a template for both primary and secondary nodes.

```
#CREATE STA101:B = LIST(ST="SPA", ND=50, TN=1, TT="NONE",-
                                                         RR="PRIMARY",
RS=100)
#CREATE STA102:B = LIST(ST="SPA", ND=51, TN=1, TT="NONE",-
                                                         RR="SECONDARY",
RS=100)
#CREATE STA100:B = LIST(ST="SPA", ND=0, TN=0, TT="EXTERNAL",-
                                                         RR="PROXY",
PS=101, SS=102, PR=1)
#SET STA101:BTT = "EXTERNAL"
#SET STA102:BTT = "EXTERNAL"
```

The second example below shows the configuration when the primary and secondary stations are in the same node. The value of TN defines the used object number in the communication module and it must be different within the same node. The value of the TN is freely selectable but the most common practise is to use the same value as in the station number. This example is applicable especially to PC-NET nodes. It is worth to note the if the primary and secondary nodes are configured to the same node, the decreases the level of the redundancy.

```
#CREATE STA101:B = LIST(ST="IEC", ND=3, TN=101, TT="NONE",-
                                                         RR="PRIMARY",
RS=100)
#CREATE STA102:B = LIST(ST="IEC", ND=3, TN=102, TT="NONE",-
                                                         RR="SECONDARY",
RS=100)
#CREATE STA100:B = LIST(ST="IEC", ND=0, TN=0, TT="EXTERNAL",-
                                                         RR="PROXY",
PS=101, SS=102, PR=1)
#SET STA101:BTT = "EXTERNAL"
#SET STA102:BTT = "EXTERNAL"
```

The operation is fully functional not until at least one process object to the destination application has been created for the proxy station and it is in automatic state (SS=2).

In case the primary and secondary STA objects are connected to different IEDs, it is possible that some data items from the process have different values depending on the IED which is transmitting the data. Even if the IEDs are controlling the same part of the process.

Because of this, it is recommended to update the process database whenever the active station changes. In practise, this can be implemented by attaching a piece of code to the command procedure which is activated by pre-defined event channel APL_EVENT. Snapshot variables %SOURCE_NR defines the STA number and the required action is dependent on used protocol. See protocol specific manuals for details.

```
#if (%SOURCE =="UN") and (%EVENT=="PRIMARY" or %EVENT=="SECONDARY"
or             %EVENT=="FORCED_PRIMARY" or %EVENT=="FORCED_SECONDARY")
#then #block
 #exec_after 15 DO_61850_REFRESH:C(@STA_NR=%SOURCE_NR) ; Delayed UP write
for IEC61850

       ;#set sta'SOURCE_NR':SGI=1 ; IEC101/104/103 master, Modbus, DNP3.0
       ;#set sta'SOURCE_NR':SSC=1 ; RP570 master
       ;#set sta'SOURCE_NR':SUP=1; SPA

    #block_end
```

For IEC61850 devices the forced update is needed because the events from the IED which becomes active can be considered as buffered and will be discarded from the External OPC DA Client. The time delay is needed to handle the situation when the system starts, the communication ok event is received before the buffered reports has been enabled and without the delay invalid data would be transmitted creating unnecessary burst to event list and possible NCC connections.

11.3 STY objects

11.3.1 General

11.3.1.1 Station types

The predefined station types listed in the ST attribute description in Section 11.2.2 are not enough for all possible station types. Using the STY object, additional station types can be defined provided that they can use an existing database interface (the database interface of a predefined station type). Using the STY object also enables that some properties of the predefined station types can be accessed.

The station types are internally recognized by an integer 0 ... 33. The station types with numbers 0 ... 21, 29 … 30 and 32 ... 33 are reserved for predefined types. Other station types are reserved as follows:

- Type number 22 for the "SPI" type (stations connected via RP570 slave protocol).
- Type number 23 for "LMK" stations (stations on LONWORKS lines, other than REx and SPA).

The station types can be given freely chosen names. However, it is recommended that the names given in Table 1 are used because these are the names used on NET unit. The table lists some station types and the corresponding database interfaces.

Table 1: Station types and the corresponding database interface

Station type	Name (recomm.)	Database interface
Stations connected via the RP570 slave protocol	SPI	STA
LSG device and other LONWORKS devices, except REX	LMK	REX
PLCs using MODBUS	PLC	RTU

11.3.1.2 STY object notation

The STY objects and their attributes are accessed with the following object notation:

STYn:Bat

where

'n'	the internal station type number, 1 ... 33. The following station type numbers are preset:

3	STA
4	RTU
5	SIN
6	PCL
7	SID
8	PAC
9	SAT
17	REX
20	LCU
21	SPA
22*	SPI
23*	LMK
24*	ADE
25*	PCO
26*	WES
27*	ATR
28*	PLC
29	IEC
30	DNP
32	OPC
33	OAE

* These station type numbers have to be created separately.

An alternative way of using freely chosen object names is described in Section 5.

11.3.2 STY attributes

The STY specific attributes are described below.

11.3.2.1 CT Cause of Transmission Mapping

The logical meaning of the CT attribute of the process objects updated from stations of the station type in question.

Data type:	Vector
Value:	A text vector of length 255. Each element in the vector has one of the following keyword values
	"UNKNOWN"

Table continues on next page

	"SPONTANEOUS"	
	"INTERROGATED"	
	"HISTORY"	
Indexing:	1 ... 255. The value of the process object attribute CT	
Default value:	The CT attributes of the STY objects are all initialized to "UNKNOWN", except for the following REX device values:	
	STY17:BCT1	"SPONTANEOUS"
	STY17:BCT2	"INTERROGATED"
	STY17:BCT3	"HISTORY"
Access:	No restrictions	

Example:

If a device of type 17 (REX_DEVICE) updates a process object's CT attribute to value 2, the attribute STY17:BCT2 defines the reason of that transmission ("INTERROGATED" by default).

11.3.2.2 CX Comment Text

A freely chosen comment text.

Data type:	Text
Value:	Any Unicode text, up to 255 characters
Access:	No restrictions

11.3.2.3 DB Data Interface

Specifies the predefined station type whose data interface will be used. When a new station type is defined, it must use the same data interface as a predefined station type, see Table 1. The predefined interface is used when NET unit issues commands to the stations of the new type.

Data type:	Text keyword
Value:	The three-letter station type name of a predefined station type, see Table 1
Access:	Read-only, configurable

Example:

Creating a station type "SPI" (type 22):

```
#CREATE STY22:B = LIST(NA = "SPI", DB = "STA")
```

11.3.2.4 LP Load Control Policy

Load control policy applied to the analog input objects of the stations of the type.

Data type:	Text keyword	
Value:	"DEFAULT"	See below.
	"KEEP_NO_ANALOGS"	All analog objects are subject to load control.
	"KEEP_TIME_STAMPED_ ANALOGS"	The analog objects that are not time-stamped by the station are subject to load control.

Table continues on next page

	"KEEP_ALL_ANALOGS"	No analog object events are subject to load control.
Default value:	"DEFAULT"	
Access:	No restrictions	

For those STY objects that have the DB attribute value "STA", the "DEFAULT" policy is equivalent to "KEEP_TIME_STAMPED_ANALOGS". In all other cases, the "DEFAULT" policy is equivalent to "KEEP_NO_ANALOGS".

This attribute is overridden on a station basis by the LP attribute of the station object.

The value of this attribute is checked only when the process database is set up for mirroring. If an immediate effect after a change is required, the mirroring of the stations of the type must be stopped and then restarted (for example by setting MR to "NONE" and then back to "HOST").

11.3.2.5 NA Type Name

The name of the station type.

Data type:	Text
Value:	Text of max. 10 characters
Access:	Read-only, configurable

Section 12 PRI objects for base system

12.1 About this section

This section describes the PRI base system objects and their attributes. It contains two sections:

Section 12.2	General: the definition of PRI objects and the PRI object notation.
Section 12.3	PRI attributes:

- Common naming attributes (BN, BT, BM)
- Common printer attributes (DC, DT, TT)
- Printer connection attributes (ND, SD, TN)
- Printout attributes (CS, HF, LN, LP, PN)
- Printer queue attributes (QM, QU)
- Printer log attributes (LD, LF, LL, OD)
- Printer control attributes (CL, OJ, ST)
- Miscellaneous PRI attributes: comment (CX), global variables (SV, UV).

12.2 General

12.2.1 PRI object definitions

All printers used by the SYS600 base system (and its applications) must be defined as PRI objects, whether they are connected directly to the base system computer, to a LAN or to a NET unit. A base system can use up to 20 printers. The printers are connected to an application using the printer mapping attribute (APLn:BPR, see Section 7). A certain printer can be used by several applications in different base systems.

It is possible to define printers that have no real physical correspondence (virtual printers). These are used for obtaining printout on disk only (see Printer Log, Section 12.3.5). It is also possible to define more than one PRI object for one physical printer. This may be useful if several types of printouts, for example semi-graphic and full graphic, are printed to the same printer.

12.2.2 PRI object notation

From SCIL the PRI object attributes are accessed with the object notation:

PRIn:Bat

where

'n'	Indicates printer number, which can be 1 ... 20. The 'n' may not be omitted from the object notation.
'at'	Attribute name

The PRI attributes of printers defined in another base system are accessed with the following object notation:

PRIn:mBat

where

'm'	The logical application number of an external application (see the TT attribute in Section 7
'n'	The object number of the printer in the other base system

An alternative way of using freely chosen object names is described in Section 5.

12.3 PRI attributes

The common naming attributes BN, BT and BM are described in Section 5.

The PRI specific attributes are described below.

12.3.1 Common printer attributes

12.3.1.1 DC Device Connection

The type of the printer connection. See examples considering DT attribute below to see how to use the attribute for different connection and printer types.

Data type:	Text	
Value:	"NONE"	No printer
	"LINE"	Printer connected directly to the base system computer or to a LAN via a printer server
	"NET"	Printer connected to a NET unit
	"CONSOLE"	Printer connected to LPT1
Default value:	"NET"	
Access:	Read, conditional write	

12.3.1.2 DT Device Type

The type of the printout: color, black and white or transparent.

Data type:	Text keyword	
Value:	"NORMAL"	Black and white printout. No color information is sent to the printer and graphical characters are replaced with ASCII characters. If the printer is connected to a NET unit, the printer must be defined with PRIn:SPT = 1 in the communication unit.
	"COLOR"	Color printout. Color information and all graphic characters are sent to the printer. The printer must be connected to a NET unit where the printer must be defined with PRIn:SPT = 3, 5, 6 or 7.
	"TRANSPARENT"	Printer for printout defined with the SCIL function PRINT_TRANSPARENT. If the printer is connected to a NET unit, the corresponding PRI communication system object must be defined with printer type "TRANSPARENT".
Default value:	"NORMAL"	
Access:	Read, conditional write	

Printers connected directly to the base system computer or to a LAN always produce black and white printout, unless the printers are defined as 'transparent'. Concerning printers connected to NET units, the value of this attribute must coincide with the communication system attribute PRIn:SPT (see Section 18). Some examples of PRI object definitions for different printer types and connections are presented later in this section.

Printer Definitions in the Base System

1. Printer connected to a base system or LAN:
   Defining a 'transparent' printer connected directly to a base system computer via a serial or parallel line or connected to a LAN:
   ```
   PRIn:BTT = "LOCAL"
   PRIn:BDT = "TRANSPARENT"
   PRIn:BDC = "LINE"
   PRIn:BSD = "//SCADA/LP"
   PRIn:BOJ = 1
   ```
2. Printer connected to a NET unit:
   Black nd white ASCII Interface (PRIn:SPT = = 1):
   ```
   PRIn:BTT = "LOCAL"
   PRIn:BND = NET node number
   PRIn:BTN = PRI system object number in NET unit
   PRIn:BDT = "NORMAL"
   PRIn:BDC = "NET"
   ```

   Black and white ASCII Interface with character translation according to the PRIn:SCT attribute, or pixel based, black and white or color printer that uses the EPSON FX-80 or EPSON JX-80 interface (PRIn:SPT = 5, 3 or 7 respectively):
   ```
   PRIn:BTT = "LOCAL"
   PRIn:BND = NET node number
   PRIn:BTN = PRI system object number in NET unit
   PRIn:BDT = "NORMAL"
   PRIn:BDC = "NET"
   ```

12.3.1.3 TT Translation Type

The operating state of the printer. See examples considering DT attribute above.

Data type:	Text keyword	
Value:	"NONE"	Off, out of use
	"LOCAL"	Normal operating state (independently of the type of connection)
	"ALIAS"	The PRI object redirects the printout to another printer determined by the TN attribute, see below.
Default value:	"NONE"	
Access:	No restrictions	

12.3.2 Printer connection attributes

12.3.2.1 ND Node Number

The node number (NOD object number) of the NET unit to which the printer is connected, if the printer is connected to a NET unit. See examples considering DT attribute above.

Data type:	Integer
Value:	1 ... 250, node number of NET unit
Default value:	None
Access:	Read-only, configurable when TT = "NONE"

12.3.2.2 SD System Device Name

The name of the printer or UNC path. The attribute applies to printers that are connected directly to a base system computer or to a LAN via a printer server (DC = "LINE"). See examples considering DT attribute above.

Data type:	Text
Value:	The device name of the printer including UNC path. UNC(Universal Naming Convention) is the way of writing the path to a file or device according to the pattern: \\PrintServer\Printername or \\Wks_267\Install\Setup.exe or \\193.80.81.82\Mydir\Myfile.exe See the Windows manuals
Default value:	None
Access:	Read, conditional write

12.3.2.3 TN Translated Object Number

Printers connected to NET units and printers defined as "alias". For a printer connected to a communication unit, the attribute is the system object number of the printer as known to the actual communication unit, see examples considering DT attribute above. For "alias" printers, it is the PRIn:B object number of the printer to which the printout is redirected (see the TT attribute).

To redirect printout, use the CL attribute that sets the TN and TT attributes automatically.

Data type:	Integer
Value:	1 ... 20
Default value:	No
Access:	Read-only, configurable when TT = "NONE"

12.3.3 Printout attributes

12.3.3.1 CS Control Sequences

Printer control sequences.

Data type:	Text vector
Value:	A text element containing the bytes to be sent to the printer, or a printout vector as described for the PRINT_TRANSPARENT function (the 'data' argument), see the Programming Language SCIL manual. The printout vector may contain texts (control codes), printer control commands and print processor commands as described in the Programming Language SCIL manual, but not vectors. The text values may be given as expressions.
Indexing:	The indices are the printer control commands used by the PRINT_TRANSPARENT function
Suggested values:	Use the following conventions:

1	New Line	Start a new line
2	New Page	Start a new page
3	Reset	Reset the printer settings. This command is automatically executed between two print jobs.
4	Init	Initialise the printer. This command is automatically executed when the printer is started or reset by CL attribute.

Access: No restrictions

This attribute is a vector of printer control sequences (printer commands and control codes) specifying the printer control commands used by the PRINT_TRANSPARENT function, see the Programming Language SCIL manual.

Each element of the vector specifies a control sequence that can be defined as a text value containing the bytes to be sent to the printer. It can also be a vector of text values, printer control commands and print processing commands. A control sequence is referred to by the index of the element. The control sequences are printer interface dependent. However, by using the same conventions for all printers, the PRINT_TRANSPARENT function can be used equally independent of printer type.

If track-keeping of page numbers and line numbers is in use (the LP attribute is set), the print processor (PRNC) tries to count the lines of the through-passing output flow. The New Line as well as New Page commands are accounted for. However, there is no possibility for the print processor to know which control sequences move the paper vertically. Therefore, use the print processor commands Increment LN and Increment PN to notify about such moves (in the CS vector element or in the print vector defined by the PRINT_TRANSPARENT function)

Example:

For example, if the printer is a simple ASCII printer, CS(1) (the control sequence for New Line) should contain the two-byte text calculated as ASCII(13) + ASCII(10) (CR / LF).

12.3.3.2 HF Header Format

The header text produced on each printed page when page numbering is in use (PN > = 0). The text can contain constant as well as variable text.

Data type:	Vector	
Value:	A vector with text elements	
Indexing:	If "COLOR" or "NORMAL" printout:	
	1	The header text. The variables included in the text are indicated with number signs ### where each sign represents a character. The variables are defined with the subsequent elements in the vector.
	2	The value of the first variable in the header text
	3 ...	The value of the second variable in the header text, etc.
	If "TRANSPARENT" printout:	
	A vector of the same format as the CS attribute, see above. The vector is sent to the printer as header text.	
Access:	No restrictions	

Example:

Header for "COLOR" or "NORMAL" printout:

```
HF(1) == "EVENT LIST  ########  PAGE  ###"
HF(2) == "DATE"
HF(3) == "%PN"
```

This HF value might generate the following page header:

EVENT LIST 97-03-20 PAGE 12.

12.3.3.3 LN Line Number

The line count of the printed text, starting from 1. Possible header lines are included in the count.

Data type:	Integer
Value:	Positive integer
Access:	Read-only

12.3.3.4 LP Lines per Page

Number of lines per page.

Data type:	Integer
Value:	0 ... 65535, 0 = no automatic form-feed
Default value:	72
Access:	No restrictions

12.3.3.5 PN Page Number

The page numbering and the generation of headers for printouts. The feature applies to the automatic process printout (event and alarm printout) and printout started by the #LIST command and #PRINT commands if the FORM_FEED variable is = 0. It does not apply to pages printed with !SEND_PIC or #PRINT with FORM_FEED <> 0, nor to pages produced by documentation tools. Such pages do not get any header and are not included in the printout numbering.

If page numbering is in use, a variable %PN that contains the page number is automatically generated and can be used in the header text, the HF attribute, see above.

Data type:	Integer
Value:	0 = The printer has page numbering and header text.
	Negative values = The page numbering and the headers are not in use.
Default value:	0
Access:	No restrictions

12.3.4 Printer queue attributes

12.3.4.1 QM Queue Length Maximum

Maximum number of printout requests in the printer queue. When this length is exceeded, an error message is displayed in the Notification Window and some of the oldest requests are lost. Event channel SYS_EVENT (see the Application Objects manual) is activated to notify the application(s) about the event.

Data type:	Integer
Value:	1... 65 535
Default value:	1000
Suggested value:	100 ... 1000
Access:	No restrictions

12.3.4.2 QU Queue Length Used

Current length of the printer queue. See attribute QM.

Data type:	Integer
Access:	Read-only

12.3.5 Printer log attributes

A printer log is a copy stored on disk of all printouts sent to the printer. Each printer can have its own printer log. The printer log directory is defined by the LD attribute. The name of a printer log file depends on the printer number and the logging period. See the LL attribute below. The name contains the following information ('nn' = PRI object number, 'yy' = year, 'ww' = week number, 'mm' = month, 'dd' = day):

Period:	File Name:	Example:
daily log	prnnyymmdd.log	pr02920201.log
weekly log	prnnyywww.log	pr0292w05.log
monthly log	prnnyymm.log	pr029202.log
yearly log	prnnyy.log	pr0292.log

Storing printout in a printer log is possible even if the printer does not exist physically. Hence, virtual printers, that is, PRI objects without corresponding physical printers, can be defined to get the printout exclusively to printer log files. The maximum size of the files is determined by the computer resources and the operating system.

If a printer has a printer log, everything sent to the printer is copied to the printer log, independent of how the printout is activated (with #PRINT, #LIST or !SEND_PIC, or automatically from the process database) or from which application the printout is activated (several applications can use the same printer). As the printout can contain picture elements, the printer log files contain the same characters as the printout to a black and white ASCII printer.

12.3.5.1 LD Log Directory

The directory where the printer log is stored on disk.

The attribute has no meaning if OD = "PRINTER". The PRI object attribute OD defines the output destination of the log.

Data type:	Text
Value:	0 … 255 characters, see the Programming Language SCIL manual for the allowed characters.
Default value:	"" (an empty text string) which means the directory /sc/sys/active/sys_
Access:	Read, conditional write

12.3.5.2 LF Log Flush Timeout

Log file updating interval.

Data type:	Integer	
Value:	< 0	The operating system controls the buffer and empties the buffer when it is full
	0	The buffering is bypassed and each new log entry is immediately stored on disk
	> 0	The value is the time interval in milliseconds of regular transfer to the log file on disk
Unit:	Milliseconds	
Default value:	1000 milliseconds	
Access:	No restrictions	

Defines how often the operating system buffer is written to the log file on disk. The operating system keeps a RAM buffer of the output to the printout log on disk. There are the following three alternatives:

- The operating system transfers the contents to disk when the buffer is full. In the case of hardware failure (for example a power break), the content of the buffer is lost.
- The buffering handled by the operating system is bypassed, so that each new log entry is immediately stored on disk. This procedure loads the system.
- The buffer is emptied regularly with a selected time interval

12.3.5.3 LL Log Length

The period (day, week, month or year) during which the printer log is written to one file. When a new period begins (always at midnight, 00:00:00), the log switches to a new file.

The attribute has no meaning if OD = "PRINTER". The PRI object attribute OD defines the output destination of the log. See below.

Data type:	Text keyword
Value:	"DAY", "WEEK", "MONTH", "YEAR"
Default value:	"DAY"
Access:	Read-only, configurable when TT = "NONE"

12.3.5.4 OD Output Destination

Defines if the printout to the printer will be copied to a printer log on disk or not.

Data type:	Text keyword	
Value:	"PRINTER"	Printer output only
	"LOG"	Log output only
	"BOTH"	Both printer and log output
Default value:	"PRINTER"	
Access:	Read, conditional write	

12.3.6 Printer control attributes

12.3.6.1 CL Printer Control

Enabling control of the printer and facilitating a redirection of the printout to another printer, for example in error situations.

When redirecting the printout from one printer to another using this attribute, the jobs in the input queue of the printer are moved to the input queue of the new printer.

Setting the attribute is meaningful only if the printer is in normal operating state (TT = "LOCAL").

Data type:	Text or integer	
Value:	"CANCEL"	Cancels the current job (format, screen or picture document) and starts the next
	"REPRINT"	Restarts the current print job
	"RESET"	Cancels the current job and all jobs queued for the printer. The printer will continue with the next new job.
	"STOP"	Cancels all jobs and stops the printer. The TT attribute is set to NONE.
	"PAUSE"	Stalls the printer until a "GO" is issued
	"GO"	Continues after PAUSE
	n	Redirects the printouts to printer number 'n' (integer). The current and queued jobs are output to the log file of the present printer (if any), but not to the printer. The TT attribute of the printer is automatically set to "ALIAS" and TN to 'n'.
Access:	Write-only	

Example:

```
#SET PRI5:BCL = "STOP"
```

12.3.6.2 OJ Open on Job Basis

States whether the printer will be opened (reserved) once and kept open continuously, or opened and closed on job basis.

Data type:	Integer	
Value:	0	Open printer once (when the printer is started)
	1	Open and close printer on job basis. The printer is opened before each print job and closed when the job is completed.
Default value:	0	
Suggested value:	Use value 1 when a printer connected to a base system or a LAN will be used by more than one base system	
Access:	Read-only, configurable	

12.3.6.3 ST Printer State

The present state of operation of the printer.

Data type:	Text keyword	
Value:	"IDLE"	No jobs to print
	"BUSY"	Printing a job
	"WAITING"	The system is waiting for the printer, for example, when the printer buffer or NET unit buffer is full.
	"STALLED"	Stalled due to a PAUSE command (#SET PRIn:BCL = "PAUSE")
	"STOPPING"	Finishing before stopping (after a "STOP" command)
Access:	Read-only	

12.3.7 Miscellaneous PRI attributes

12.3.7.1 CX Comment Text

A freely chosen comment text.

Data type:	Text
Value:	Any Unicode text, up to 255 characters
Access:	No restrictions

12.3.7.2 SV System Variables

Global variables used in tools, etc. The attribute should not be used in application programs.

Data type:	Vector
Value:	Defined by Hitachi Power Grids
Default value:	A vector of 10 integer 0’s
Access:	No restrictions

12.3.7.3 UV User Variables

Global variables for use in application programs.

Data type:	Vector
Value:	Defined by the application
Default value:	A vector of 10 integer 0’s
Access:	No restrictions

Section 13 Base System Object Navigator

13.1 About this section

This section describes the **Base System Object Navigator** tool located in the **SYS600 Tool Manager System Configuration** page.

The tool handles the following base system objects:

- Base object (SYS)
- Applications (APL)
- Monitors (MON)
- Links (LIN)
- Nodes (NOD)
- Stations (STA)
- Station types (STY)
- Printers (PRI).

13.2 General

The Base System Object Navigator tool is an on-line tool that provides the following common functionality:

- Recognizing of the base system objects in SYS600 system.
- Viewing of the base system related attributes and their values.
- Editing of the base system object related attribute values (see certain limitations further in this section).
- Adding of base system objects.

Because the tool is an on-line tool, the modified attribute values or added base system objects affect only the running system. If there is need to configure the system permanently, the changes should be made to base system configuration files (e.g. SYS_BASCON.COM).

13.3 Recognizing of the base system objects

During start-up, the tool reads all the found base system objects regarding to the system configuration. For each found base system object, the object related set of base system attributes is read.

13.4 Viewing and editing of the base system objects

13.4.1 Base system objects

In the main view of this tool, all of the above mentioned base system objects are presented in hierarchical way in an object tree. Navigation in the object tree, e.g. expanding the Applications category, shows a set of leaves, each of which represent one application that was located from the SYS600 system. The shown leaves in the tree are identified by additional information to provide easier location of base system object under user's interest. Depending of the base system objects, the following identification is provided in the tree:

- Application number, name, operating and application states. In applications of operation state PROXY application numbers constituting a HSB pair are shown instead of application state
- Monitor number, monitor type and application number into which the monitor is connected
- Node number, station address of the node and LAN node name (if any)
- Station number and station type
- Number of station type, name of the type, database interface and comment text (if any)
- Printer number, printer type and operating state

The number, name, type and states are displayed either by using textual information or using different type and/or color in icon representation. The current application and monitor are marked by asterisk. Asterisk stays after the application and monitor number.

13.4.2 Base system object attributes

In the main view of this tool, all the base system object related attributes and their values are also presented in hierarchical way in attribute tree. For example, selecting of one leaf under certain category lists all the base system attributes and values regarding the selection. The amount of shown attributes depends on the base system object type. Each unique attribute belongs to a group of attributes. The used group names are identical to those used in the documentation of SYS600 system (System Objects manual), e.g. for Application base system object:

- Base system object naming attributes
- Basic application attributes
- Supervision configuration attributes
- Path, representation and text database attributes
- External and alias attributes
- Mapping attributes
- Shadowing attributes
- Resource handling attributes
- Functional definition attributes
- User interface related attributes
- Operation control attributes
- Loop control attributes
- Revision compatibility attributes
- Application diagnostics attributes
- Mirroring configuration attributes
- Mirroring diagnostics attributes
- Proxy application attributes
- Miscellaneous application attributes

Attributes may also be shown in alphabetical order by selecting **All Attributes in Alphabetical Order** from the **View** drop-down list. Each unique attribute is displayed with its two-char identification name (e.g. AN, AS or HP), together with its attribute description (e.g. Application Name, Application State or History Logging Policy) and data type identification (in the form the different type in icon representation). The attribute values are presented with or without attribute value descriptions (e.g. 10, or 1 - Enabled). If an attribute value has a certain unit for the values, the unit information is also included in the end of attribute description inside the brackets (e.g. Shadowing Flush Time [ms].

When an attribute is selected in the tool, a help text of the selected attribute is displayed in the information bar of the tool.

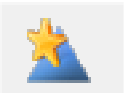
13.4.3 Base system object attribute values

In the main view of this tool, the base system attribute values are also presented, and they may be modified in the Edit Attribute area. When a base system attribute is selected under a certain group name, its current value is presented. The user can then use the appropriate dialog items to modify them. For example, when an attribute value contains certain keywords, the possible keywords are presented in a drop-down list from which the user may select an item. The role of the Edit Attribute area is to contain verification of the attribute value. If the user enters a non-valid value, the previous valid value is silently returned. If the attribute value is formally right, but it conflicts somehow with the system, the appropriate message box is displayed to the user as a result of non-valid data entry.

Limitations: In the edit attribute area, it is not possible to insert new elements into complex data types (vectors and lists, e.g. Global Paths, Representation Library and Text Database attributes of APL). Also, the existing element cannot be removed. The capabilities of the edit area are to display the data structures as static and edit values of existing elements. However, it is possible to modify complex data types in the edit area for structured attributes. In that area, data is presented as a SCIL expression. The edit area for structured attributes is shown when a node of a complex attribute is selected in the attribute tree Figure 10. Note that the simple data type BYTE_STRING can only be modified in the edit area of structured attributes.

13.5 Adding a new base system object

When adding a new base system object, first select the category of the base system object in the object tree of the tool. E.g. when there is need to add a new application, the Applications category is selected first. Then, select **Object** ->**New** from the menu, enter keystroke CTRL+N or click the **New Object** toolbar button.



New Object dialog is displayed and an object number is requested from the user. When adding a monitor object, a monitor type is required. Depending on the base system object type to be added, different set of attributes need to be specified by the user before the base system object adding may proceed. For example, at least the following information needs to be specified for a new application object:

- Application name or
- Operating state (Translation Type, TT) attribute.

If an error occurs during the creation of base system object, an appropriate message box is shown to the user with additional status information.

13.6 Starting the tool

To start the Base System Object Navigator tool, click the Base System tool icon in the **SYS600 Tool Manager System Configuration** page and select **Open** from **File** menu or double-click the tool icon Figure 7.

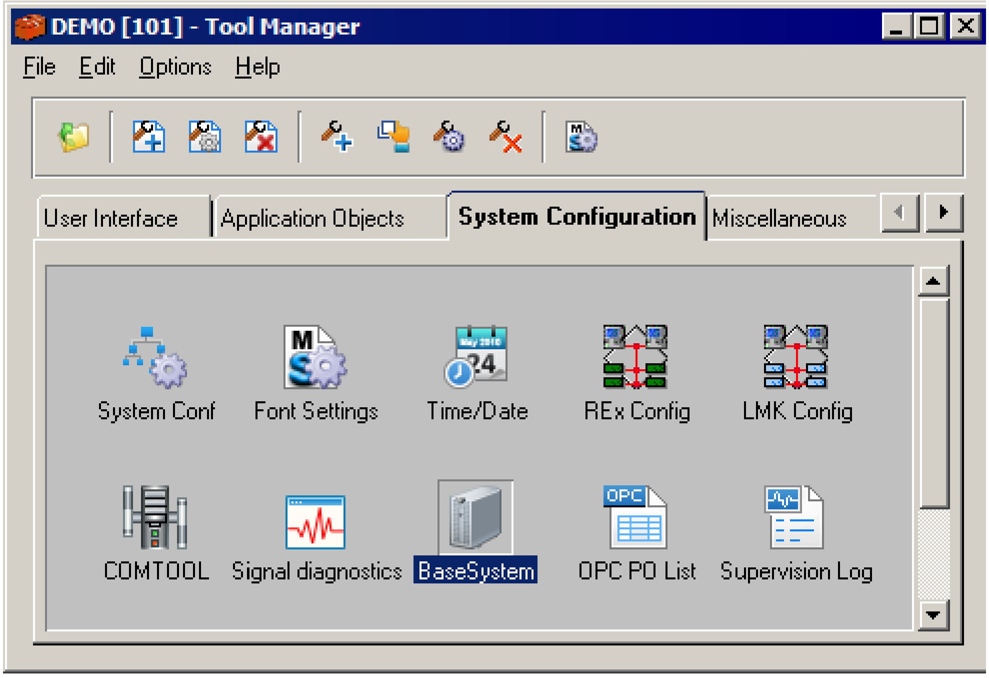


Figure 7: Base System Object Navigator tool on the System Configuration page of Tool Manager.

The **Base System Object Navigator** tool page includes a menu bar and a toolbar, which can be selected from the **Options /Toolbar** . Below the toolbar, there is an object tree on the left, an attribute tree in the middle and an attribute editing area on the right hand side. In addition, there is an information text bar at the bottom of the page.

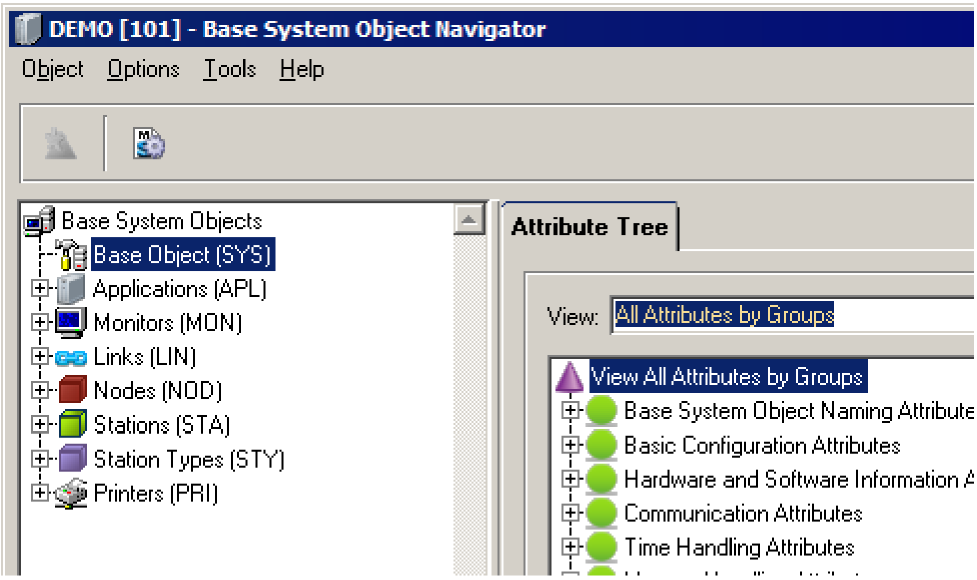


Figure 8: The menu bar, tool bar, object tree and attribute tree in the Base System Object Navigator tool.

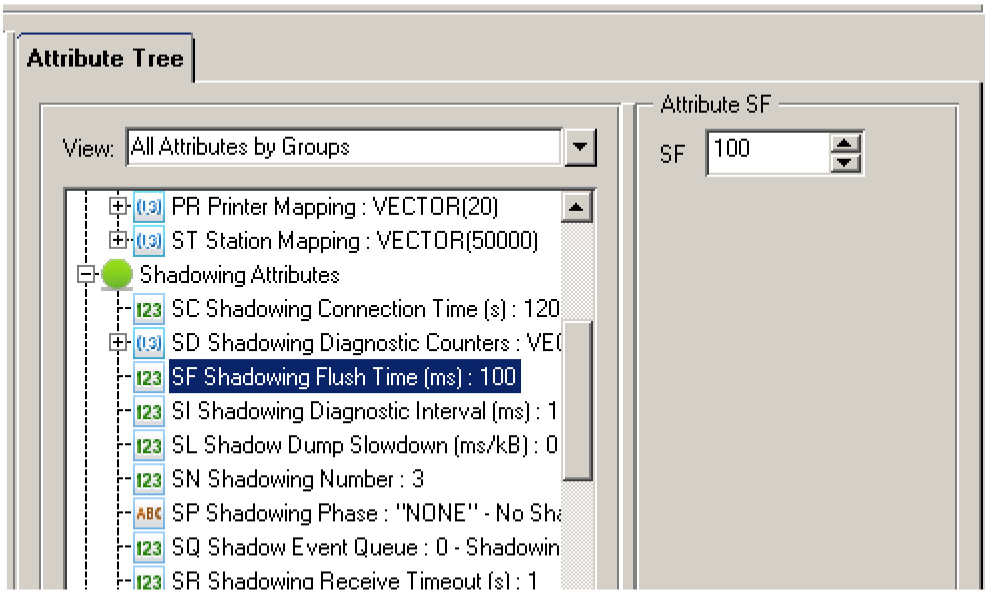


Figure 9: The attribute area and attribute edit area in the attribute tree of the Base System Object Navigator tool.

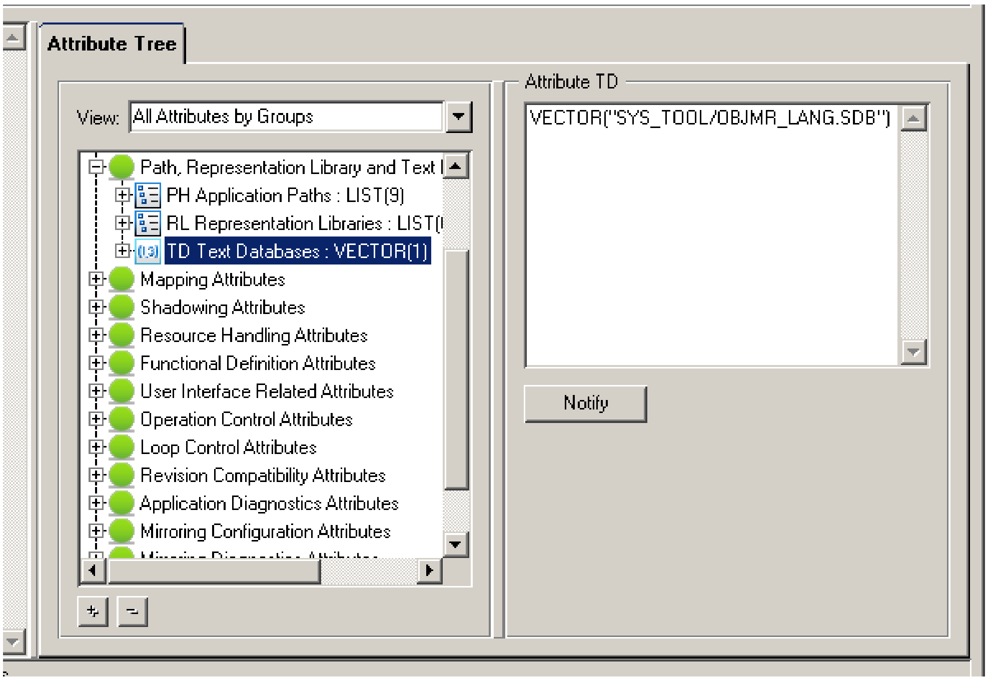


Figure 10: The attribute area and edit area for structured attributes in the attribute tree of the Base System Object Navigator tool. By clicking Notify button after editing saves the given value.

13.7 How to handle the object and attribute trees

When an object is selected from the object tree, all the attributes linked to it are shown in the attribute tree (see Figure 8). The working order is from left to right: after selecting an object in the object tree, an attribute can be selected in the attribute tree and the selected attribute can be edited in the attribute editing area.

A tree can be expanded by clicking the **+** sign on the left or by double-clicking the text area on the right. Likewise, the tree can be collapsed by clicking the **-** sign or double-clicking the text area. The **-** sign means that the branch of the tree cannot be expanded any further.

The whole attribute tree can be expanded and collapsed using the **+** and **-** buttons that are situated below the tree or by right-clicking the tree and selecting **Expand All** or **Collapse All** from the pop-up menu (see Figure 14).

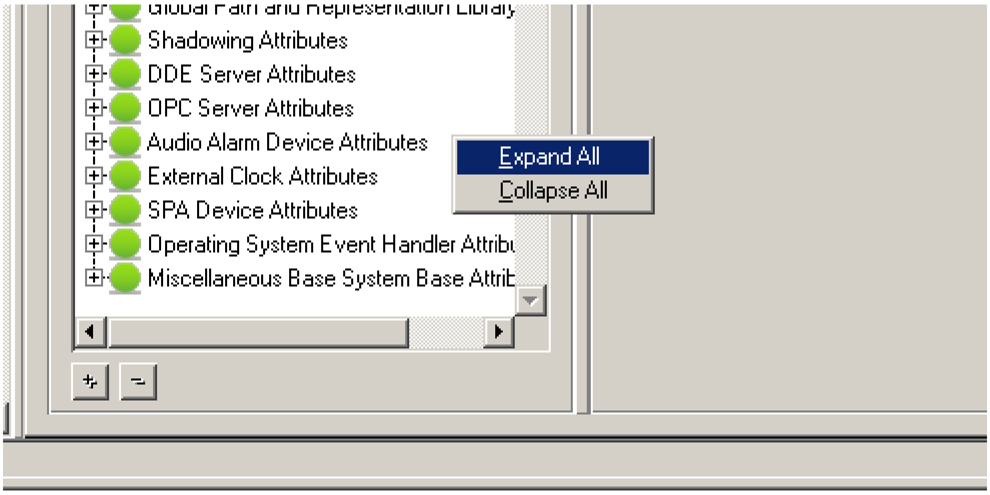


Figure 11: The Expand and collapse buttons and pop-up menu for the attribute tree.

13.8 Tools in the tool

The following tool is introduced to ease the management of shadowing applications:

- Hot Stand-By Management tool.

Start the management tool either by selecting **Base Object** from the object attribute tree or by selecting **Tools/HSB Management** . The menu is enabled if the shadowing attribute SH of Base Object is on.

The tool has two pages

- **Packages**
- **Shadowing Applications**.

The **Packages** page indicated the versions and statuses of the installed package and disk package. Install the disk package by clicking the **Install** button (see Figure 12).

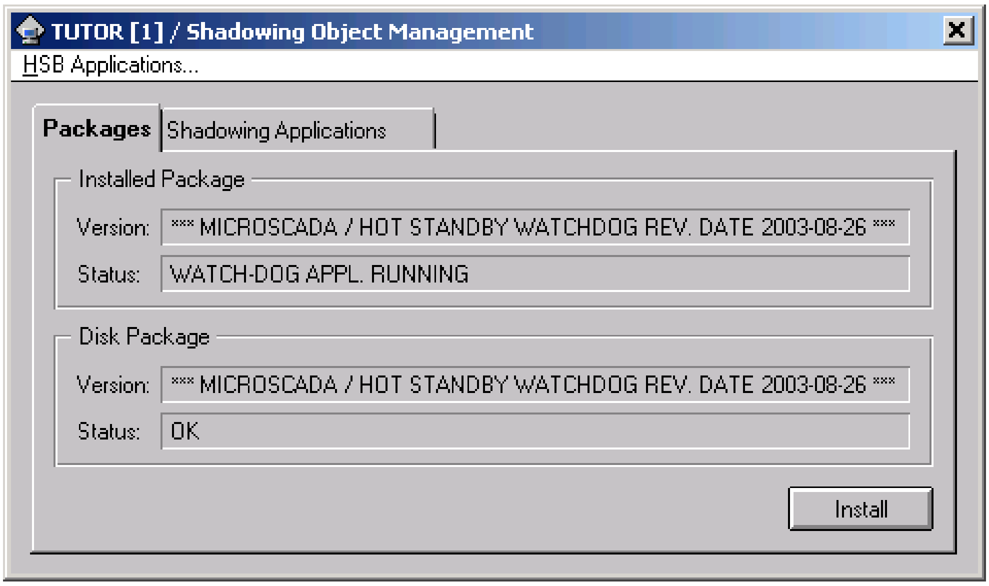


Figure 12: The Packages page of the HSB Management tool.

The **Shadowing Applications** page shows information from the local watchdog application and shadowing applications (see Figure 13).

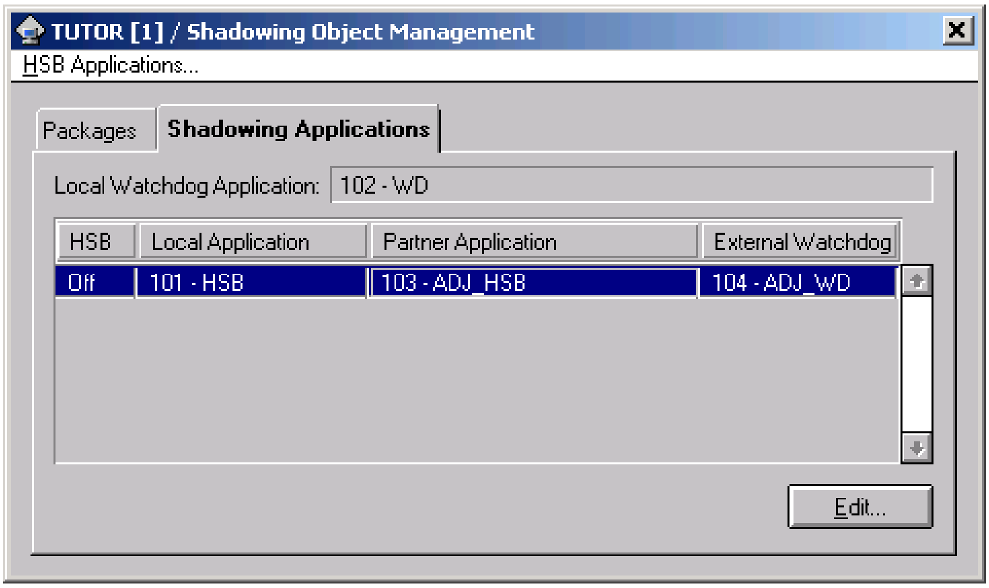


Figure 13: The Shadowing Applications page of the HSB Management tool.

Selecting a shadowing application and clicking the **Edit** button starts the edit dialog of the application (see Figure 14).

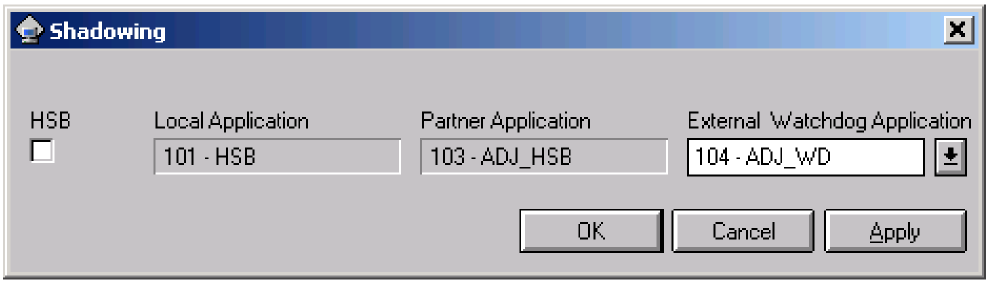


Figure 14: The edit dialog of a shadowing application.

Section 14 Communication system objects, overview

14.1 About this section

This section introduces the communication system objects:

Section 14.2	General: This section gives a general description of communication system objects, their features and functions.
Section 14.3	Communication system object types: The types of the communication system objects. The types are illustrated by a picture and each type is described briefly.
Section 14.4	Defining communication system objects: The principles for defining communication system objects, the configuration software and the required definitions.
Section 14.5	Attributes: The function of the communication system attributes and the attribute access levels.

14.2 General

14.2.1 Features

The communication system objects and their attributes specify the NET unit configurations and handle the process communication. Each NET unit (NET) has its own set-up of system objects. The communication system objects give the NET unit an image of the communication lines and the connected devices. The connected devices may be:

- Other communication units
- Base systems
- Applications in the connected base systems
- Stations (RTUs, PLCs, protection equipment, central stations, etc.)
- Printers

During operation, the communication system objects are included in the communication program running in the host PC.

14.2.2 Function

Each device connected to the process communication system (base system, application, communication unit, printer and station) has a device image. The device image is in form of a communication system object in the NET unit to which it is directly connected. The communication system object attributes of the unit describe the data communication with the devices.

The base systems and communication units are regarded as nodes. These devices are defined by NET or NOD objects in each NET unit where they are to be recognized, even if they are not directly connected to it (see Figure 15). As the node numbers are recognized by each communication unit, a message can be routed to a device connected to another node.

The application engineer can configure the communication network, define routes, control I/O devices and RTUs, read diagnostic counters, handle system messages, etc. using the communication system objects and their attributes.

14.3 Communication system object types

14.3.1 Communication system object names

The system object names consist of a predefined three-letter descriptive name, and an object number that distinguishes units of the same type from each other:

NODn or NETn	Node objects: communication units and base systems, 'n' = 1 ... 250
APLn	Application objects, 'n' = 1 ... 250
STAn	Station objects, 'n' = 1 (0) ... 255 (each station type has its own number series)
PRIn	Printer objects, 'n' = 1 ... 8

The 'n' here is the device number as known to the NET unit (integer). In this manual, the number is called a communication system object number or, where no misunderstanding can occur, simply object number. The object number must be unique for a specific object type within a certain communication unit, except that STA objects of different station types can have the same numbers. The NOD (NET) object numbers are global node numbers and must be unique within the entire SYS600 network.

When accessing the STA and PRI communication system objects from SCIL, the object number in the object notation is the number under which the device is known to the application. The APL objects are accessed from SCIL through the NOD (NET) objects.

Alternatively, communication system objects of type NOD, STA and PRI may be accessed by their user-defined name, i.e. the name of the corresponding base system object. For example, if the name of base system object STA5 is PICCADILLY_STATION (STA5:BBN == "PICCADILLY_STATION"), then notations STA5:S and PICCADILLY_STATION:S are equivalent. See Section 5 for naming base system objects.

Figure 15 illustrates the system object types and their roles in the system. After that follows a brief presentation of each system object type.

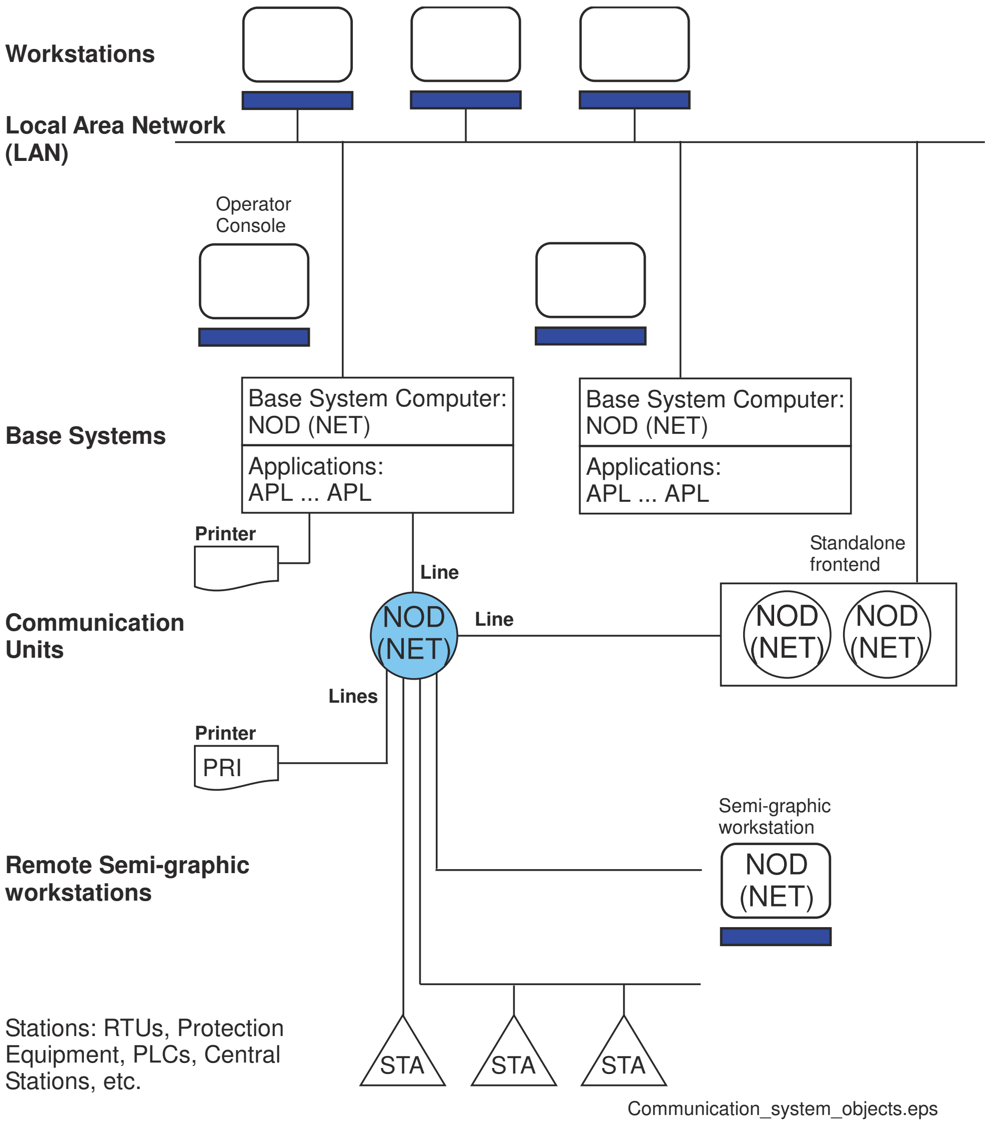


Figure 15: The communication system objects defined in the shaded NET unit that is a PCNET running in the base system. The unbolded and unmarked devices require no object definitions in the NET.

14.3.2 NOD (NET) objects

The NOD or NET objects (the names are equivalent) represent the NET unit itself as well as every other known node (communication units, base systems) in the SYS600 system. The object numbers of the NET objects are global and may be 1 ... 250.

The SYS600 system engineer uses the NET objects and their attributes for example for the following purposes:

- Defining the NET itself, e.g. its station address and the system message handling.
- Adding, modifying and removing communication system objects definitions for connected devices.
- Accessing application object attributes.

By using indices in the NET object notation, the NET line attributes are accessed. The index refers then to the line in question. The system engineer uses the NET line attributes, for example, for the following purposes:

- Defining the NET lines by choosing protocols for the communication lines.
- Defining communication line features (baud rate, number of stop bits, etc.).

In this manual, the NET device attributes and the NET line attributes are described in different sections.

14.3.3 APL objects

The APL communication system objects refer to applications in the connected base systems. Each application known to the NET unit must be defined as an APL object. A NET unit can recognize up to 250 APL objects (applications).

The attributes of the application objects are accessed using the NOD (NET) object notation and giving the APL object number as the index.

14.3.4 STA objects

This system object type is used to supervise and control properties of connected stations: RTUs, protective equipment, central stations, etc. Each station (logical or physical) connected to the NET unit is usually defined as a STA object. The STA objects specify and handle, e.g.:

- The data transfer to and from stations
- Diagnostics (type of diagnostic commands used, diagnostic counters)
- Timeouts, addresses and other device specific parameters
- Operational status (connected/not connected, in use/not in use)

The STA objects are defined in different ways and have different attributes depending on the station type they represent. See protocol specific sections and manuals for more information.

One NET unit can be connected to up to 2047 stations and the configuration is made using System Configuration Tool or SCIL scripts. See SYS600 System Configuration for more information.

14.3.5 PRI objects

Each printer connected to the NET unit must be defined as a PRI object. The PRI objects describe printer features, such as:

- Printer type (colour printer, black and white printer, etc.)
- Operational status (In use or not in use)

14.4 Defining communication system objects

14.4.1 Principles

Connecting a new device to a NET unit requires that the corresponding system object is defined in that unit. This is a condition for the physical connection to work. Defining a new communication system object means that it is assigned a type specific object number and a line number. The procedure is:

- Defining the line to be used by assigning it the desired protocol (the PO attribute, see Section 16).
- Giving the line its communication properties using the line attributes (Section 16).
- Creating the object by giving it an object number and assigning it the line number.
- Setting the attributes of the created object.
- Taking the line and the device into use.

The configuration of the NETs can be extended and modified online using SCIL or tools.

In SCIL, the objects are defined using the #SET command. New communication system objects are created with the device specific NETn:S attributes NE, SY, DV, ST, RT, etc. which are described in Section 15. The communication system objects cannot be created, deleted or modified with the SCIL commands #CREATE, #DELETE and #MODIFY.

If a PC-NET unit is stopped and restarted, System Configuration Tool loads the saved configuration automatically for the unit.

14.4.2 Initialization file of PC-NET units

Each PC-NET unit has an initialization file (PC_NET.CF1) which defines the NET unit itself, the host base system and an application in the host base system. The initialization file is a text file that is read each time the PC-NET unit is started. This file is used by the System Configuration Tool and should not be edited by the user.

14.4.3 On-line configuration

In addition, the communication system objects can be written online with application dependent command procedures and pictures, with SCIL or with tools. For example, they can be written with command procedures started by the event channels APL_INIT_1, APL_INIT_2, and APL_INIT_H, see the Application Objects manual. It is also possible to build command procedures that are started at each start-up of a NET unit.

14.4.4 Required Object Definitions

Table 2 provides an overview of the communication system object definitions required by various configuration projects. It also refers to the sections and sections where the attributes are detailed. The communication system objects required for various system set-ups are detailed in the System Configuration manual.

Table 2: An overview of the communication system object definitions required for various configuration projects

Configuration	**Basic Definition**	**Other definitions and properties**	
project		**Object notation**	**Attributes**
Defining NET unit to itself	Node number (NN) Set in the configuration file (PC_NET.CF1) of PC-NET units	NETn:Sat ‘n’ = node number ‘at’ = attribute	The NET attributes in Section 15
Defining NET lines	PO attribute = protocol	NETn:Satl ‘l’ = line number	Line attributes in Section 16.
Defining connected NET units	External nodes NE attribute = object number	NETn:Satm ‘m’ = node number of NET unit	NET attributes in Section 15.
Defining connected base systems	External nodes NE attribute	NETn:Satm	NET attributes in Section 15.
Defining applications	NET attribute SY	NETn:Sata ‘a’ = application number	NET application attributes, Section 15.
Defining connected stations	NET attributes in Section 15.	STAn:Sat	STA object attributes in Section 17.
Defining connected printers	NET attribute PR in Section 15.	PRIn:Sat	Printer attributes in Section 18.

14.5 Attributes

14.5.1 General

There are two types of system object attributes:

- Configuration attributes, which affect the system configuration of the NET unit but do not cause any immediate data exchange with connected devices.
- Communication attributes, which cause data communication between different devices but do not affect the system configuration.

14.5.2 Attribute access

The communication system object attributes are of the following main access levels:

- Read-only: The attribute cannot be written. There are still a few exceptions in which the values can be reset.
- Read, conditional write: The attribute can be given values with #SET provided that the object first is taken out of use (IU = 0).
- Write-only: The attribute cannot be read, only written with the #SET command. It is not stored in NET unit, but transmitted directly to a station.
- No restrictions: The attribute can be read and written without restrictions.

The terms above are used in the subsequent attribute descriptions.

Section 15 NOD (NET) objects for communication system

15.1 About this section

This section describes the NOD (NET) objects and their attributes.

Section 15.2	General: The definition of NOD (NET) objects and the object notation.
Section 15.3	Basic NET attributes: Basic Definitions (NN, SA, SX), Functional Specifications (TL), NET Information (KP, NT, VE), System Message Attributes (MI, MS, SE, UI, UA), Connected Nodes (AD, LI).
Section 15.4	Object definition attributes: External Nodes (NE), Applications (SY), Station Definition Attributes (LC, LM, PC, RT, RX, SM, SP, ST), Adding Devices of Exchangeable Device Types (DV), Printer Definition Attributes (PR).
Section 15.5	Application attributes: DS, SU, SW.
Section 15.6	Authentication attributes: KS.
Section 15.7	Miscellaneous NET device attributes: FM, LT, MA, MU, TM.

The NET attributes related to the communication lines are described in the next section.

15.2 General

15.2.1 NET object definition

The following devices must be defined as NOD (NET) objects in a NET unit (NET):

- The NET unit itself. This is done by the System Configuration Tool for each created NET unit.
- All other communication units which will be recognized by the unit.
- All base systems, which will communicate with the unit.

The NET unit itself must is defined in the initialization file if of the PC-NET. The other nodes (external nodes) are defined by setting the NE attribute (see Section 15.4) of the NET node.

The system object numbers of the nodes (node numbers), 1 ... 250, must be unique within the entire SYS600 network. For communication units and base systems, they must be the same as the node numbers defined with the NOD base system objects (NODn:B) (see Section 10). The node numbers can be taken into use in any order.

In case the node number given to PC-NET node is bigger than 50, the default addresses of the general NET Messages and application diagnostic messages defined with NET node attribute MI may overlap with some other NET node having a number = node number - 50. If this happens, a new, not overlapping value must be given to attribute MI. The object addresses of the SYS_NETD:Px created by the System Self Supervision must be re-entered manually.

15.2.2 Object Notation

From SCIL, the NET attributes of the NET unit or a CPI-based communication module is accessed with the notation (the NOD and NET names are equivalent):

NODn:Sat (or NETn:Sat)

where

'n'	The object number (= node number) of the unit, 1 ... 250
'at'	An attribute name

The NET attributes of other nodes (external nodes) defined in the communication unit, that is, other communication units and base systems are accessed with the notation:

NODn:Satm (or NETn:Satm)

where

'n'	The node number of the NET unit where the object is defined.
'm'	The node number of the base system or communication unit. The application attributes are accessed in an analogue manner.

15.3 Basic NET attributes

15.3.1 Basic definitions

15.3.1.1 NN Node Number

The NET object number. The number is defined when the PC-NET unit reads its configuration from the PC_NET.CF1 file. An external NET object is created with the NET attribute NE (Section 15.4). As a rule, the attribute should be regarded as read-only.

Data type:	Integer
Value:	1 ... 250
Access:	Read-only, configurable

Example:

The NET number of an external NET (base system or communication unit) is changed (not recommended):

```
#SET NET1:SNN2 = 3
```

15.3.1.2 SA Station Address

The ACP station address of the NET object (communication unit or base system). The station address is used in all communication between nodes. The address must be unique among all NETs and base systems in the entire network. At online station address configuration the communication program checks that this uniqueness is maintained.

Data type:	Integer
Value:	1 ... 254
Indexing:	If no index is used, the attribute means the station address of the NET itself. If an index or index range is given, the attribute value is the station addresses of the external NET node specified by the index.
Access:	No restrictions

Example:

Defines that the NET2 connection in NET1 will have the station address 27:

```
#SET NET1:SSA2 = 27
```

15.3.1.3 SX X3.28 Station address

The station address of the NET used in the communication with ANSI stations. The address is used as the source address in messages to the stations and should be given as destination address in the stations. It must not be the same as a station address used by any connected ANSI station. However, it may be the same as the ANSI address of stations connected to other NETs. The attribute has no meaning if no ANSI station is connected to the communication unit.

Data type:	Integer
Value:	1 ... 254
Default value:	The default value of the SA attribute of the node
Access:	No restrictions

15.3.2 Functional specifications

15.3.2.1 TL Translation Limit

The maximum number of logical destination translations accomplished on the same message. The attribute concerns node data communication (NET-NET communication), that is communication with other communication units and base systems.

This attribute prevents messages from circulating eternally in a network composed of several nodes (for example because of some configuration error).

Data type:	Integer
Value:	1 ... 255
Default value:	10
Suggested value:	5 (normally there is no need to change the default value)
Access:	No restrictions

15.3.3 NET information

15.3.3.1 KP Known Protocols

The protocols supported by the NET in question.

Data type:	Vector
Value:	Vector of integer elements representing the number codes of known protocols, see the PO attribute in the Section 16.
Access:	Read-only

Example:

When NET node attribute KP is read, vector with the following elements is returned (1,2,3,4,7,9,12,13,14,15,16,17,18,23,25,26,27,28,30,31,32,33,35,41,43,44,45)

15.3.3.2 NT Node Type

The type of the node: NET, base system, workstation or gateway. Attribute is not supported by the communication modules using CPI-library.

Data type:	Integer	
Value:	0 ... 6:	
	0	Unknown
	1	Base system
	2	NET (communication unit)
	6	Gateway
Default value:	2	
Access:	Read-only	

15.3.3.3 VE Program Version

The version number and generation date of the NET program.

Data type:	Vector	
Value:	Vector of three elements	
Indexing:	1	Version number as a text
	2	Generation date as a text
	3	Program generation time as an integer counted from the formula 10000 * hours + 100 * minutes + seconds
	No index	Index 1
Access:	Read-only	

15.3.4 System message attributes

The attributes of this section affect the transmission of system messages from the NET unit to one or more applications in one or more base systems.

The system messages are integer values, which inform an application about changes in the device communication. System messages are generated by STA, PRI and NET objects at the appearance and disappearance of abnormal situations. A system message contains a status code, which describes the state of the device (see the manual Status Codes). As a rule, the status code in a system message is zero, if the message indicates recovery from an error situation, otherwise non zero (an exception is the application messages, see below).

Communication units generate system messages, for example, in the following situations:

- The communication program has started: code 10001.
- Various situations on communication lines: protocol dependent codes.
- Various situations on dial-up lines: a number of messages.
- Changes in the handshaking signal states on ANSI X3.28 Full Duplex lines.
- The APL communication is suspended or recovered (see the SU attribute, Section 15.5): suspension = application number, recovery = 1000 + application number (the application number as defined to NET unit).
- The MU attribute, Section 15.7, has been updated: code 16633.

In the process database of an application the attributes MS and MT direct the communication system messages from NET unit to a fictitious process object. The transmission of system messages from NET unit can be enabled or disabled using the SE attribute. The status codes of the system messages can be used in the application (specified by the MS attribute) as follows:

1. Create a fictitious analog process object (AI) with the object address OA = the MI attribute below. The system message codes of the device will be registered as the object value of this process object.
2. Define the consequential operations by means of event, alarm and printout attributes, see Figure 16. For more information, see the System Configuration manual.

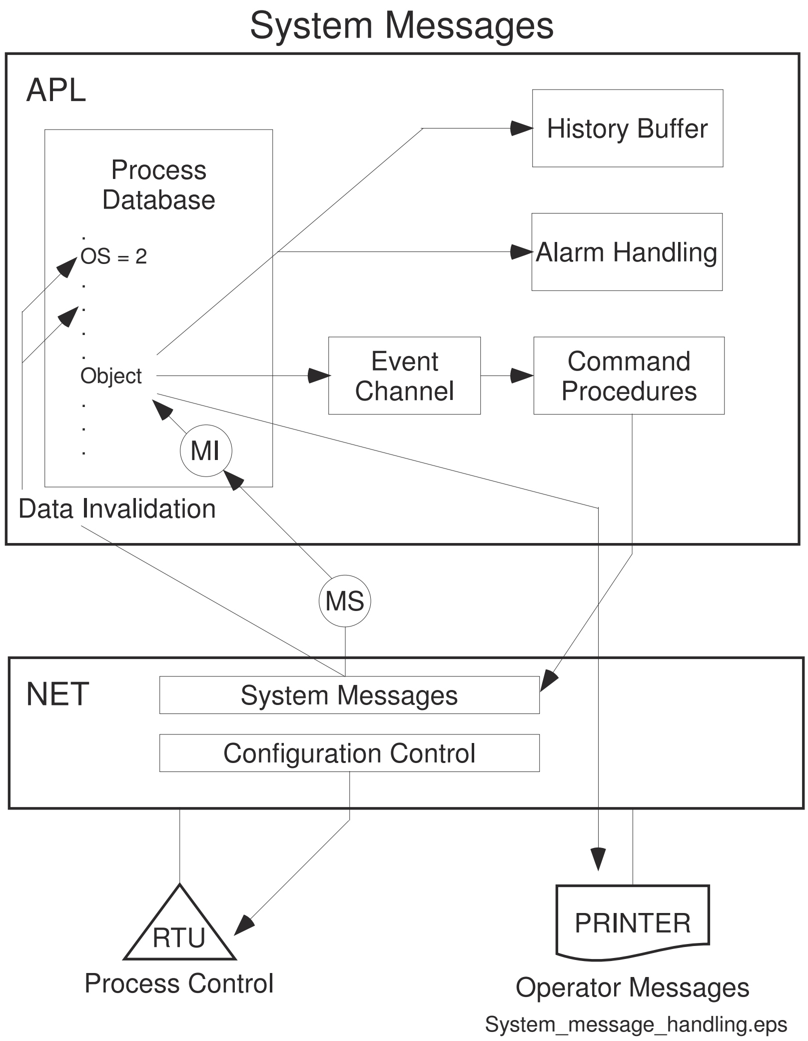


Figure 16: System message handling

15.3.4.1 MI Message Identification

The object addresses (the OA process object attribute) of the process objects, which will receive the system messages generated by the device. At the generation of a system message, the status code of the message is updated in the OV attribute of the process objects. The process objects are specified by the MI attribute in the application specified by the MS attribute. The status code is not registered if there are no fictitious process objects with the specified address. NET unit generates two types of system messages, which are directed to process objects as follows:

General NET messages: MI

Messages related to application diagnostics: MI + 50

If the NET node attribute SE (System messages enabled) has a value 4, a binary process object is updated at the same time as the analog process object mentioned above. The binary process object must be of type ANSI Binary Input and its unit number UN must also be 0. The address of this process object is MI+16777216 (MI+1000000hex) and its OV attribute has values 1 (station is OK) or 0 (station is not OK). See the example at the end of this description.

In case the node number given to PC-NET node is bigger than 50, the default addresses of the general NET Messages and application diagnostic messages defined with NET attribute MI may overlap with some other NET node having a number = node number - 50. In case the node number given to PC-NET node is bigger than 100, the overlapping may occur with the NET line supervision objects, too (see formula below). If either of these happens, a new, not overlapping value must be given to attribute MI. The object addresses of the supervision process objects created by the System Self Supervision (SYS_NETD:Px) must be re-entered to match with the given MI value.

Data type:	Integer	
Value:	1 ... 32760	
Default value:	The following default values are also the recommended values:	
	NET objects:	6000 + NET number
	The MI attribute of other devices have the following default values (recommended values):	
	NET lines:	6000 + (100 * NET number) + line number
	STA, ANSI:	1000 + station number
	STA, SPIDER RTU:	8000 + station number (NET messages)
		8500 + station number (terminal status)
		9000 + station number (terminal event)
		9500 + station number (terminal message)
	STA, SPACOM:	1000 + station number
	STA, P214:	1000 + station number (NET messages)
		1500 + station number (terminal status)
	Exchangeable device types:	1000 * type code + object number
	PRI:	3000 + printer number
	These default values can be used as such (copied to the process object address), or they can be changed.	
Access:	No restrictions	

Example:

The system message receiving process object in application 3 should have the following attribute features:

```
#SET STA2:SMS = 3
```

Logical Name:	Any
Unit number:	UN = 0
Object Address:	OA = the MI attribute of the system object
Type:	AI (ANSI Analog Input)

If the NET node attribute SE (System messages enabled) has a value 4, following process object is also updated:

Logical Name:	Any
Unit number:	UN = 0
Object number:	OA= the MI attribute of the system object+16777216 (MI+1000000hex)
Type:	BI (ANSI Binary input)

15.3.4.2 MS Message Application

The communication system object number of the application, which receives the system messages from the device. If the application is suspended, the message is sent to the first application that is not suspended in number order.

The NET start-up system message code (10001) is sent to all known applications, independently of the MS attribute.

Data type:	Integer
Value:	Integer in the range 1 ... 250. Application number as known to the communication unit.
Access:	No restrictions

Example:

If MS == 3, the system messages related to the NET unit are sent to the application defined as number 3 in the NET unit

15.3.4.3 SE System Messages Enabled

Makes it possible to disable the system messages of a NET unit, state whether system messages related to a NET unit are to be sent to the base system or not. When the sending of system messages is disabled, the messages (up to 20 per object) are queued in the communication unit. The queued messages are sent all at a time when the transmission of system messages are enabled.

The address of the analog status process object is defined by the MI attribute. The corresponding binary status object has an object address OA = MI+16777216 (MI+1000000hex), OB = 0. The meaning of the binary status object is to indicate whether the line or the application is OK or not (1=OK, 0=not OK).

Data type:	Integer	
Values:	0	System message generation disabled
	1	System message generation enabled (analog status points updated)
	2	NET start-up system messages enabled, all other system messages disabled.
	3	System message generation enabled (analog status points updated). See note 2 below.
	4	System message generation enabled (analog and binary status points updated). See notes 1 and 2 below.
Default value:	4 (1 in SYS600 version 9.1 or older)	
Access:	No restrictions When a value 1, 3 or 4 is written to the SE attribute, the statuses of all station objects and lines are resent to the specified applications in the base system.	

Note 1:	SE value 4 can not be used in SYS600 version 9.1 or older.
Note 2:	With values SE=3 and SE=4, the APL_EVENT event channel events "RUNNING" and "SUSPENDED" are generated with all station types.
Note 3:	The default station type defined with SYS:BDS may have an effect on the default OB value of the created binary status point. If the created point is not updating as expected, the OB attribute should be checked and modified to 0 if non-zero.

Changing the default value of SE attribute is not recommended. If changed, it is possible that all tools do not work anymore.

15.3.4.4 UI UAL event Identification

The UI attribute is used to define the name for the node for the UAL (User activity logging) events. It is used to identify the source of the UAL events. Since this string is added to the identification information of all user UAL events from this node, a unique value within the system is preferred. In case the source of the event is a STA object or a line object within the node, identification information will also contain the contents of the station or line level attribute UI. In case the event is generated from the node level, the identification information is formed based only on the contents of the node level attribute UI. If node identifier is not needed, empty string should be assigned to this attribute.

Data type:	String
Value:	String containing a node level identifier with maximum length of 16 characters
Indexing:	No indexing (if used, line attribute will be referred)
Default value:	”NETx”, where x = node number
Access:	Read, write

15.3.4.5 UA User activity

Attribute UA defines whether the user activity events from this node are generated or not. If the UAL event generation is enabled for the node, it may still be disabled by the STA object.

Data type:	Integer
Value:	0 = Disabled 1 = Enabled
Indexing:	No indexing
Default value:	1 (Enabled)
Access:	Read, write

15.3.5 Connected nodes

15.3.5.1 AD Station Address Table

Indicates whether a station address is reserved by a node or not. This attribute gives access to the ACP (application communication protocol) station address table of the communication unit. The table contains the NET object numbers and the corresponding station addresses.

Data type:	Vector
Value:	Vector with 254 integer elements. The value of a vector element is a code specifying the NET system object that occupies the station address given by the index. The code is obtained from the following expression: 4096 + (NET object number) If the station address is free, the value is 0.
Indexing:	The attribute is indexed with a station address or a station address interval, 1 ... 254. The value of the AD attribute tells which node possesses the station addresses given by the index.
Access:	No restrictions

Example:

The expression NET1:SAD10 could for example yield the value 4106, which is 4096 + 10. The ACP station address 10 is reserved by NET10.

15.3.5.2 LI Line Number

The number of the communication line, to which a node (another communication unit or a base system) is connected. If the node is indirectly connected to the communication unit, the attribute is the number of the line to the nearest node. The line number is selected when the connected NET object is created.

Data type:	Integer
Value:	NET line number
Indexing:	The NET object number of the connected NET unit
Access:	Read-only

Example:

```
!SHOW  LINE  NET3:SLI2
```

15.4 Object definition attributes

15.4.1 External nodes

Each communication unit and base system that will be recognized by a communication unit, must be defined as a node, a NOD (NET) communication system object, in the unit. Also all communication units connected in a series and all base systems connected to any of them must be defined as a node. The external nodes are defined by giving them a device number (= object number, node number) and assigning them a line number of an ACP or Integrated Link line.

With SCIL or in the system configuration, the nodes are added by means of the NE attribute described below. After a NOD (NET) object has been defined, it must be given a station address with the SA and SX attributes described in Section 15.3.

15.4.1.1 NE NET

Adding and removing the NET objects of connected nodes in communication units and base systems.

When adding a node, the NE attribute is assigned the value of the connection line, which must have been defined as an ACP or Integrated Link line (see Section 16). The line number for a node that is connected indirectly via other communication units is the line number of the nearest communication unit. PC-NET communication units are connected to the base system on line number 13. When adding a node (that is a new NET object), an error code is produced if the object already exists.

When removing a NET unit, the attribute is assigned a "D". Removing a node means that the NET object, including all its attributes, is deleted.

Data type:	Integer
Value:	Adding an object: Integer, 1 ... 12 (ACP) or 13 (Integrated Link). The line must be a ACP or Integrated Link line.
	Removing an object: Text: "D".
Indexing:	The object number, node number, of the NET object. The number must be unique within the entire SYS600 network.
Access:	No restrictions

Example:

A NET3:S object is added to line 4 of NET2:

```
#SET  NET2:SNE3 = 4
```

The connection to NET3 is removed from NET2:

```
#SET  NET2:SNE3 = "D"
```

15.4.2 Applications

All applications that will communicate with a NET unit, i.e. all applications that will receive any type of messages from NET unit (spontaneous messages, data, system messages, etc.), and all application that will write system configuration data in NET unit must be defined as APL objects in the unit. An application is defined by the node number of the base system, the APLn:B object number in the base system in question and by an APLn:S system object number (device number). Applications are defined by means of the SY attribute described below.

15.4.2.1 SY Application

Adding, removing and redefining an APL object, that is an application, with SCIL. Adding an application means that an APL object is created. Removing an application means that the APL object, including all its attributes, is deleted. A previously created APL can be redefined, for example, so that it refers to an application in another base system.

Data type:	Vector or integer	
Value:	When adding or redefining an application: Vector or two integers (node, appl):	
	'node'	The node number of the base system where the application is found
	'appl'	The application number as known to the base system in question
	When removing an application: 0	
	When read: 10000*node + appl	
Indexing:	The communication system object number of the APL object, 1 ... 250	
Access:	No restrictions	

Example:

The application 5 (APL5:B) in the base system with node number 9 is defined to the NET unit NET1 as APL5:S:

```
#SET NET1:SSY5 = (9,5)
```

15.4.3 Station definition attributes

Each IED or station directly connected to the NET unit must have a corresponding STA object. A STA object is defined by a station type, a system object number (device number) and the NET connection line. Before a STA object can be created, a NET line with the correct protocol must be defined. Most station types allow the connection of several stations to one line. Stations using the ANSI full duplex and the RP570 slave protocols require one NET line per station.

The station types are denoted by a three-letter abbreviation as follows:

STA	Station using the ANSI X3.28 protocol
RTU	S.P.I.D.E.R. RTUs or Collector
PCL	P214 RTUs
REX	REx relays - REF, REL, REC, RED, etc. - connected via LON
LCU	Load Control Unit, LCU500
SPA	SPA modules connected directly to a NET unit or to a LONWORKS line via an LSG device (a LON/SPA gateway)
SPI	Stations connected via the RP570 slave protocol
LMK	LSG devices connected to LONWORKS lines and other LONWORKS devices (for example a Weidmüller node), but not REX type stations.

Some other station types can be defined as exchangeable device types, see Section 15.4.4.

The STA object numbers must be unique for a certain station type, but STA objects of different types can use the same object number. Stations of the types SPA, RTU, PCL and LCU use station number 0 for broadcast station. This station is automatically created by the communication unit.

Instead of a complete device image, the STA (ANSI stations) and PRI type system objects can contain a reference to another system object. The system object has to be of the same type in the same or in another communication unit. See Figure 17. These system objects have no attributes. Each time the NET unit reads the name of a system object, the object number is interpreted according to a translation table. The translation of an object name gives a device image in the same communication unit, or a reference to another system object name of the same type defined in the same or in another unit. A NET unit can know one device image and one reference under the same name (the same object number). If a system object has both a device image and a reference to another object, the reference precedes the device image, and the device image is searched according to the reference. Hence, a switch between the devices is obtained by adding or deleting a reference object.

A new STA object (if it is a real device image) is assigned certain default values, which are given in the STA object descriptions in Section 17.

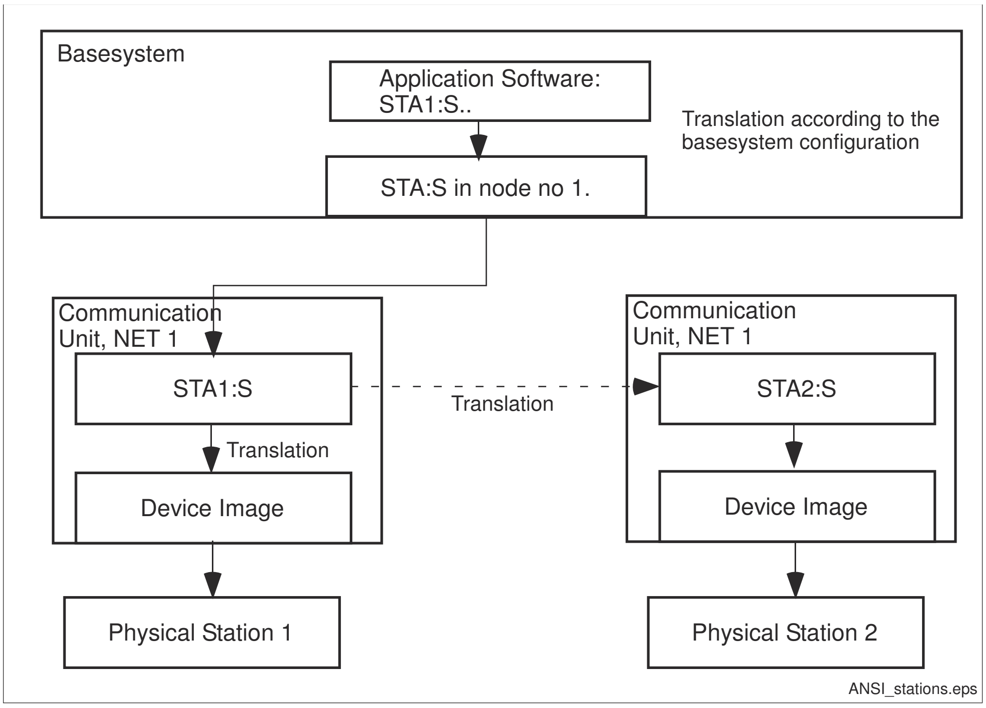


Figure 17: Provided that the stations are of type ANSI, the notation STA1:S in an application program can refer to either station 1 or station 2. A switch from station 1 to station 2 is obtained for example with the statement: #SET NET1:SST1 = (2,2). PRI objects can be used in a corresponding way.

15.4.3.1 LC Load Control Unit

Defining and removing STA objects corresponding to LCU500.

Data type:	Integer
Value:	Adding a device: Integer, 1 ... 12. Line number. The line must have been defined for the LCU500 protocol
	Removing a device: Text, "D"
Indexing:	Integer 0 ... 255. The STA object number. The number must be unique for STA objects of type LCU. 0 = broadcast station (created automatically).
Access:	No restrictions

15.4.3.2 LM LMK Stations

Creating STA objects corresponding to stations on LON: LSG devices and other process devices connected to LON. However REF, RED and REC protective relays are created with the RX attribute instead of LM attribute (see below).

The NET line number to which LMK is connected must be defined as a LONWORKS line.

Data type:	Integer
Value:	Adding an object: Integer, 1 ... 12. Line number. The line must be defined for the LonTalk protocol.
	Removing an object: Text, "D"
Indexing:	Integer, 1 ... 2047. The STA object number of the station. The number must be unique for STA objects of type LMK.
Access:	No restrictions

Example:

Creating a LMK station (the STA attributes are described in Section 17.):

1. Creating a LonTalk protocol line:

```
#SET NET1:SPO1=27    ;LonTalk protocol on NET line 1
#SET NET1:SIU1=1     ;taking the line in use
```

2. Creating a STA object of station type LMK:

```
#SET NET1:SLM1 = 1   ;LONWORKS device nr. 1 on NET line 1
#SET STA1:SAL = 1
#SET STA1:SUT = 2
#SET STA1:SNN = 1    ;LONWORKS node number 1
#SET STA1:SSN = 1    ;subnet number 1
#SET STA1:SIU = 1    ;Starting  the device
```

15.4.3.3 PC Procontrol P214

Defining and removing STA objects corresponding to P214 RTUs.

Data type:	Integer or text
Value:	Adding a device: Integer, 1 ... 12. Line number. The line must have been defined for the P214 protocol.
	Removing a device: Text, "D"
Indexing:	Integer 0 ... 255. The STA object number. The number must be unique for STA objects of type P214. 0 = broadcast station (created automatically).
Access:	No restrictions

15.4.3.4 RT S.P.I.D.E.R. RTU

Defining and removing STA objects corresponding to S.P.I.D.E.R. RTUs or Collector 100 and 300.

Data type:	Integer or text
Value:	Adding a device: Integer, 1 ... 12. Line number. Stations type S.P.I.D.E.R. RTU require the RP570 protocol, and stations of type. Collector the ADLP180 Master protocol.
	Removing a device: Text, "D"
Indexing:	Integer, 1 ... 2047. The STA object number of the station. The number must be unique for STA objects of type S.P.I.D.E.R. RTU. 0 = broadcast station (created automatically).
Access:	No restrictions

15.4.3.5 RX REX Stations

Creating STA objects of station type REX (REC, RED, REF, etc.)

Data type:	Integer or text
Value:	Adding a device: Integer, 1 ... 12. Line number. The line must have been defined for the LonTalk protocol.
	Removing a device: Text, "D"
Indexing:	Integer 1 ... 2047. The STA object number. The number must be unique for STA objects of type REX.
Access:	No restrictions

15.4.3.6 SM S.P.I.D.E.R. SCADA Master Stations

Creating STA objects of station type SPI.

Data type:	Integer or text
Value:	Adding a device: Integer, 1 ... 12. Line number. The line must have been defined for the RP570 slave protocol.
	Removing a device: Text, "D"
Indexing:	Integer 1 ... 255. The STA object number. The number must be unique for STA objects of type SPI.
Access:	No restrictions

15.4.3.7 SP SPACOM

Creating STA objects corresponding to SPA units (SPACOM modules or other stations) using the SPA or LonTalk protocol.

It is possible to configure 2047 SPA devices to one PC-NET. In one PC-NET, a maximum of 12 SPA lines can be created. The amount of SPA devices in one SPA line is limited only by the SPA device addressing but the recommended value 30 or less. Because SPA is a master/slave polled protocol, the number of devices per lines affects directly the performance and response times of the systems. Especially the analog value update times are sensitive to the system configuration, because usually they have to be polled one by one. Therefore, the more devices there are per line, and the more there are configured analog measurements, the slower the response time in the system. In other words, to keep the response times fast, spread the SPA devices to as many lines as possible.

Connecting SPA units to a LONWORKS line requires LSG device, which must be defined as an LMK type station with the LM attribute in the NET unit.

Data type:	Integer or text
Value:	Adding an object: Integer, 1 ... 12. The number of the NET line to which the station is physically connected. If connected to LON, the line is the number of the NET line to which the LSG device is connected. The line must be defined for the SPA or LonTalk protocol.
	Removing an object: Text, "D"
Indexing:	Integer, 0 ... 2047. The STA object number of the station. The number must be unique for STA objects of type SPACOM.
	STA0 = broadcast station that is created automatically.
Access:	No restrictions

15.4.3.8 ST Substation

Defining and removing STA objects corresponding to stations using the ANSI X3.28 protocol. If ANSI half duplex is used, several stations can be situated on the same line.

Data type:	Integer, text or vector
Value:	Adding a device image: Integer, 1 ... 12. Line number. The line must be defined for the ANSI protocol. Removing a device image: Text, "D" Adding a referencce: A vector of two integers of the form (NETnr, device nr) where 'NET nr' is the object number of the NET where the STA object number is translated, and 'device nr' is the translated object number. Removing a reference: 0
Indexing:	Integer, 1 ... 255. The STA object number of the station. The number must be unique for stations using the ANSI x3.28 protocol
Access:	No restrictions

Example:

STA number 5 is added to line 4 of NET1:

```
#SET NET1:SST5 = 4
```

Addition of the system object STA20 in NET1. STA20 refers to STA3 in NET1:

```
#SET NET1:SST20 = 10003
```

Addition of the system object STA4 in NET1. STA4 refers to STA2 in NET2:

```
#SET NET1:SST4 = 20002
```

Removal of the logical name STA20:

```
#SET NET1:SST20 = 0
```

15.4.4 Adding devices of exchangeable device types

Internally in a NET unit, each object type is represented by an integer code, 1 ... 31. The code numbers 1 ... 30 are predefined, including all the station types enumerated in Section 15.4.3. The codes 24 ... 30 are predefined to following station types and the creation of these STA objects must be done using the attribute DV.

24 = ADE (General ASCII/ADEMCO)

25 = PCO (Procontic/RCOM)

26 = WES (Westinghouse F4F)

27 = ATR (Alpha meter/IEC 61107)

28 = PLC (Modbus)

29 = IEC (IEC 60870-5)

30 = DNP (DNP3)

15.4.4.1 DV Device

Adding and removing devices of exchangeable device types. In PC-NET, this attribute can also be used to create stations of fixed type numbers if the protocol is supported. The maximum value of the object number is also the maximum amount of STA objects of that type in one PC-NET instance. The given object number and value of the TN (Translated Object Number) attribute of the base system object number binds the basesystem STA object and STA object of the NET unit to each other.

Data type:	Vector	
Value:	Write:	Vector of two elements:
	Element 1:	Communication system object number. Limits: For device types SPA, RTU, PLC, REX, LMK, IEC and DNP : 1 .. 2047. For other station types : 1 .. 255
	Element 2:	Line number, when adding a device "D", when removing a device
	Read:	The defined device numbers of the given device type. 0 = undefined
Indexing:	Write:	One index, the device type code 24 ... 30 or fixed device type codes 3 ... 23
	Read:	1000 * type code + object no
Access:	No restrictions	

Example:

A station of type number 29 (IEC) is added on line 3 as STA2:

```
#SET NET1:SDV(29) = (2,3)
```

The station added above is removed:

```
#SET NET1:SDV(29) = (2,"D")
```

The variable A will get the value 0 if no such object is defined, otherwise the value 5:

```
@A = NET1:SDV29005
```

15.4.5 Printer definition attributes

Printers are defined to the NET unit by a PRI object with an object number (device number) and a connection line number. Printers can be defined with SCIL or with tools. With SCIL, printers are defined when new PRI objects are created using the PR attribute described in this section.

Like the STA objects of type ANSI stations, a PRI object can be either a real device image or a reference to another printer connected to the same or another communication unit, see Figure 17. One device image and one reference can be defined with the same object number.

When a new PRI object is created (which is a device image), it gets certain default values. The default values are listed in appendix A.

This attribute is needed only when the printer is connected to the serial line of a NET unit.

15.4.5.1 PR Printer

Adding and removing a printer with SCIL, that is a PRI object, or a reference to another PRI object.

Data type:	Integer, text or vector
Value:	Adding a device image: Integer, 1 ... 12. Line number.The line must be defined as an ASCII line.
	Removing a device image: Text, "D"

Table continues on next page

	Adding a reference: A vector of two integers of the form (NETnr, device nr) where 'NET nr' is the object number of the NETwhere the PRI object number is translated, and 'device nr' is the translated object number
	Removing a reference: 0
Indexing:	The object number (1 ... 8) of the new object as known by the NET in question.
Access:	No restrictions

Example:

Printer number 2 is removed:

```
#SET NET1:SPR2 = "D"
```

15.5 Application attributes

The attributes in this sub-section apply to the application objects defined in NET unit. The SU and SW attributes specify the communication between the application and the communication unit. The DA and DS attributes provide means for monitoring the communication between NET unit and the application.

The attributes are accessed in SCIL by the notation:

NODn:Satm (or NETn:Satm)

where

‘n’	Is the node number of the NET unit where the application is defined. The ‘at’ is the attribute name, and ‘m’ is the object number of the application.

15.5.1 DA Diagnostic Counters for APL Connections

Each application defined in NET unit has 4 diagnostic counters accessed with the notation:

NETx:SDAs

where

‘s’ Is the APL number in NETx. The counters are:

1. PL_SUSPENSIONS
2. APL_QUERIES
3. APL_TIMEOUTS
4. APL_ERROR_REPLIES

Data type:	Vector
Value:	Vector of four integers, the values of the diagnostic counters
Indexing:	Application object number in NET unit
Access:	Read-only, values can be reset

Example:

The variable A is a vector of four integers, the diagnostic counters of application 2:

```
@A = NET1:SDA2
```


15.5.2 DS Diagnostic Status

The current status of the connection between the applications and the NET unit.

Data type:	Vector	
Value:	Vector of integer values, 0, 1 or 255:	
	0	Suspended
	1	OK
	255	Undefined
Indexing:	Application object number in NET unit (1 ... 250)	
Access:	Read-only	

Example:

A is a vector with the status of all applications known to NET1:

```
@A = NET1:SDS(1 .. 250)
```

15.5.3 SU Application Suspension Time

The time in seconds between diagnostic commands issued to a suspended application.

The communication sends cyclically diagnostic status requests to all known applications. As long as the APL connection is OK, NET unit sends diagnostic commands with an interval of double the SU attribute (2*SU) if there is no other communication with the application during this time.

NET unit suspends an APL connection when it receives no reply or a reply with a non-zero status code to a diagnostic command, a split message or a message from an RTU. While an application is suspended, diagnostic commands, and exclusively diagnostic commands, are sent to it until it replies. When the application replies to a diagnostic command, the suspension is cancelled and the APL will again receive spontaneous messages from the stations.

Some types of stations cause an application to be suspended if there is no base system STA object mapped for a station that sends a message to the application. Also, if the station type in question is not the default station type, the application can be suspended.

When an application is suspended, NET unit sends a system message (see Section 15.3.4) primarily to the message application (the MS attribute). If this application is suspended, NET unit sends the message to the first connection in number order that has an OK connection. When the application recovers, NET unit sends one system message to the application itself and one the application that received the suspension message. The system messages are addresses to MI(2), see the MI attribute in Section 15.3.4.

Data type:	Integer	
Value:	0 ... 300	
	0	No diagnostics
	1 ... 300	Number of seconds
	If set > 300, the value of the attribute will be 300	
Unit:	Seconds	
Default value:	60 seconds	

Table continues on next page

Indexing:	Application object number in NET unit
Suggested value:	30 ... 120
Access:	No restrictions

Example:

The suspension time for APL5 is set to 2 seconds:

```
#SET NET1:SSU5 = 2
```

15.5.4 SW Application Reply Wait Time

The reply wait time in seconds for diagnostic commands and split substation messages (see Section 17) transmitted to an application. If no reply is received until this time has run out, the application connection will become/remain suspended. The RTUs with a suspended application as the Allocating Application (see the STA attribute AS) are not polled.

Data type:	Integer
Value:	Integer
Unit:	Seconds
Default value:	5
Indexing:	Application object number in NET unit
Suggested value:	5
Access:	No restrictions

15.6 Authentication attributes

15.6.1 KS Key Storage File

The KS attribute is used to define the key storage file which contains the user information and authentication keys needed by the station objects created to this NET node. If none of the station objects uses authentication, the definition of KS is not needed and the value may be an empty string. All station objects configured to same NET node use the same key storage file. The creation of the key storage file is done using the Authority Tool. If an empty string is written to the KS attribute, the node is detached from the key storage file. This is needed is HSB configuration. For more information, see the System Configuration manual.

Data type:	Text
Value:	The directory and name of the key storage file, up to 128 characters, or empty string when key storage is detached (HSB setup)
Default value:	Empty string
Access:	No restrictions

Example:

Files "SS_HillSt_NCC.dat" and "SS_HillSt_IEDs.dat" have been created for the COM500i computer.

```
#SET NET1:SKS="C:\sc\sys\active\sys_\SS_HillSt_NCC.dat" ; database for
NCC lines #SET NET2:SKS="C:\sc\sys\active\sys_\SS_HillSt_IEDs.dat" ;
database for process devices
```

15.7 Miscellaneous NET device attributes

15.7.1 FM Free Memory

The amount of free memory in the NET unit.

Data type:	Integer
Value:	Integer
Unit:	Bytes
Access:	Read-only

15.7.2 LP Local Port

The LAN protocols created to NET nodes will reserve IP port numbers for their interprocess communication. In case there are multiple NET nodes using the same local IP address and same line numbers, there is a limitation of creating these lines. This limitation is described in the protocol specific manuals in the description of the line attribute LD. In large systems, where it necessary to create multiple NET nodes and tens of LAN protocol lines using the same local IP address, it may necessary to override this limitation by setting a new base value for the reserved port number range using the NET node attribute LP, local port. In practice, this means that a unique value of LP for each NET node instance should be given.

The rules for setting LP:

basevalue:	This could be the default 2502 or an unassigned value from the TCP/UDP port number list maintained by IANA (Internet Assigned Numbers Authority).
offset:	This should be 12 or more (256 or more if IEC60870-5-104 redundancy is used).

The assigned values should be:

NET'a':SLP = basevalue

NET'b':SLP = basevalue + offset

NET'c':SLP = basevalue + offset*2

The value of LP should be set before creating the NET lines using LAN protocols.

If the maximum amount of the communication lines (12) is created to the NET node, the reserved port numbers are LP ... LP + 11 (or LP ... LP + 255 if IEC60870-5-104 redundancy is used). Each created line reserves one port from the mentioned range (16 if IEC60870-5-104 redundancy is used). The port numbers are reserved from the IP addresses defined by the line attribute LD.

Data type:	Integer
Value:	1024 ... 65280
Default value:	2502
Access:	No restrictions (see setting rules above)

Example:

```
#set NET1:SLP=2514  ;ports 2514 .. 2525 reserved
```

SYS600 versions 9.2 and older have used a fixed basevalue of 2502 in each NET node. The default value of LP will provide the same characteristics as the previous versions.

The following port numbers are reserved by the listed protocols for external communication. The port number range defined by LP should not overlap these values:
2404 IEC60870-5-104 slave, 20000 DNP3.0 LAN/WAN slave, 20000 DNP3.0 LAN/WAN master (in UDP mode only), 21845 and 21846 SYS600 LAN Link
All master protocols using TCP/IP (IEC60870-5-104 master, DNP3.0 TCP master, Modbus TCP, SPA-TCP) are operating as TCP clients. Consequently, no protocol specific port numbers are reserved.

15.7.3 LT Last Transaction Number

NET unit has a buffer for storing the last data messages received from S.P.I.D.E.R. RTUs and SPACOM units. Using the LT attribute, the last transmitted transaction number can be read, and a forced re-transmission to the application of the latest transactions can be started.

Data type:	Integer
Value:	When read: Integer. The transaction number of the last data message which NET unit has forwarded to the application (the application from where the read command is issued).
	When written: Integer. A transaction number. All stored transactions above this number (if any) are transmitted to the application.
Access:	No restrictions

Example:

The transactions occurred after the last received transaction is transmitted to the application that issues the command:

```
#SET NET1:SLT = NOD1:BLT
```

15.7.4 MA Mailbox

The content of a buffer, which can be freely used for transferring messages between NET unit and the base system. When NET unit is started, all elements in the MA attribute are set to zero. After that, the attribute can only be changed with SCIL. The attribute can, for example, be used for checking that NET unit has started (provided that the attribute has been previously set to another value than 0).

Data type:	Vector
Value:	A vector of 250 real or integer elements
Indexing:	1 ... 250 real elements, 1001 ... 1250 integer elements
Access:	No restrictions

15.7.5 MU Mail Update Identification

When this attribute is written, a system message (16633) is sent to the application specified by the MS attribute.

Data type:	Vector
Value:	Vector of two integers
Access:	No restrictions

15.7.6 TM Time

The time of the NET unit in SYS600 time format (32 bits integer). Using this attribute, the computer clock can be set manually. The attribute is supported because of the backward compatibility.

Data type:	Integer, vector	
Value:	Integer or vector of three integer elements	
Indexing:	No index or index 1:	SYS600 time data given as an integer (with a resolution of one second). Index 1 cannot be given time values older than 80-01-01 (63072000).
	Index 2:	The milliseconds of the SYS600 time
	Index 3:	Increments the current computer with the given number of milliseconds
	Index 3 is used for adjusting the computer time. Index 3 can be given negative values, which means a decrease of the computer time with the given number of milliseconds.	
Access:	Index 1,2:	No restrictions. Reading and writing to index 2 must be done together with index 1.
	Index 3:	Write only

Example:

The computer time is is moved forward with 10 seconds.

```
#SET NET1:STM = CLOCK + 10
```

The computer time is set and incremented with 50 milliseconds.

```
    @A = HR_CLOCK
    #SET NET1:STM(1..2) = (A:VCL,(A:VUS DIV 1000))
    #SET NET1:STM3 = 50
```

15.7.7 TZ Time Zone Correction

The value of the TZ attribute is added to the synchronization time read from an external clock or set from a central station connected to a RP570 slave, IEC 101 or DNP 3.0 slave line. The time zone can be changed on-line from the base system with this attribute.

Data type:	Integer
Value:	-32767... 32767. Time zone correction in minutes.
Default value:	0
Unit:	Minutes
Access:	No restrictions

Example:

```
#SET NET20:STZ=60
```

Section 16 NET lines for communication system

This section describes the configuration of NET lines and the NET line attributes. The section is divided in the following sections:

Section 16.1	General: The numbering and use of NET lines, the definition of NET lines and the object notation for accessing NET lines. An overview of the NET line attributes applicable for different protocols.
Section 16.2	Basic line attributes: PO, SD, LD, IU, LK, PM, PS, NB, PB, MI, MS, AO, AU.
Section 16.3	Data transmission attributes: BR, ER, PY, RD, RE, SB, TD.
Section 16.4	Communication control attributes: DE, EN, HT, NA, OS, PD, RI, RK, RL, SG, TI, TD, OM.
Section 16.5	Polling attributes: AW, CP, PD, PL, PP, PT, RP.
Section 16.6	Autodialing attributes: AC, AS, CL, CN, CS, CT, DD, MC, PU, RC, RW, SR.
Section 16.7	LON configuration attributes: NC, XA.
Section 16.8	Miscellaneous NET line attributes: Diagnostic counter (DC) and clock synchronization attributes (LK, SF, SS).

The attributes in this description cover mainly the protocols listed in Table 3. The attributes of other protocols are described in separate documents, which are available on request.

16.1 General

16.1.1 NET communication lines

NET communication lines are created to an instance of PC-NET communication unit.

The PC-NETs communicate through the serial ports (COM ports) of the host computer, possible PCLTA cards (one or two) or through the network interfaces. The COM ports can physically be in the mainboard of the computer or in the extension card. It is also possible to use virtual serial ports by using simulated states of the CTS and DCD signals, see description of the line attribute CM (COM Port Mode).

In case the protocol of the NET line uses COM port, the port name is assigned using the line attribute SD (System Device Name).

The protocols using TCP/IP or UDP/IP are configured to use some of the network interfaces present in the system. The IP address which is used in the communication is defined with the line attribute LD (Local Address). The used port number is selected by the operating system (Protocol operates as a TCP client) or is fixed for the protocol (Protocol operates as a TCP server).

The NET line numbers of the LONWORKS channels (up to two per card) can be freely chosen among the free NET line numbers 1 .. 12. On the LONWORKS lines, only the LonTalk protocol is supported. The connection between the NET line numbers and the LONWORKS channels are also defined by a NET line attribute SD.

PC-NET communicates with the base system (kernel) through line number 13, which is a fast internal communication link used only for the basesystem communication.

16.1.2 Autodialing

Autodialing can be used on all NET serial lines defined for the following protocols:

- ANSI X3.28 Half Duplex or Full Duplex
- ACP (MicroPROTOCOL)
- Modbus
- IEC 61107
- RP570 master protocol
- SPA
- Alpha
- IEC 60870-5-101 master and slave
- IEC 60870-5-103 master
- DNP3.0 master and slave

16.1.3 Network protocols

Protocols which use the LAN connection are IEC60870-5-104 master and slave, DNP V3.00/LAN master and slave, Modbus TCP and SPA-TCP. Protocols other than SPA-TCP are also described in the protocol specific manuals.

For protocols DNP V3.00/LAN master and slave, Modbus TCP and SPA-TCP, the operation mode which uses network is defined using the line attribute SD. In general, the IP address of the remote end is defined with the IA station attribute and the IP address which locally used by the line is defined with the line attribute LD.

16.1.4 Line definition

Using a NET line requires that it is defined in the NET in question. A NET unit line is basically defined by assigning it a protocol. It is recommeded to use System Configuration Tool for line creation, but if configuration is created using SCIL scripts, the line creation is done setting the attribute PO as described in Section 16.2. See SYS600 System Configuration for more information.

The line is further specified by a number of line attributes. Depending on the protocol used on a line, some of the line attributes are applicable some are not. Table 3 lists the line protocols and the applicable line attributes.

When a line is defined online, separate attribute writes is needed to complete the configuration of the created line. In most cases, the line attributes can be changed only while the line is out of use (the In Use attribute, IU, is 0).

16.1.5 Object notation

The NET line attributes are accessed from SCIL with the notation:

NETn:Sati (or NODn:Sati)

where

'n'	NET object number (node number)
'at'	The attribute name
'i'	An index (normally the line number)

If not otherwise mentioned in the attribute descriptions, all line attributes are indexed with the NET line number (1...12 and 13).

Table 3: An overview of the NET line attributes

Protocol Attributes	ACP and ANSI full duplex	Integrated link	SPA	LONWORKS	RP570/571 Master	RP570 Slave
Basic attributes -Basic definition -In Use -System Message -Other	- PO IU MS, MI LK, PS	- PO IU - PS	- PO IU MS, MI LK, PS	- PO IU MS, MI PS, SD	- PO IU MS, MI PM, PS	- PO IU MS, MI PS
Data transmission	BR, ER, RD RE, SB, TD		BR, PY, RD SB, TD		BR, PY, RD RE, SB, TD	BR, PY SB, TD
Communication Control	CM, EN, NA, SG, TI	EN, NA, TI	CM, DE, HT, RL	HT, PD, TI	CM, DE, EN, HT RI, RK, SG TI, TW	CM, DE
Polling			PD, PP, PR PT		AW, CP, PD PP, RP	
Autodialing	AC, AS, CL CN, CS, CT DD, MC, PU RC, RW, SR		AC, AS, CL CN, CS, CT DD, MC, PU RC, RW, SR		AC, AS, CL CN, CS, CT DD, MC, PU RC, RW, SR	
Communication Loops					BO, BU, CF DR, LU, LS MD, MT	
LONWORKS Conf.				NC, XA		
Miscellaneous -Diagnostics -Clock Synchr.	- DC	- DC	- DC	- DC LK	- DC	- DC

Table 4: An overview of the NET line attributes.

Protocol Attributes	ANSI half duplex	ASCII printer	General ASCII	Modbus	IEC 60870-5 (-101 and -103)		
Basic attributes -Basic definition -In Use -System Message -Other	- PO IU MS, MI LK, PS	- PO IU MS, MI PS	- PO IU MS, MI PM, PS	- PO IU MS, MI PS	- PO IU MS, MI LK, PS		
Data transmission	BR, ER, RD RE, SB, TD	BR, PY, RD SB, TD	BR, PY, RD SB, TD	BR, PY, RD SB, TD	BR, PY, RD SB, TD		
Communication Control	CM, DE, EN, HT, RK, SG, TI TW, OM	CM, OS, SG, TI	CM, SG, TI	CM, DE, EN, HT, SB, RI, TI	CM, DE, TW, RK RI, EN		
Polling	AW, CP, PD PP, RP			PD	PD, PP, PL RP		
Autodialing	AC, AS, CL CN, CS, CT DD, MC, PU RC, RW, SR			AC, AS, CL CN, CS, CT DD, MC, PU RC, RW, SR	AC, AS, CL CN, CS, CT DD, MC, PU RC, RW, SR		

Table continues on next page

Protocol Attributes	ANSI half duplex	ASCII printer	General ASCII	Modbus	IEC 60870-5 (-101 and -103)		
Communication Loops							
LONWORKS Conf.							
Miscellaneous -Diagnostics -Clock Synchr.	- DC	- DC	- DC SF, SS	- DC	- DC		

16.2 Basic line attributes

16.2.1 Basic definition

The PO attribute creates and removes NET line definitions. A line that has not been defined in NET cannot be used.

Using the SD attribute, the connection between the NET line number and the physical line can be changed on LONWORKS lines.

16.2.1.1 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 line attributes, are deleted.

When the line is defined, its line attributes get the protocol dependent default values given in the attribute descriptions.

The PC-NET communication units support the following protocols (as well as some other protocols):

- ACP, Application Communication Protocol. This protocol is used for the communication between SYS600 nodes (base systems and communication units). It is a protocol for point-to-point lines, where both ends transmit spontaneously. The ACP protocol is based on ANSI X3.28 Full Duplex but additional features have been added to the upper protocol layers.
- ANSI X3.28 Full Duplex. This protocol is used for the communication with stations of type Allen-Bradley, SPACOM via SRIO, Westronic D20 and M4000, DTU1 and 2, SELMA II and SCP-micro. It is a protocol for point-to-point lines, where both ends transmit spontaneously.
- ANSI X3.28 Half Duplex. This protocol is used for RTU communication on multidrop lines. The stations are usually polled cyclically. The protocol is used by the same stations as for ANSI X3.28 full duplex.
- ASCII printer protocol. This protocol is used for printer communication.
- RP570/571 master protocol. The protocol is used for communication with S.P.I.D.E.R. RTUs. NET is the master and the connected RTUs are slaves.
- RP570 slave protocol. When using this protocol on a NET line, NET is regarded as the slave and the communicating device as the master. The master sees NET as a S.P.I.D.E.R. RTU200.
- SPA protocol for direct communication with SPACOM modules. The protocol is allowed on max. four lines per NET.
- The P214 protocol (Indactic 35). The protocol is used for communication with P214 RTUs (Indactic 35).
- ADLP180 Master for communication with Collector 100 and 200.

- LCU500 for communication with LCU500 stations.
- General ASCII for communication with clock synchronization receivers, or ADEMCO alarm receiver.
- RCOM (Procontic) for communication with Procontic PLCs.
- MODBUS RTU mode master for communication with PLCs, process automation systems, etc.
- ABB Alpha for communication with Alpha meters.
- IEC 61107 for metering device communication.
- IEC 60870-5 for communication with external device which uses the IEC 60870-5-103 master or IEC 60870-5-101 or IEC 60870-5-104, master or slave.
- DNP V3.00 slave and master protocols.
- Integrated link for communication with base systems in the same PC. This protocol can be used only on line 13.
- LonTalk protocol for communication with process units connected to the LONWORKS network. Such process units are the REx protective relays, SPA relays via LSG device, and Weidmüller process control devices.

Writing something else than 0 to the attribute is possible only if the line is undefined. Changing protocol on a line requires that the line definition first is deleted (PO = 0). Reading the PO attribute for undefined lines returns the value 0.

Data type:	Integer	
Value:	0 ... 43	
	0	None. The line is not defined
	1	ANSI X3.28 Full Duplex or ACP
	2	ANSI X3.28 Half Duplex (HF)
	4	ASCII protocol for printer (or printer simulating device) (AS)
	7	RP570 protocol (SR)
	9	P214 protocol (PR)
	12	LCU500 (LC)
	13	ADLP180 Master (AM)
	14	SPA protocol (SP)
	15	General ASCII (AG)
	16	RP570 Slave (RP)
	17	RCOM (Procontic) (PC)
	18	Westinghouse F4F
	23	ABB Alpha meter protocol
	24	General PLC protocol
	25	Modbus RTU master protocol
	26	IEC 61107 protocol
	27	the LonTalk protocol
	28	Integrated link protocol (used on line 13 of PC-NETs)
	29	IEC 60870-5-101 Unbalanced slave
	30	IEC 60870-5-101 Balanced slave
	31	IEC 60870-5-101 Unbalanced master
	32	IEC 60870-5-101 Balanced master
	33	IEC 60870-5-103 Unbalanced master
	35	DNP V3.00 slave protocol
	43	DNP V3.00 master protocol

Table continues on next page

	44	IEC 60870-5-104 master
	45	IEC 60870-5-104 slave
Indexing:	Line number	
Access:	Read, conditional write	

Example:

```
#SET NET3:SPO1 = 14
```

Defining line 1 of NET number 3 for the SPA protocol and taking the line into use:

```
#SET NET3:SIU1 = 1
```

Taking the line out of use and removing the line:

```
#SET NET3:IU1 = 0
    #SET NET3:PO1 = 0
```

Deleting a line definition is possible only if there are no devices connected to the line.

16.2.1.2 SD System Device Name

Associates the NET line numbers of the PC-NET with the device names of the physical communication interfaces.

On LONWORKS lines:

Each physical connection from PCLTA card or from network connected Loytec device (a LONWORKS channel) is associated with a specific device name (see the Installation Manual, Section 7).

If network connected Loytec device is used, see a separate Application Note for the definition of the SD value.

If the LONWORKS device driver MiSCLONP is installed and configured according to the recommendations in the Installation manual, "LONP0" is the device name of channel A on the "first" PCLTA card, "LONP1" is the device name of channel B, and so on.

When a NET line is defined and assigned the LonTalk protocol (PO = 27), it is related to a device name with "LONP" as the first four letters and a number calculated as the NET line number minus 1 as the last digit. For example, if NET line 2 is defined as a LonTalk protocol line (PO = 27), it is by default assigned the device name "LONP1". The name corresponds to channel B of PCLTA card 1 (if the LONWORKS device driver is installed as recommended in the Installation manual).

It is possible to override these default values by using the SD attribute.

If the Echelon device driver is used, the device name for channel A is "LON1" and for channel B "LON2". The channels of the second PCLTA card are named "LON3" and "LON4".

On lines that use serial ports:

SD attribute gives the possibility to associate the NET line number to any COM port number.

Data type:	Text
Values:	The device name of the communication interface
Indexing:	NET line number
Access:	Read, conditional write (IU must be 0)

On lines that use LAN:

The SD attribute is used to define the mode of the operation. When a value "TCP" or "UDP" is given, the line uses LAN instead of serial port.

Data type:	Text
Values:	"TCP" for SPA, Modbus Master and DNP 3.0 master and slave protocols "UDP" for DNP 3.0 master and slave protocols
Indexing:	NET line number
Access:	Read, conditional write. (This attribute should be defined before the line is taken into use for the first time)

Example:

To associate line 1 of the NET 3 to COM port 9:

```
#SET NET3:SSD1="COM9"
```

The SCIL statement connects NET line number 2 of NET1 with the LONWORKS device name LONP0 (channel A of PCLTA card 1 if installed according to the advises in the Installation manual):

```
#SET NET1:SSD2 = "LONP0"
```

To configure the created SPA line to operate in TCP mode:

```
#SET NET1:SSD2 = "TCP"
```

16.2.1.3 LD Local Address

The IP address which is locally used. The setting of this attribute is necessary when the computer has multiple IP addresses and it is defined which address the line must use. The setting of this attribute must be done before the line is taken into use for the first time. This attribute is supported in the IEC60870-5-104 protocols, DNP-LAN protocols, Modbus TCP protocol and SPA protocol in TCP. For more information on protocols other than SPA-TCP, read also the protocol specific manual. This attribute is not available with serial protocols and the LonTalk protocol.

Data type:	Text
Values:	String containing a valid IP address, max 29 characters
Access:	Read, conditional write

This attribute accepts the IP address in form

```
#SET NET1:SLD1 = "10.0.0.10"
```

16.2.2 Other basic attributes

16.2.2.1 IU In Use

The state of the line, whether it is in use or not. When a line is not in use, no data can be transmitted on it, and no data is received from it. The line attributes can be read as usual. Generally, a line must be taken out of use by setting this attribute to 0 before the line attributes can be written.

When a line is stopped by setting the IU attribute = 0, all data transmission on the line ceases. However, before that, NET executes to the end all on-going data transactions. E.g., the polling of the station in turn is completed.

If the line is a LonTalk protocol line, the station objects should be taken into use after the line object of the stations is set to IU=1. Correspondingly, all station objects connected to a LonTalk protocol line should be taken out of use before the line object is set to IU=0.

Data type:	Integer	
Value:	0	Not in use, the line communication is stopped
	1	In use
Indexing:	Line number	
Default value:	0	
Access:	No restrictions	

16.2.2.2 LK Link Type

The type of data link connection used on the line. The attribute has no meaning for the printer lines, nor for LONWORKS lines. (LonTalk protocol lines do have a LK attribute but this has another meaning.)

In protocols using serial ports, the states of the CTS and DCD signals may have simulated values and the usage of the special serial cables are not needed. See line attribute CM for more information.

Data type:	Integer	
Value:	0	Direct RS232C connection (used for direct lines, modem lines and Data Highway)
	1	Modem line. A normal RTS CTS modem signal handshaking is applied
	2	VY85 (a special type of radio telephone connection (mobile telephone link)). Possible only on ANSI X3.28 half duplex lines.
	3	Radio telephone. Possible only on ANSI X3.28 half duplex lines.
	4	Radio link. Possible only on ANSI X3.28 half duplex lines.
	6	Data Highway
	7	PAC-5 lines
	8	A special radio link possible on ANSI X3.28 half duplex lines. NET checks DCD before trying to send anything. If DCD is active, NET waits for passive DCD for the time specified by the TI attribute, the number of times specified by the EN attribute. Transmission starts when DCD becomes passive. If DCD does not become passive within the time EN*TI, the transmission is regarded as failed and the station is suspended.
	9	Modem signals disabled. CTS is not used, and the transmission starts after a finite delay given by the DE attribute.

Table continues on next page

	10	RTS is set and transmission starts immediately. No other modem signals are issued. NET does not wait for CTS signal. Used on optical lines.
	14	Full duplex. No collision avoidance, Data Carrier Detect signal is handled as in other protocols.
	15	Half duplex. Collision avoidance on, transmission when the Data Carrier Detect signal of the line is not set.
Indexing:	Line number	
Default value:	0	
	15 for DNP V3.00 protocol	
Access:	Read, conditional write	

16.2.2.3 PM Protocol Mode

The mode of the protocol. The attribute applies to the general ASCII protocol (PO = 15), RP570/571 master protocol (PO = 7), the RCOM protocol (PO = 17), SPA protocol (PO = 14), LCO500 protocol (PO = 12) and the Modbus master protocol (PO = 25).

Data type:	Integer	
Value:	General ASCII:	
	4	Receive only. This mode is used for ADEMCO alarm receiver.
	5	Time Synchronization receive only. This mode is used for all clock synchronization receivers, except TAIP.
	6	Time synchronization receive/transmit with pulse synchronization. When the NET line is take into use (IU set to 1), NET sends configuration messages to the clock device and continues until it receives a synchronization message from the clock. Carrier detect signal (CD) is used for sync pulse receiving. PM = 6 must be used if the sync format is TAIP (SF attribute = 6). It cannot be used on lines using other sync formats.
	RP570/571 master and slave:	
	0	RP570
	1	RP571
	RCOM:	
	0	Event polling disabled
	1	Event polling enabled
	Modbus master:	
	0	Modbus RTU mode
	1	Modbus ASCII mode
	SPA/SPA-TCP:	
	0	Time synchronization enabled
	1	Time synchronization disabled
	LCU500:	
	0	Low priority pulse counter polling
	1	High priority pulsecounter polling
Indexing:	NET line number	
Default value:	General ASCII: 5	
	RP570 and RCOM: 0	
	Modbus master: 0	
	SPA (serial): 0	

Table continues on next page

	SPA-TCP: 1
	LCU500: 0
Access:	Read, conditional write

16.2.2.4 PS Buffer Pool Size

The number of message buffers reserved for the line. Each buffer can contain one message. The maximum data content length of a message depends on used protocol. In general, the temporary buffer consumption is higher with master protocols than with slave protocols. The allocated buffers are divided into two pools and the current state of these pools can be checked using line attributes NB (Normal Buffer Pool) and PB (Priority Buffer Pool).

In SYS600 version 9.3 FP1 and newer, fixed buffer pool size values are used. This attribute is supported for backward compatibility, but if written, the written value is ignored.

Data type:	Integer
Value:	2000 for IEC60870-5-104 master protocol and DNP3.0 master protocol 250 for other protocols
Indexing:	NET line number
Access:	Read, conditional write (write ignored)

16.2.2.5 NB Normal Buffer Pool

The usage of the normal buffer pool can be checked using the line attribute NB. This attribute is available with all protocols. The size of the normal buffer pool is half of the size defined by line attribute PS.

The number of available buffers in the pool is read using the following description. This is the recommended method to monitor the pool usage e.g. in the test dialog.

Data type:	Integer
Value:	The number of available buffers
Indexing:	NET line number
Access:	Read only

If the value of the attribute is 0, the communication line has most probably entered an abnormal state.

See Example 1 below.

For more detailed information, the availability of individual buffers can be checked using the following description. This method is used by various tools.

Data type:	Integer vector	
Value:	10×line number + 1 255	Buffer with defined index available Buffer not available
Indexing:	100*line number + 'first' .. 100*line number + 'last' minimum of 'first' is 1 maximum of 'last' is 99	
Access:	Read only	

Example 1:

```
@NB = NET1:SNB2;           the amount of available buffers in the normal
buffer pool of line 2 in NET1 is stored in variable NB
```

Example 2:

```
@NB = NET1:SNB(201..299);           the availability of buffers 201..299
in the normal buffer pool of line 2 in NET1 is stored in variable NB
```

16.2.2.6 PB Priority Buffer Pool

The usage of the priority buffer pool can be checked using the line attribute PB. This attribute is available with all protocols. The size of the priority buffer pool is half of the size defined by line attribute PS.

The number of available buffers in the pool is read using the following description. This is the recommended method to monitor the pool usage e.g. in the test dialog.

Data type:	Integer
Value:	The number of available buffers
Indexing:	NET line number
Access:	Read only

If the value of the attribute is 0, the communication line has most probably entered an abnormal state.

See Example 1 below.

For more detailed information, the availability of individual buffers can be checked using the following description. This method is used by various tools.

Data type:	Integer vector	
Value:	10×line number + 1 255	Buffer with defined index available Buffer not available
Indexing:	100*line number + 'first' .. 100*line number + 'last' minimum of 'first' is 1 maximum of 'last' is 99	
Access:	Read only	

Example 1:

```
@PB = NET1:SPB2    ; the amount of available buffers in the priority
buffer pool of line 2 in NET1 is stored in variable PB
```

Example 2:

```
@PB = NET1:SPB(201..299)    ; the availability of buffers 201..299 in the
priority buffer pool of line 2 in NET1 is stored in variable PB
```

16.2.3 System message handling

The attributes in this subsection apply to all lines and all protocols, except printer lines. The NET lines generate system messages, for example, in the following situations:

- When the line is taken into use (IU = 1) (does not concern all protocols).
- At dial-up (concerns autodialing lines).
- When no time synchronization message is received from a General ASCII line.

Refer to Section 15 and the System Configuration manual to learn how to handle generated system messages.

16.2.3.1 MI Message Identification

Object address of system messages. If the NET node attribute SE (System messages enabled) has a value 4, a binary process object is updated at the same time as the analog process object defined with MI.

See the MI attribute in Section 15 for more information.

Data type:	Integer
Value:	1 ... 32760
Indexing:	NET line number
Default value:	When a line is defined on-line, the MI gets a default value obtained from the expression:
	6000 + (100 * NET number) + line number
	This default value can be used as such (copied to the process object address), or it can be changed.
Access:	Read, conditional write

16.2.3.2 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. The APL object number as known to the communication unit.
Indexing:	NET line number
Default value:	1
Access:	Read, conditional write

16.2.3.3 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:	NET line number
Default value:	".LINEx", where x = line number
Access:	Read, conditional write

16.2.4 Protocol analyzer interface

The attributes in this subsection apply to all lines and protocols.

The AO attribute is used to define and enable/disable the interface for the analyzer. When an analyzer is enabled, it can be used to view the transmitted and received messages. The analyzer module prints out the bytes as they are passed to or returned from the operating system calls which control the communication hardware. The timestamp in the output is taken from the local clock of the computer. It defines the time point of the Read/Write system call made.

The AU attribute is used to format the analyzer output.

16.2.4.1 AO Analyzer Output

The IP address which is locally used as an interface to the protocol analyzer.

The setting of this attribute is necessary when there is a need to analyze the incoming and outgoing messages of a communication line. This attribute is supported with all protocols. The port number is optional and if used, it must be separated with a colon.

When an empty string is written to the attribute, the analyzer interface is removed from the system. When the attribute is read and an empty string is returned, no interface for this line exists. The analyzer interface operates as a TCP server and any TELNET program can be used as a user interface. For problem solving it is necessary to have a possibility to record the analyzer output to a file.

Data type:	Text
Value:	String containing the IP address "127.0.0.1". The optional port number is separated with a colon, see examples below.
Indexing:	NET line number
Default value:	Empty string
Access:	Read, write

If the port number is not specified, port number 50 000 + line number is used. Only the local IP address 127.0.0.1 is supported at the moment.

The analyzer interface reads the bytes sent by the client program, but does not use them. In case the total amount of the received bytes is bigger than 1024, the TCP session is closed by the analyzer interface.

Examples:

```
#SET NET1:SAO1="127.0.0.1:10001"

; A protocol analyzer interface for line 1 is enabled to local IP address
; 127.0.0.1, port number 10001

#SET NET1:SAO10="127.0.0.1"

; A protocol analyzer interface for line 10 is enabled to local IP
address 127.0.0.1, default port number 50010 is used

#SET NET1:SAO1=""

; Protocol analyzer interface for line 1 is disabled
```

Note: The given address and port number should not overlap with other TCP servers. If this situation occurs and the creation of the server fails, no error code is returned, but the AO attribute will return an empty string if read. Assigning a different port number is needed in this situation. It is recommended to keep analyzer interface disabled when not used.

16.2.4.2 AU Analyzer Usage

This attribute is a bit pattern, which defines the type of the output.

Bit 0: If this bit is 1, the contents of the data messages are printed out. This is the default operation. If this bit is 0, the contents of the data messages are not printed out.

Bit 1: If this bit is 1, the changes in the control signals (RTS, CTS, DCD, DTR) of the RS232 lines and the changes in the TCP connection states with TCP protocols are printed out. This is the default operation. If this bit is 0, the changes in control signals are not printed out.

Bit 2: If this bit is 1, the internal printouts are enabled. This option can be used to detect the completion timepoint of the of the write operation with the serial hardware. The completion timepoint defines the behavior of the RTS signal of the line.

Bit 3: If this bit is 1, the contents of the data messages are printed out in ASCII format. This applies to decimal numbers 32 – 126, which contain the ASCII printable characters. All other decimal numbers are printed in hexadecimal format. If this bit is 0, the contents of the data messages are printed out in hexadecimal format. This is the default operation. The ASCII printout format can be useful when using SPA, LON or Modbus ASCII protocols.

Bits 4..15: Not used, do not set.

Data type:	Integer
Value:	0 ... 65535
Indexing:	NET line number
Default value:	3
Access:	Read, write

Examples:

```
#SET NET1:SAU1=0                ; printout is disabled
#SET NET1:SAU1=3                    ; both data and control signals are
                                printed out
```

16.3 Data transmission attributes

16.3.1 BR Baud Rate

Transmission rate used on the line. The attribute is valid for all serial lines and all types of protocols using serial lines.

Data type:	Integer
Value:	1 ... 19200
Unit:	Bits/s
Indexing:	Line number
Default:	Protocol dependent, usually 300, 600, 1200, 2400, 4800, 9600 or 19200. The given value is passed as such to the serial hardware in use and thus, the support is hardware dependent. Values 384, 560, 576, 1152 and 2560 are used indicate speeds 38400, 56000, 57600, 115200 and 256000.
Access:	Read, conditional write

Example:

The baud rate of line 4 in NET 2. set to 9600 bps:

```
#SET NET2:SBR4 = 9600
```

Example:

Example on configuration of Modbus Master line:

```
¦ PO Protocol                     00025 ¦
¦ IU In Use                       00000 ¦
¦ MS Message Application          00001 ¦
¦ MI Message Ident.               00000 ¦
¦ LT Link Type                    00000 ¦
¦ BR Baud Rate                    09600 ¦
¦ SB Stop Bits                    00001 ¦
¦ PY Parity                       00002 ¦
¦ RD Receiver Data Bits           00008 ¦
¦ TD Transm. Data Bits            00008 ¦
¦ OS Output Synchroniz.           00001 ¦
¦ RE Redundancy                   00000 ¦
¦ TI Timeout Length               02000 ¦
¦ NA NAK Limit                    00000 ¦
¦ EN ENQ Limit                    00003 ¦
¦ DE CTS Delay Length             00050 ¦
¦ ER Embedded Response            00000 ¦
¦ RP Reply Poll Count             00000 ¦
¦ PD Poll Delay                   00300 ¦
¦ PS Buffer Pool Size             00016 ¦
¦ PP Polling Period               00001 ¦
¦ CN Connection                   [Ign] ¦
```

16.3.2 ER Embedded Response

Indicates if NET transmits embedded responses (ACK, NAK) and ENQ:s within the messages, see the illustration in Figure 18. The attribute applies to ANSI X3.28 Full Duplex and ACP lines (PO = 1).

The use of embedded responses increases the performance of a line especially when there is heavy simultaneous communication in both directions. The ER attribute affects only the transmitter of the communication unit. The NET unit is always able to pick embedded responses and ENQs from received messages, independent of the ER value. Some ANSI X3.28 station types may lack support for Embedded Response.

Data type:	Integer	
Value:	0	No embedded responses transmitted
	1	Embedded responses are transmitted
Indexing:	Line number	
Access:	Read, conditional write	

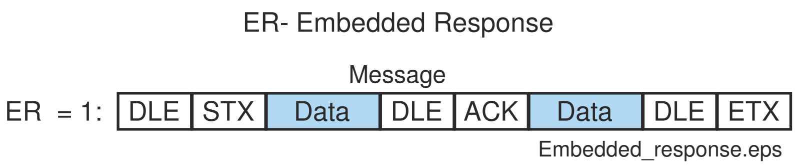


Figure 18: An illustration of the ER attribute

16.3.3 PY Parity

The parity check (if any) used for the characters transferred on the line. The attribute is essential for all types of protocols using serial hardware.

Data type:	Integer	
Value:	0	No parity check
	1	Odd parity
	2	Even parity
Indexing:	Line number	
Default:	Protocol dependent, the usage of the default value is recommended. For ANSI X3.28 lines, even parity should be used if the checksum used (the RE attribute) is BCC (Block Check Character). If CRC-16 (Cyclic Redundancy Check) is used, no parity check is needed.	
Access:	Read, conditional write	

16.3.4 RD Receiver Data Bit Count

The number of data bits in each received character. The attribute is valid for all protocols using serial hardware.

Data type:	Integer
Value:	5, 6, 7 or 8
Unit:	Data bits
Indexing:	Line number
Default:	Protocol dependent, the usage of the default value is recommended.
Access:	Read, conditional write

16.3.5 RE Redundancy

The type of checksum added to each message. NET knows two types of checksums: CRC-16 and BCC.

The attribute applies to the ANSI X3.28 full and half duplex. On P214 and RP570 lines, the checksum is always BCC.

Data type:	Integer	
Value:	0	No checksum
	1	CRC-16 (Cyclic Redundancy Check). Possible only on ANSI X3.28 lines.
	2	BCC (Block Check Character)
Default:	2 (BCC) On an ANSI X3.28, line one of the checksum types must always be used. The RE value affects only the checksum type of message packets. Thus, the checksum used in the polling packets of the half duplex protocol is always a BCC character (even though RE would be 1).	
Indexing:	Line number	
Access:	Read, conditional write	

16.3.6 SB Stop Bits

The number of stop bits attached to each transmitted byte. The attribute applies to all protocols using serial hardware.

Data type:	Integer
Value:	1 … 3
Unit:	Bits
Default:	Protocol dependent, the usage of the default value is recommended.
Indexing:	Line number
Access:	Read, conditional write

16.3.7 TD Transmitter Data Bit Count

The number of data bits in each transmitted character. The attribute is essential for all protocols using serial hardware.

Data type:	Integer
Value:	5, 6, 7 or 8
Unit:	Bits
Default:	Protocol dependent, the usage of the default value is recommended.
Indexing:	Line number
Access:	Read, conditional write

16.4 Communication control attributes

16.4.1 DE CTS Delay

Time delay between the activation of the RTS signal (Request to Send) and the start of a new transmission, see Figure 19. The attribute has somewhat different functions depending on the protocol used on the line.

On ANSI lines:

The half duplex communication is controlled by the RTS and CTS (Clear to Send) signals of the V.24 interface. When the NET unit has something to transmit (ACK, NAK, a polling packet or a message), it activates the RTS (Request to Send) signal. The NET unit also waits for the CTS (Clear to Send) signal to become active. In some cases, e.g. on some radio lines, the activation of CTS does not guarantee that transmitted data will go through to the receiver. For example, switching on the carrier may need extra time.

The DE attribute defines the delay (in milliseconds) from the activation of RTS until the transmission is started. If DE = 0, the transmission starts immediately when CTS is activated. If DE is larger than 0, the time indicated by DE is waited. When this time has run out, the transmission starts if CTS is active.

On RP570 lines:

If the TW attribute (see below) is 0, the DE attribute controls both the CTS wait time and the transmission delay. If the line does not get CTS (Clear To Send) after DE milliseconds have elapsed, it will return the message including an error code to the sender. If DE = 0 the line will start transmitting immediately after it has got the CTS. However, it will wait for at most 500 ms. If the TW attribute has a value greater than 0, then the transmission delay is controlled by the TW attribute, and the DE attribute controls only the CTS waiting time.

On P214 lines: Delay after receiving CTS before starting the telegram transmission.

Data type:	Integer
Value:	0 ... 65535
Unit:	Milliseconds
Suggested value:	Up to 500 ms, depending on the link type used. When a fiber optic modem is used, which is normally the case on SPA lines, no CTS delay is required.
Default value:	50 ms for IEC 60870-5-101 and DNP V3.00 protocols
Indexing:	Line number
Access:	No restrictions

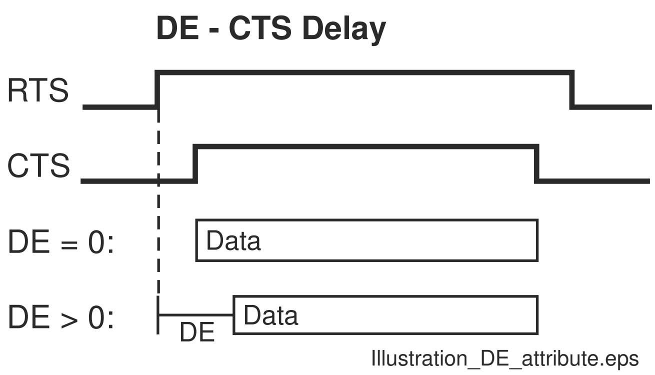


Figure 19: An illustration of the DE attribute on ANSI Half Duplex and RP570 lines

16.4.2 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 with any protocol. See also line attribute CM (Com Port Mode), bit 3.

Value:	0 ... 20
Unit:	Bytes (absolute time depends on baudrate)
Access:	Read/Conditional Write
Default:	1

Some notifications concerning this attribute:

For the standard serial port of a PC the value must be 1 or more. With a standard serial driver, the functionality of the write operation may be dependent on multiple factors such as transmit FIFO buffer setting. Testing of different settings may be necessary. The setting of the bit 3 of the line attribute CM (Com Port Mode) should be considered.

For the Rocket port serial card the value can also be 0 because the write operation is seen as complete not until all bytes are actually sent. This applies only if the 'Wait on physical transmission before completing write' flag is set in the driver configuration.

For the Digi Neo serial port card the value can also be 2 or more. With the serial driver the write operation is seen as complete when there is still two bytes to be sent. Thus, value 0 will cause the RTS to be in nonsignaled state before the message is completely sent.

16.4.3 EN Enquiry Limit

The maximum number of times that a telegram is retransmitted after a timeout. The attribute applies to the ANSI X3.28 full and half duplex and the RP570 protocols.

On RP570 lines a timeout occurs when an RTU fails to respond with a correct response telegram within the time specified by the HT attribute. When the message has been sent the number of times specified by the EN attribute, the transmission is considered as failed and the RTU is suspended. The line returns a command including an error code to its sender and the telegram transmission is no more repeated. On RP570 lines NET starts to send SCIs (Status Check Instructions) to the suspended RTU.

On ANSI full duplex lines time-out occurs when no ACK or NAK is received to a transmitted message within the time defined by the TI attribute. The NET unit transmits an enquiry (ENQ) at most the number of times stated by the EN value. If no response is received, the NET unit refrains from further retries.

On ANSI half duplex lines time-out occurs when no ACK is received to a transmitted message, or no response is received (EOT or message) to a polling packet within the time defined by TI. If no ACK is received, the NET unit retransmits the message until the EN limit is reached. If a station connected to a half duplex line does not respond to a polling packet, it is polled at most the number of times stated by the EN value. After that, the next station will be polled.

Data type:	Integer
Value:	1 ... 10
Suggested value:	1 ... 4 (2 ... 3 on RP570 lines)
Default value:	2 for DNP V3.00 protocol
	3 for IEC 60870-5-101
Indexing:	Line number
Access:	Read, conditional write

16.4.4 HT Header Timeout

With protocols other than ANSI Half duplex, HT is the maximum waiting time in milliseconds within which the first byte of a response from the RTU should have been received after the transmission of a message. If no response has been received within this time, new attempts are performed the number of times specified by the Enquiry limit. If still no response is obtained, the station is suspended. Figure 20 illustrates the HT and the TI attributes.

On SPA lines: Max. transmission time of the complete message is calculated automatically based on the message length and baud rate.

On RP570 lines: If TI = 0, the reception time is calculated automatically. If both HT and TI are = 0, the program sets HT to 700 ms. If HT == 0 and TI > 0, there is no separate header timeout supervision, but the entire message must be received within TI seconds from the end of NET's transmission.

On ANSI Half Duplex lines: The HT (if >0) or TI attribute value is always used as a timeout for the complete message, since the length of the received message cannot be calculated.

Data type:	Integer
Value:	0 ... 65535
Unit:	Milliseconds
Indexing:	Line Number
Default value:	Protocol dependent
Access:	Read, conditional write

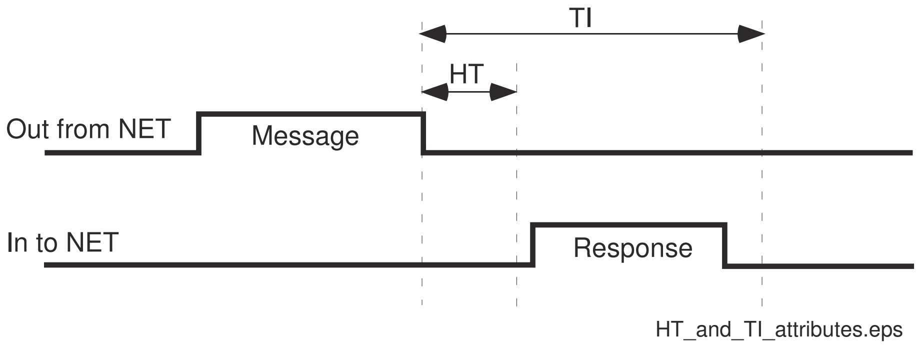


Figure 20: An illustration of the HT and TI attributes

16.4.5 NA NAK Limit

The number of NAK responses that the line accepts at the transmission of a message, without considering the transmission failed. The attribute is essential only for ANSI X3.28 Full Duplex (ACP).

Data type:	Integer
Value:	1 ... 255
Indexing:	Line number
Suggested value:	3
Access:	No restrictions

16.4.6 OS Output Synchronization

States the flow control principle used by the printer. Hence it states how the printer informs NET that the reception buffer is full and the character transfer should temporarily cease. It also states that there is enough free space in the buffer for the data transfer to be restarted. The OS attribute applies to printer lines only.

The cable wiring of the printer connection differs depending on the OS attribute, see the Installation Manual.

Data type:	Integer	
Value:	0	No synchronization (no flow control applied)
	1	XON/XOFF synchronization (the printer transmits XON when it is able to receive and XOFF when the buffer is full)
	2	CTS Synchronization with DCD monitoring. The printer tells by activating and passivating the CTS signal, when the buffer can be filled/is full

Table continues on next page

	3	XON/XOFF with DCD supervision. A system message is produced if NET looses contact with the printer.
Suggested value:	1, 2 or 3. Do not use OS = 1 on printers used by Hot Stand-by base systems.	
Access:	Read, conditional write	

16.4.7 PD Repeat Time Delay

LonTalk protocol bus repeat timer. This is given as a code in the range 0...15 as defined in NEURON Chip Distributed Communication and Control Processors. The repeat timer is not used with transparent SPA messages in LonTalk protocol.

Data type:	Integer
Values:	0 ... 15
Indexing:	Line number
Access:	Read, conditional write

16.4.8 RI Receive Interrupt Enable Delay

Defines when the receiver of a NET line should be enabled after a message has been issued. The attribute applies to P214 and RP570/571 lines.

Data type:	Integer	
Value:	0 ... 65535	
	0 ...255 ms for IEC 60870-5 and for DNP V3.00 protocol	
	0	Receiver always enabled
	1 ... 9	Receiver enabled by the link layer process when message has been completely transmitted
	10 ... 65535	Delay time in milliseconds between the completion of the transmission and the enabling of the receiver
Indexing:	Line number	
Default value:	0	
	0 for IEC 60870-5-101 protocol (balanced mode) and DNP V3.00 protocol	
	5 for IEC 60870-5-101 protocol (unbalanced mode)	
Access:	No restrictions	

16.4.9 RK RTS Keep Up Padding Characters

The number of padding characters (null characters) inserted after the message to the end of telegram to delay the passivation of the RTS signal (Request To Send). See Figure 21.

With some modem circuit types, the data bytes are delayed much more than the handshaking signal state changes. This means that if RTS is passivated immediately when the last byte is transmitted to the modem, the carrier will be broken before the last byte is transmitted by the modem. By inserting padding characters after the message, the passivation of RTS can be delayed to give the modem enough time to transmit all characters belonging to the message. The number of padding characters is given by the RK attribute. The extra delay needed by the modem is about two bytes.

The attribute is valid for all ANSI Half Duplex, IEC 60870-5 and RP570 modem lines.

Attribute RY RTS Keepup Delay can be used instead of RK.

Data type:	Integer
Value:	0 ... 255. Number of padding characters
Unit:	Padding characters
Indexing:	Line number
Default value:	0
Access:	Read, conditional write

Figure 21: An illustration of the RK attribute

16.4.10 RL Retry Limit

On P214 lines:	Defines how many times in a sequence a telegram may be transmitted to a station before giving up
On SPA lines:	Number of telegram repetitions before moving the station into the sub cycle (suspended). Telegrams to stations in the sub cycle will not be repeated.
Data type:	Integer
Value:	1 ... 255
Default value:	2 (SPA)
Indexing:	Line number
Access:	No restrictions

16.4.11 CB Carrier Blocking

This attribute determines whether the incoming Carrier Detect (DCD) signal of the serial port must be set in order for the line to receive messages. This attribute is available only with IEC60870-5-101 and IEC60870-5-103 protocols. See protocol specific manual for details. The same functionality can be configured using line attribute CM, bit 2, which is supported with all serial protocols.

Data type:	Integer
Value:	0 = Carrier blocking not used, messages are received regardless of the DCD state. 1 = Carrier blocking used, DCD must be signaled in order to receive messages. Attribute is available only with in IEC60870-5-101 / 103 protocols.
Indexing:	Line number
Access:	Read, conditional write

16.4.12 CM COM Port Mode

This attribute consists of a set of flags which control the behaviour and functionality of the serial port of the line. This attribute is available on all protocols using the serial line. Each flag is one bit of this attribute.

Data type:	Integer
Value:	0 ... 15 (see below)
Default value:	0
Indexing:	Line number
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 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 'high' state all the time and the protocol configured for the line behaves according to that. This setting may be necessary with the virtual serial port or for easier cabling. Bit 1 can also be controlled using line attribute SG. See line attribute DE how transmission starts when CTS is constantly 'high'.

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 'high' state all the time and the protocol configured for the line behaves according to that. This setting may be necessary with the virtual serial port or for easier cabling. Bit 2 can also be controlled using line attribute SG.

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 baudrate 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.

Bit 4: No RTS Control

When this bit is 0, RTS pin is controlled as configured for the protocol. Usually the RTS pin is set to 'high' before message is transmitted.

When this bit is 1, RTS pin is not controlled. This setup may be useful with virtual serial ports, especially if the RTS signaling is not required by the modem devices connected to virtual serial port server device. Network delays and/or a malfunction in virtual serial port server or driver may cause delays also for the RTS pin controlling. These delays may be seen as a disconnection between PC-NET node and base system. If bit 4 is set in one line, it is strongly recommended to use it for all lines using the same virtual serial device. Bit 4 can be set for the non-virtual serial devices, too, but depending on the characteristics of the serial device and connected modem devices, the communication does not necessarily work at all. Thus, the setting of bit 4 with non-virtual devices is useful in very rare cases.

Note for bits 1 and 2
The different protocols are handling the DCD and CTS differently. See also line attribute descriptions LK, DE and CB for the protocol in question. Having a simulated value in CTS or DCD may have an effect how RS-232 line disconnection is detected and reported to the MicroSCADA application.

16.4.13 SG Modem Signal

The direct supervision and control of the state of the modem signal. The attribute applies to all protocols. It is used for diagnostics and testing.

If the incoming signal DCD or CTS is wanted 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 the virtual serial ports. If value 0 is written to these signals, the actual state of signal will be used. The default mode of operation is the actual state.

Data type:	Integer	
Value:	0	Passive
	1	Active
Incoming:	DCD and CTS signals	
Outgoing:	DTR signal	
Indexing:	100 * line nr + 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 behaves as DCD is 'high' all the
time #SET NET1:SSG205 = 1 ;Line 2 of NET1 behaves as CTS is 'high' all
the time #SET NET1:SSG208 = 0 ;Line 2 of NET1 uses the real state of the
DCD #SET NET1:SSG205 = 0 ;Line 2 of NET1 uses the real state of the CTS
```

The different protocols are handling the DCD and CTS differently. See also line attribute descriptions LK, DE and CB for the protocol in question. Having a simulated value in CTS or DCD may have an effect how RS-232 line disconnection is detected and reported to the MicroSCADA application.

16.4.14 TI Timeout Length

The time limit applied when the NET unit is waiting for response to a message or polling packet. The attribute applies to the ACP, ANSI and P214 protocols and to General ASCII when TAIP sync format is used (PM = 6 and SF = 6).

The value of TI is the time in seconds, that NET waits for acknowledgements (ACK or NAK on full duplex lines, ACK on half duplex lines). TI also states the maximum time NET waits for response to a polling packet on half duplex lines (EOT or message). The NET unit starts the response wait timer when the last byte of a message or polling packet has been transmitted to the receiving station.

TI is used to specify maximum reception time, while the HT attribute should be used to check whether a station is responding or not. The timeout on SPA and RP570 lines are specified exclusively by the HT attribute (see above). See the illustration of the HT and TI attributes in Figure 20.

On General ASCII lines with PM = 6, the TI attribute is the time NET waits for clock synchronization messages. If NET receives no synchronization message within this time, it starts to reconfigure the clock device with the time interval given with the PD attribute above and sets the SS attribute to Time invalid.

LonTalk protocol lines have a TI attribute with another meaning, see below.

Data type:	Integer	
Value:	1 ... 255	
Unit:	Seconds	
Suggested value:	ANSI Full duplex:	1
	ANSI Half duplex:	Depending on the transmission rate used:
		300 bits/s: 8
		600 bits/s: 4
		1200 bits/s: 2
		2400 bits/s: 1
		Anyhow, the TI value should be larger than (11 * (byte count in longest message)) / (transmission rate)
	General ASCII:	The timeout must be longer than the time between sync messages (defined in the clock). A suitable value could be 30 seconds.
Indexing:	Line number	
Access:	No restrictions	

16.4.15 TI Timeout Between Retries

Time between retries when acknowledged or request/response service is used. This is given as a code in the range [0..15] as defined in NEURON Chip Distributed Communication and Control Processors. Applies to LonTalk protocol lines only. The timeout of transparent SPA messages in LonTalk protocol is defined with line attribute TW, see corresponding description for more information.

Data type:	Integer
Values:	0 ... 15
Indexing:	Line number
Access:	Read, conditional write

16.4.16 TW Transmission Wait Delay

The transmission delay in milliseconds, i.e., the time that the NET must wait after receiving a CTS signal until starting the transmission of a message.

If TW = 0, the DE attribute controls both the CTS wait time and the transmissions delay as before. If the TW attribute is greater than 0, it specifies the transmission delay while the DE attribute specifies the maximum waiting time for the CTS signal.

The attribute applies to the following protocols:

- RP570/RP571 master
- ANSI
- IEC60870-5-101/103 master and slave
- DNP 3.0 master and slave
- RCOM master

Data type:	Integer
Value:	0 ... 30000
	0 ... 65535 ms for IEC 60870-5-101, DNP V3.00 and RCOM protocols
Unit:	Milliseconds
Access:	No restrictions

With LON line the TW attribute has a different meaning, see below.

16.4.17 TW Acknowledge Timeout of Transparent SPA Messages

LON acknowledgement timeout given as a code in the range [0..15] used with transparent SPA messages.

Data Type :	Integer	
Value:	0 ... 15:	
	0	16 ms
	1	24 ms
	2	32 ms
	3	48 ms
	4	64 ms
	5	96 ms
	6	128 ms
	7	192 ms
	8	256 ms
	9	384 ms
	10	512 ms
	11	768 ms
	12	1 024 ms
	13	1 536 ms
	14	2 048 ms
	15	3 072 ms
Default:	2 (= 32 msecs)	
Access:	Read, conditional write	

16.4.18 OM Operating Mode

This attribute consists of a set of flags which control the behaviour and functionality of the line. From the protocols which are described in this manual, OM attribute is available only in ANSI X3.28 half duplex protocol. Each flag is one bit of this attribute, see detailed descriptions below.

Data type:	Integer
Value:	0 ... 65535 (see below)
Indexing:	1..12 (Line number)
Default value:	0
Access:	Read, conditional write

ANSI X3.28 half duplex:

Bit 0: No suspension in reply timeout

When this bit is 1, the STA object is not set to a SUSPENDED state when a request goes to timeout defined with station attribute RT or if the IED does not acknowledge the request. The process data from the station also remains as valid. The setting of this bit recommended if there is a separate IED connected to each STA object. When this bit is 0 and a request goes to timeout i.e. statuses 12337 = STAP_REPLY_TIMEOUT or 16205 = XHCP_TIMEOUT_WHILE_WAITING_ACK is returned to SCIL, the station object is also set to SUSPENDED state and the process objects are set to invalid state. This is the default mode of operation and should be used with the gateway devices.

Bit 1: No link layer retries for 'Write' messages

When this bit is 1, no link layer retries are made for all messages of type 'Write' from the application layer. The setting of this bit is recommended e.g. if the message contains time synchronization information or if the retransmission of the message is implemented to the SCIL application. When this bit is 0 and a no ACK message is received from the remote device to a the request, the link layer retries are sent according to the line attribute EN. This is the default mode of operation.

Bit 2: No link layer retries for 'Read' messages

When this bit is 1, no link layer retries are made for all messages of type 'Read' from the application layer. The setting of this bit is recommended e.g. if the retransmission of the message is implemented to the SCIL application or if the retransmissions cause problems in the IED or in the transmission media. When this bit is 0 and a no ACK message is received from the remote device to the request, the link layer retries are sent according to the line attribute EN. This is the default mode of operation.

Bit 3: No link layer retries for 'Reply' messages

When this bit is 1, no link layer retries are made for all messages of type 'Reply' from the application layer. The setting of this bit is recommended e.g. if the retransmission of the message is implemented to the SCIL application or if the retransmissions cause problems in the IED or in the transmission media. When this bit is 0 and a no ACK message is received from the remote device to the request, the link layer retries are sent according to the line attribute EN. This is the default mode of operation.

Example for ANSI X3.28 half duplex:

IEDs are time synchronized from SCIL application and data is also requested cyclically from SCIL by using #GET or by reading the ME-attribute. The recommended value for OM is

```
#SET NET1:SOM2=3    ; No suspension in RT timeout and no retries for
control commands and timesyncronization
```

16.4.19 TB Transmit Bytes

It is possible to send control characters to serial communication line. The functional purpose of these characters is to control the switches located in the line. The functionality is common for all the communication protocols. When this attribute is written, the given bytes are sent to the serial line.

Value :	Vector of bytes, given as string or bytes separated with colons
Access:	Write only

At the beginning of the vector, bytes 0 (NULL character) and 32 (Space character) have a special meaning:

- each NULL character at the beginning of the message results in a 10-byte preceeding delay in the transmission.
- each SPACE character at the beginning of the message results in a 1-byte preceeding delay in the transmission.

The NULL character cannot be entered as string. See the examples below.

The length of the vector is limited by the lenght of the SCIL statement i.e. 255 bytes.

After the dial-up connection is established, the SCIL sequence could be the following:

```
#PAUSE 1
#SET NETx:STB'line'=(64,49)    ; send "@1" to switch to select COM1
#PAUSE 1
#SET STA'sta':siu=1 ; polling starts
```

The control characters can be transmitted during normal communication as well.

```
#SET NETx:STB'line' = (0,0,0,64,49) ; send "@1"  with preceeding 30-byte
delay in transmission
#SET NETx:STB'line' = "   @1" ; send "@1"  with preceeding 3-byte delay
in transmission
#SET NETx:STB'line' = (0,32,32,32,32,32,64,49) ; send "@1"  with
preceeding 15-byte delay in transmission
```

16.5 Polling attributes

Polling is performed on ANSI X3.28 Half Duplex, SPA, RP570, IEC 60870-5 and P214 lines (PO = 2, 14, 7 or 9). If nothing else is mentioned, the attributes described in this section are essential for these protocols. The PD attribute applies also to General ASCII with PM = 6. The polling attributes have no meaning for lines using other protocols.

The example below illustrates the polling of an ANSI station. It is supposed that STA4 replies to the second reply poll. Suppose that the base system issues a reply message for STA4, when the NET is polling STA2. The polling will continue as follows:

Table 5: Polling of an ANSI station:

NET	STA1	STA2	STA3	STA4
DLE ENQ 1 BCC				
	DLE EOT			
DLE ENQ 2 BCC				
8		DLE EOT		
REPLY TO 4				
				DLE ACK
DLE ENQ 3 BCC				
			DLE EOT	
DLE ENQ 4 BCC				
				DLE EOT
etc.				

If the message from the base system to STA4 is a command, the polling continues as follows:

Table 6: Polling of an ANSI station after a command to STA4 from the base system:

NET	STA1	STA2	STA3	STA4
DLE ENQ 1 BCC				
	DLE EOT			
DLE ENQ 2 BCC				
		DLE EOT		
COMMAND TO 4				
				DLE ACK
DLE ENQ 4 BCC				
				DLE EOT
DLE ENQ 4 BCC				
				REPLY
DLE ACK				
DLE ENQ 3 BCC				
			DLE EOT	
DLE ENQ 4 BCC				
				DLE EOT
etc.				

16.5.1 AW Application Wait Time

The time that the line waits for an acknowledgement or response from an application to a data telegram or message sent from an RTU. The attribute applies to RP570 and P214 lines.

If no acknowledgement is received within this time, due to a loss of connection with the base system, the application is suspended. The polling of all stations allocated by the application is stopped. The NET unit starts to send diagnostic commands to the base system and when it has recovered, the polling of the stations continues.

Data type:	Integer
Value:	1 ... 65535
Unit:	Milliseconds
Default value:	5000
Indexing:	Line number
Access:	No restrictions

16.5.2 CP Command Poll Count

Defines how many times in a sequence a station will be polled when receiving an event from it. The attribute applies to P214 lines only.

Data type:	Integer
Value:	0 ... 255
Access:	Read, conditional write

16.5.3 PD Poll Delay

Delay between polling telegrams.

When the NET unit is polling stations, it inserts a delay between the response to a poll and the transmission of the next polling packet. The length of this delay (in milliseconds) is given by the PD attribute. The purpose of the poll delay is to prevent the polling from overloading the communication unit.

A message sent from NET to a station during the poll delay time is serviced immediately.

On general ASCII lines using TAIP clock sync format (PM = 6 and SF = 6), the PD attribute determines the time between configuration messages to the clock. The time is determined in situations when NET has not received clock synchronization messages within the time specified by the TI attribute.

IEC 60870-5: The delay between polling messages. The purpose of this attribute depends on the IEC protocol mode. The unbalanced slave uses this attribute only to detect if the master is polling it. The unbalanced master sends the polling messages (for class 1 or class 2) with interval of PD attribute. With balanced mode, the link checks that communication is alive if the time between messages is more than PD time.

Data type:	Integer
Value:	0 ... 65535
Unit:	Milliseconds (ms)
Indexing:	Line number
Suggested value:	Min. 100 (100 ms for RP570 and P214)
Default values:	50 for IEC 60870-5-101 master protocol (unbalanced mode)
	500 ms for DNP V3.00 protocol
	5000 for IEC 60870-5-101 slave and master protocols (balanced 30000 for IEC 60870-5-101 slave protocol (unbalanced mode) mode)
Access:	Read, conditional write

16.5.4 PL Polling Limit

Controlling the polling sequence of unbalanced master protocol. The purpose of PL attribute is to limit the number of successive polls of same link address. Normally the MicroNET polls the same link address until the all data from the same link address is read. This attribute is used only for unbalanced IEC 60870-5-101 master mode.

Data type:	Integer
Value:	2 … 100
Indexing:	Line number
Default:	10
Access:	No restrictions

16.5.5 PP Polling Period

The polling frequency of suspended stations. The attribute specifies how often suspended stations on the line are polled. Normally, the NET unit is continuously polling the stations. Each station gets the permission to transmit, when its turn comes. The polling of a station stops completely when it is taken out of use (the IU attribute is set to 0).

On RP570 and P214 lines: A polling cycle is completed, when all stations have been polled at least once (RTUs responding with priority 1 information are polled several times in a row). When a polling cycle is completed, another one starts. A suspended station is polled every PP:th beginning of a polling cycle.

On SPA lines: A polling cycle is completed, when the stations have been polled as defined with attribute PT Polling Ratio. One suspended SPA station is polled at the end of every PP:th polling cycle.

On ANSI Half Duplex lines: The station can respond to a polling packet with a message, or if it has nothing to transmit, with an EOT. If the number of polls specified by the EN attribute (see above) are transmitted to the station and no response is received, the station is suspended (is classified as faulty). The suspended stations are polled less often than other stations. The PP attribute specifies the number of poll cycles completed between each poll to a suspended station. Only one suspended station per PP number of poll cycles is polled. See also the PC attribute above.

When a suspended station responds to a poll, the suspension is cancelled if the suspension reason was that the station did not respond to polls (suspensions due to other reasons are not affected). From that moment it will be polled in every polling cycle.

The PP attribute is supported also by protocols IEC 60870-5-101 master (unbalanced mode), IEC 60870-5-103 master, DNP3.0 Master and Modbus Master. See protocol specific manuals for more information.

Data type:	Integer
Value:	1 ... 255
Indexing:	Line number
Suggested value:	2 ... 20 (with RP570, SPA, P214 and ANSI Half duplex)

Table continues on next page

Default value:	5 for RP570 master, 10 for SPA, 20 for ANSI Half duplex and P214
Access:	Read, conditional write
Example:	Four ANSI stations, denoted 1, 2, 3 and 4, connected to a line, are polled as follows: 1, 2, 3, 4, 1, 2, 3, 4 If a station does not respond to its poll, the it is polled again immediately. If it still does not respond, it will be polled until the total number of polls is equal to the EN attribute. If no response is received, the station is suspended. Suppose that a fault occurs on station 3, and it stops responding. The following poll sequence will be transmitted by NET (EN = 3): 1, 2, 3, 3, 3, 4, 1, 2, 4 Those stations classified as suspended are polled less frequently than those, which respond continuously. After the completion of each polling round, at most one suspended station is polled. The PP attribute gives the number of entire polling rounds to be completed without polling any suspended station. If the stations 2 and 3 are faulty and PP = 1, we get the sequence: 2, 1, 4, 3, 1, 4, 2, 1, 4, 3, 1, 4 For each polling round, one faulty station is polled. If PP = 2, the polling sequence will be: 1, 4, 1, 4, 2, 1, 4, 1, 4, 3, 1, 4, 1, 4, 2 Notice that a suspended station is polled only once although it does not respond to the poll.

16.5.6 PT Polling Ratio

A poll ratio concept is used by NET when requesting data and events from SPACOM units. This means that a certain pattern of poll messages is sent cyclically. During one poll cycle NET polls a certain number of SPA units for data and events. Each slave (station) has a certain event poll priority class, 1 or 2, which is defined by the STAn:SEP attribute, see Section 17.

The PT attribute specifies how many stations of each event poll class NET polls during a poll cycle and how many data polls NET performs during a poll cycle.

If suspended SPACOM units exist, such a unit may be interrogated when a certain number of poll cycles have been completed. The PP attribute, see above, defines how often this is done.

The PT attribute applies to SPA lines only.

Data type:	Integer	
Value:	0 ... 10	
Unit:	Polls per cycle	
Indexing:	(100 x line number) + poll type where 'poll type' = 1, 2 or 3:	
	1	Event buffer polls from priority class 1
	2	Event buffer polls from priority class 2
	3	Data polls
Default values:	Poll type 1	4
	Poll type 2	2
	Poll type 3	1
Access:	No restrictions	

Example:

Each poll cycle on line 3 will contain 2 data polls:

```
#SET NET1:SPT303 = 2
```

16.5.7 RP Reply Polling

The attribute applies to ANSI half duplex and RP570 lines.

On RP570 lines:	Specifies the maximum number of consecutive polls sent to a station while waiting for the process response to an object command or an analog setpoint
On ANSI lines:	Maximum number of consecutive polls sent to a station while the NET unit is waiting for a reply

The value of the RP attribute sets a limit to the number of reply polls transmitted to a station after a command has been transmitted to it. If e.g. RP = 3, the receiver station will be polled 3 times for a reply, after a command message has been transmitted to it. If no reply is received, the station is classified as faulty and the next station will be polled.

Messages from the base systems and stations will always override the polling. Therefore, if a message from the base system is to be transmitted to a station on a half duplex link, NET transmits this message when the current polling packet gets a response or a time-out occurs. After the message transmission, NET continues to poll the station in turn. However, if the message from the base system (or a station) was a command message, the receiving station will first be polled for a reply to the command transmitted.

IEC 60870-5:	This attribute is used only with unbalanced master protocol mode IEC 60870-5-101
Data type:	Integer
Value:	0 ... 255
Suggested value:	2 ... 8
Default value:	1 for IEC 60870-5-101
Access:	No restrictions

16.6 Autodialing attributes

The following attributes are significant only when an autocaller (a modem with functions for automatic dial-up) is connected to the line. The availability of these attributes are controlled by the AC attribute of the line.

Autodialing is possible only on ACP, ANSI X3.28, RP570 master, SPA, Modbus, IEC 61107, Alpha, IEC 60870-5-101 master and slave, IEC 60870-5-103 master, DNP 3.0 master and slave protocols. The link type (the LK attribute) must be 1 or 2.

The dial-up connection may be initiated by the PC-NET or by the remote IED or system.

16.6.1 AC ACE

States whether an autocaller is connected to the line or not. The autocaller must use the AT (Hayes) command set.

Data type:	Integer	
Value:	0	No autocaller
	1	Yes, autocaller is connected
Default value:	0	
Access:	No restrictions	

16.6.2 AS ACE State

The state of the autocaller.

Data type:	Integer	
Value:	0	IDLE, the ACE is ready to make a call
	1	CONNECTED, transmission is activated
	2	BUSY, the ACE is busy, e.g., dialling
	3	INITIAL, the ACE is uninitialized
	4	CONFIGURE, the IU attribute of the line is set to 0
Access:	Read-only	

16.6.3 CL Connection Time Limited

Makes it possible to put a time limit for the duration of the connection. The value of this limit is given by the attribute CT.

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

16.6.4 CN Connection

Dialling devices from NET and for breaking telephone connections. Using CN presupposes that there is an autocaller connected to the line (AC = 1).

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

When dialling a station, the station number should be given at the end of the phone number string, preceded by the letter "S". This option is normally used to increase the communication performance on multidrop lines.

Data type:	Text
Values:	Max. 25 characters:
	Dialling: The telephone number, and possible the station number preceded by an S. A leading zero and fill characters such as "/", "-" and " " may be used. By inserting comma characters (","), delays may be inserted into the dialling. In IEC 60870 master protocols, the value that is given is the address of the station, which the user wants to connect to.
	Breaking the connection: An empty string (" ")
Access:	No restrictions

Example:

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

16.6.5 CS Connected Station

Indicates which station NET is communicating with. The attribute is used on incoming calls on ANSI X3.28 half duplex and RP570 lines. With IEC 60870 master protocols, if the station is explicitly defined in writing to the CN attribute, the returned value is the link address of the polled station.

Data type:	Integer
Value:	0 ... 255. The STA object number
	0 = No station
Access:	Read-only

16.6.6 CT Connection Time

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

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

16.6.7 DD Radio Disconnection Delay

Delay between last data transfer and line disconnection.

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

16.6.8 MC Modem Command

Using this attribute, a modem can be controlled directly from SCIL with AT/Hayes commands. When an AT command is written to the MC attribute it is transmitted to the modem on the line. Modem commands can only be sent when the line is not in use.

Data type:	Text
Value:	Write: An AT/Hayes command
	Read: The response from the latest command
Access:	No restrictions

Example:

With AT/Hayes command ATS0=3 you can set the register S0 value in the modem to value 3, this is the amount of ringsignals before the modem accepts the call. See the Modem Manual for more information.

Line 1 on NET3 is used for dial up line in the example.

```
#SET NET3:SIU1=0
#SET NET3:SMC1="ATS0=3"
#SET NET3:SIU1=1
```

16.6.9 PU Pulse Dialing

The dialing principle used.

Data type:	Integer	
Value:	0	Tone dialling
	1	Pulse dialling
Default value:	0	
Access:	No restrictions	

16.6.10 RC Remote Calls Enabled

States whether remote calls are enabled on a line, i.e., if NET can be called from the stations connected to the line in question. The attribute applies to lines with autocaller (AC = 1).

When a station using RP570 protocol has called NET, it informs NET which station to poll by sending a PRI (Poll Request Instruction). After that, the line works as an ordinary modem line.

Data type:	Integer	
Value:	0	Remote calls not enabled
	1	Remote calls enabled
Default value:	0	
Access:	No restrictions	

16.6.11 RW Radio Connection Wait Time

Normally the DCD (Data Carrier Detect) signal is used to indicate an active connection. There are cases, however, e.g. on radio telephone lines using half duplex links, where this is not possible. The RW attribute gives the amount of seconds to wait in such a situation, from the finishing of the dialling until the transmission is started.

Data type:	Integer
Value:	0 ... 32767
Unit:	Seconds
Access:	No restrictions

16.6.12 SR ACE AT S Register

The S registers used by autocallers following the AT (Hayes) de facto standard.

All autocallers using the AT command set have a number of S registers. The number of registers used and the meaning of the individual registers slightly varies 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 registers number 2, 6, 7, 8, 9, 10, 11 and 12 are accessed. By using the MC attribute (see above) also other S registers can be accessed.

The S registers 11 and 12 cannot be set.

Values:	See the autocaller manuals
Indexing:	100 * line number + register number

Example:

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

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

16.7 LON configuration attributes

16.7.1 NC Network Variable Configuration

Reading and writing of the network variable indices. Each LonTalk protocol line acts as an interface to the LONWORKS device bus. It is possible to read the Network Variable (NV) indices 0 ... 4095 to other entities on the LONWORKS network. External tools can configure each of these indices in the same way as for other LONMARK™ [1] devices. Using the NC attribute, it is possible to read and write the network variable indices 0 ... 4095. Writing to the NC attribute is equivalent to issuing corresponding network management command directly from a LONWORKS network configuration tool.

The attribute is used indexed with a code calculated from line number and network variable index. The corresponding SPA point or LMK point (see Section 17) must exist before a network variable index can be configured.

Data type:	Vector		
Value:	Each index is a vector with 4 integer elements in the range ...255. The vector elements are defined as follows:		
	element	1:	p | d | nv_selector_high
		2:	nv_selector_low
		3:	t | st | a | addr_index
		4:	ext_addr_index
	where		
	‘p’	Network variable priority, 1 bit	
	‘d’	Direction, 1 bit value 0 for IN and 1 for OUT	
	‘nv_selector_high’	Nv selector, 6 msb bits	
	‘nv_selector_low’	Nv selector, 8 lsb bits	
	‘t’	Turnaround, 1 bi	
	‘st’	Service type, 2 bits	
		0 = Ackd, acknowledged	
		1 = Unackd_repeated, Unacknowledged/repeated	
		2 = Unackd, unacknowledged	
	‘a’	Authentication, 1 bit	

Table continues on next page

[1] LONMARKis a trademark of Echelon Corporation.

	'addr_index'	Index to address table, 4 bits.
		The value should be 0x0F
	'ext_addr_index'	Index to address table, 4 bits.
		The value should be 0x0F
Indexing:	(4096 * NET line number) + network variable index	
Access:	No restrictions	

More information can be found in the manual Connecting LONWORKS Devices to SYS600.

Example:

Reading the network variable configuration for NV-index 1 on line 1 (more information is found in the Neuronchip Data Book):

```
@NV_IX_CFG = NET1:SNC4097
```

Writing the configuration back:

```
#SET NET1:SNC4097 = %NV_IX_CFG
```

16.7.2 XA Extended Address Table

Reading and writing of the LONWORKS device address table configuration. The attribute is indexed with a code calculated from NET line number and address index. Using the XA attribute, it is possible to read and write the extended address table indices 0 ... 255 (0 ... 14 might be reserved by the processor and therefore not recommended to be used).

Data type:	Vector			
Value:	Each index is a vector with 5 integer elements in the range 0...255. The elements are defined as follows:			
	Element	1:	type	
		2:	d	node_or_member
		3:	rpt_timer	retry
		4:	rcv_timer	tx_timer
		5:	subnet_or_group	

where	
'type'	Type of address entry or 80H + group size
	Type of address:
	0 = Unbound address table entry (if element 2 = 0)
	0 = Turnaround address (if element 2 = 1)
	1 = Subnet or node address
	3 = Broadcast address
	80H...8FH = group address. The bits 0...6 specify the size of the group.
'd'	Domain index, 1 bit
'node_or_member'	Node number or member of a group, 7 bits
'rpt_timer'	Time between repetitions in unack_repeated service, 4 bits. The values of timers are coded, see the reference table in the Neuronchip Data Book.

Table continues on next page

	'retry'	Retry count, 4 bits
	'rcv_timer'	Receive timer for multicast (group) messages, 4 bits. The values of timers are coded, see the reference table in the Neuronchip Data Book.
	'tx_timer'	Transmit timeout for acked or request /response service, 4 bits. The values of timers are coded, see the reference table in the Neuronchip Data Book.
	'subnet_or_group'	Subnet number or group number
Indexing:	(4096 * NET line number) + address index	
Access:	No restrictions	

Example:

Reading the address table entry information from address table index 2 of NET line 1 (more information is found in the Neuronchip Data Book):

```
@ADDR_TBL_ENTRY = NET1:SXA4098
```

16.8 Miscellaneous NET line attributes

16.8.1 Diagnostic counter

This attribute applies to all protocols. For other protocols but ANSI X3.28 Half duplex / Full duplex, SPA and RP570/RP571, see protocol specific manual for information.

16.8.1.1 DC Diagnostic Counters

The line protocols gather statistical information about the events on the lines by incrementing a number of diagnostic counters. All major events and error situations of the communication have their own counters.

Each line has a number of diagnostic counters numbered 1 ... 16 (or 32 if the line contains autodialing). The meaning of the individual counters in number order is:

1	TRANSMITTED MESSAGES/TELEGRAMS This counter is incremented whenever a message is transmitted successfully. On ANSI X3.28 lines, a successful transmission includes the reception of a positive acknowledgement (ACK). On Half Duplex lines, the counter is not incremented by polling packets, only by messages (commands or replies).
2	FAILED TRANSMISSIONS The counter is incremented when a message transmission fails. On an ANSI line, the transmission has failed if no positive acknowledgement (ACK) is received in spite of retrials. The counter is also incremented if the states of the modem signals CTS and DCD prevent transmission. For the special DCD link type LK = 8, the counter is updated after each retry while waiting for passive DCD.
3	TRANSMITTED TIMEOUTS The counter is incremented each time a timeout occurs during the waiting for a response. If for example 3 time-outs occur at the transmission of a message (with retrials), the counter is incremented 3 times. When the retry limit is reached, finally also the counter 2 is incremented once.
4	TRANSMITTED ACKS/FETCH Is incremented on ANSI X3.28 lines each time a positive acknowledgement (ACK) is transmitted. Not used on RP570 lines.

Table continues on next page

5	TRANSMITTED NAKS/POLLS Is incremented on ANSI X3.28 lines each time when a negative acknowledgement (NAK) is transmitted. Not used on RP570 lines. UNPROCESSED LON MESSAGES The amount of messages read from the driver but not yet processed. Used in LON line only.
6	TRANSMITTED ENQS/BROADCAST Is incremented on ANSI X3.28 lines each time when an enquiry (ENQ) is transmitted. Not used on RP570 lines.
7	RECEIVED ACKS RECEIVED EVENT TELEGRAMS (SPA lines) RECEIVED XONS (printer lines) Is incremented on ANSI X3.28 lines each time a positive acknowledgement is received from the line. Not used on RP570 and P214 lines.
8	RECEIVED NAKS RECEIVED DATA TELEGRAMS (SPA lines) RECEIVED XOFFS (printer lines) Is incremented on ANSI X3.28 lines each time when a negative acknowledgement (NAK) is received from the line. Not used on RP570 and P214 lines.
9	RECEIVED ENQS/TIMEOUTS (P214) Is incremented on ANSI X3.28 lines each time when an enquiry (ENQ) is received from the line. Not used on RP570 lines.
10	RECEIVED EOTS (ANSI X3.28 lines) APPLICATION FAILURE (RP570 lines) APPLICATION CONNECTION TIMEOUTS (P214 lines) Is incremented on ANSI X3.28 lines each time when an end-of-transmission (EOT) is received from the line.
11	RECEIVED MESSAGES/TELEGRAMS Is incremented each time a message has been received from the line without errors.
12	PARITY ERRORS Is incremented when a received message is rejected because of a parity error. TCP CONNECT COUNT (IEC60870-5-104 master)
13	OVERRUN ERRORS Is incremented when a received message is rejected because of an overrun error. TCP ACCEPT COUNT (IEC60870-5-104 slave)
14	CHECK SUM ERRORS (ANSI lines) REDUNDANCY ERRORS (P214, SPA and RP570 lines, LON) Is incremented when a received message is rejected because of a discrepancy in the checksum (BCC or CRC). TCP CLOSE COUNT (IEC60870-5-104 master and slave)
15	FRAMING ERRORS (ANSI, P214, SPA, RP570, LON) Is incremented when a received message is rejected because of a framing error.
16	BUFFER OVERFLOW ERRORS Is incremented when a received message is longer than 259 bytes and therefore does not fit into the message buffer.
17	ACE_CONNECTIONS_ LOCAL_ORIGIN Calls initiated locally.
18	ACE_CONNECTIONS_REMOTE_ORIGIN Calls initiated remotely.
19	ACE_RECEIVED_ERROR_RESPONSES

Table continues on next page

20	ACE_RECEIVED_NO_CARRIER_RESPONSES TCP CONNECT COUNT (SPA-TCP, DNP V3.00/LAN master, Modbus TCP)
21	ACE_TIMEOUTS TCP ACCEPT COUNT (DNP V3.00/LAN slave)
22	ACE_FAILED_DIALINGS TCP CLOSE COUNT (SPA-TCP, DNP V3.00/LAN master and slave, Modbus TCP)
23	ACE_FAILED_CONNECTINGS_OF_REMOTE_CALLS
24	ACE_FAILED_DISCONNECTIONS
25	ACE_IGNORED_RINGS
26	ACE_RECEIVED_RINGS

None of the line protocols updates all of the counters. See protocol specific manuals for details. The following counters are updated for the ANSI, SPA, LON and printer lines:

ANSI X3.28 Full Duplex: nr 1 - 9 and 11 -16

ANSI X3.28 Half Duplex: nr 1 - 5, 7 , 10 - 16

ASCII printer protocol: nr 1 and 3

SPA protocol: 1, 2, 3, 11, 12, 13, 15 and 16

LON: 1, 2, 3, 11, 13, 14, 15

Data type:	Integer or vector
Value:	0 ... 30000 (modulo 30001)
	When the value 30000 is reached, the following counter values are 0, 1, 2, etc. When the IU attribute is changed from 0 to 1, the NET program resets the counters of the line.
Indexing:	When accessing diagnostic counters, the attribute is indexed according to the formula.
	100 * (line number) + (diagnostic counter number)
Access:	Read-only, the values can be reset

Example:

The diagnostic counters 1 ... 16 of line 3 in NET1 are displayed in the window COUNTER:

```
!SHOW  COUNTER  NET1:SDC(301..316)
```

The counters 1 ... 32 of line 1 are reset.

```
#SET  NET1:SDC(101..132) = 0
```

16.8.2 Clock synchronization attributes

16.8.2.1 LK Link Type

The clock synchronization of LONWORKS lines. The lowest 3 bits in the attribute specifies clock synchronization functionality as follows. Note that time synchronization cannot be used for several base systems, which are connected to the same device. This creates erroneous time stamps.

0	No clock sync
1	Send LSG clock sync (for the relays that utilise nv warning and nv clock telegrams)
2	Send minute pulse (for the relays that utilise nv time telegram)
3	Send LSG and minute pulse
4	Receive LSG clock sync
5	Receive the minute pulse. It is recommended to use minute only when the other synchronization methods do not work, or when the exact time is not needed because of the inaccuracy on high channel load on LON line with minute pulse.
6	Send SLCM Reference Time

On an IEC 60870 line this attribute controls the behavior of RTS-control line:

12	RTS always on, full duplex (balanced slave default)
13	RTS / CTS controlling also with balanced mode

On DNP V3.00 protocol:

14	Collision detection in use, transmission when the Data Carrier, Detect signal of the line is not set
15	No collision detection, Data Carrier Detect signal is handled as in other protocols

Data type:	Integer
Values:	0 ... 15
Default:	0 (3 recommended for LONWORKS lines)
	15 for DNP V3.00 protocol
Indexing:	NET line number
Access:	Read, conditional write

16.8.2.2 SF Sync Format

Time sync message format for the clock synchronization reception. This attribute has a meaning only on General ASCII lines with PM = 5 or 6.

Data type:	Integer	
Value:	1 ... 6:	
	1	COMPUTIME
	2	RCC8000
	3	Chinese TV clock, BLF format
	4	MAC02
	5	Meinberg GPS 166
	6	TAIP (when TAIP format is used PM must be = 6)
Default value:	1	
Access:	Read, conditional write	

1.8.2.3 SS Sync Status

The status of the clock synchronization on the line. This attribute has a meaning only for General ASCII lines with PM = 5 or 6.

Data type:	Integer	
Value:	Integer where the bits has the following meanings:	
	Bit	Meaning if bit value = 1
	0	Not used
	1	Summer time (Valid only with GPS166 format
	2	Not used
	3	Summer Time Change (Valid only with GPS166 format)
	4	Not used
	5	Not used
	6	Not used
	7	Time invalid, synchronization not done
Access:	Read-only	

Example:

Checking if NET1 is synchronized via line 2 (Bit 7 of the SS attribute is checked):

```
#IF BIT(NET1:SSS2,7)==1 #THEN #BLOCK
........................;Clock not synchronized
#BLOCK_END
#ELSE  #BLOCK
........................;Clock synchronized
#BLOCK_END
```

Section 17 STA objects for communication system

17.1 About this section

This section describes the STA objects and their attributes. The section is divided into nine sections as follows:

Section 17.2	General: The station types, the definition of STA objects, the object notation.
Section 17.3	Common STA attributes: This section describes in details the attributes that are common to all station types: Basic Attributes (IU, LI), Device Reservation Attributes (AL, AS), System Message Handling Attributes (MI, MS, OS, SE).
Section 17.4	STA attributes, ANSI stations: Basic Attributes (PH, SA, ST), Polling Attributes (CP, PA), Suspension Attributes (FS, RT, SU), Diagnostic and Counter Attributes (CT, DC, DE, DI, DS, LS), Station Communication Parameters (EN, NA, TI), Memory Area Definitions (AD, AT, BF, CO, DT, LE, MC, MR, TS), Message SPLIT (SL, SP), Memory Access (ME), Time Synchronization (SY).
Section 17.5	STA attributes, S.P.I.D.E.R. and collector RTUs: Basic Attributes (HR, SA), Diagnostic Counters (DC), RTU Configuration Attribute (FC, FT, SY), Process Communication (DA, RD, RT, SC, SM, TA, TD), Terminal Reports (TE, TM, TS), Communication Loop Attributes (DR, LS, LU, LW).
Section 17.6	STA attributes, SPACOM: Basic Attributes (BL, SA, UN, UT, RL, EC, EL, EP), Diagnostic Counters (DC), Station Suspension Attribute (RT), SPA Point Definitions (ED, SP), Miscellaneous SPACOM STA Attributes (DA, PR, SM, ST, UP), SPACOM attributes for TCP interface (CT, ET, IA).
Section 17.7	STA Objects, P214 RTUs: Basic Attributes (IU, LI, SA), Device Reservation (AL, AS), Suspension and Diagnostics (DC, RT), System Message Handling (MI, MS, OS), Data Communication (DA, EC, FC, FE, GP, NR, TV), Priority Control (PC, PM)
Section 17.8	STA attributes, REX stations: Basic Attributes (NN, SN, UN, UT), Session Handling (SC, SH, SI, SK, SR, SS), Process Communication (DA, RQ, SM, TC, TQ, GI, GO, IL), Event Handling (EF, HI, HS, RM), Suspension Attributes (RT), SPA Point Definition (SP), File Transfer Handling (FO, FP).
Section 17.9	STA attributes, LMK stations: Basic Definition Attributes (NN, SN, UT), Polling Attribute (CT), Process Communication (DA, GI, LM, RT), Diagnostic Attributes (DC, DI), LON Point Definition (LP).
Section 17.10	STA attributes, SPI stations: Basic Attributes (SA), Configuration Attributes (FT, FV, OL, RM), Process Data Communication Attributes (AV, DD, EI, EX, ID, PC, TA, TR). Process Data Communication attributes apply also to Modbus Slave station, Function Control Attributes (CB, CT, DC, DI, EC, MM, RT, TI), Terminal Messages (ST, TV), Loop Control Attributes (LC, LT), Redundant Line Attributes for RP-570 Slave Protocols (LI, RU), Application Based Command Controlling Attributes (AT, CS).

17.2 General

17.2.1 Station types

All the station types that can be connected to NET unit were listed in Section 15. If not otherwise mentioned, the station type can be connected only to PC-NET unit. This section describes the attributes of the following station types:

- STA stations: the process units that communicate with NET unit using the ANSI X3.28 protocols, for example, Allen-Bradley PLCs and SRIO.
- RTU stations: Relays and control devices and NCC connections, which communicate by using the RP 570 master protocol. S.P.I.D.E.R. RTUs and Collector 100 and 300.
- SPA stations: bay control units, mainly SPACOM relay units, connected via the SPA protocol or via the LonTalk protocol and LSG devices. The SPA stations connected via a LSG device are configured as SPA units connected via the SPA protocol, except for two attributes. These attributes are UT attribute and the RL attribute. There are also some differences in the SPA point definition attributes.
- REX stations: REx type relays (REF, REC, RED, REL, etc.) communicating with the base system through a LONWORKS line and the PC-NET unit.
- LMK stations: LSG devices and other devices connected to the LONWORKS network through a standard LONWORKS interface (for example Weidmüller).
- SPI type stations: SCADA systems and other control systems, which communicate with NET unit through the RP570 slave protocol in a master-slave relation where NET unit is the slave.
- Modbus Slave station: SCADA systems and other control systems, which communicate with CPI-NET unit through the Modbus Slave protocol in a master-slave relation where the CPI-NET unit is the slave. Modbus Slave station can only be connected to CPI-NET unit.
- IEC type stations: Relays and control devices and NCC connections, which communicate by using the IEC 60870-5-101, -103 and -104 Protocols.
- DNP type stations: Relays and control devices and NCC connections, which communicate by using DNP 3.0 protocols.
- PLC stations: Relays and control devices, which communicate by using Modbus Master protocols, Modbus RTU, Modbus ASCII and Modbus TCP.

The attributes of other station types are described in separate documents.

17.2.2 Definition

Each station must be defined as a STA object in the NET unit to which it is directly connected. The STA object can be defined with System Configuration Tool or on-line with the NET station definition attributes described in Section 15. ANSI stations are defined with the ST attribute, RTUs with the RT attribute, SPA stations with the SP attribute, REX stations with the RX attribute, and LMK stations with the LM attribute. Station type PLC, IEC and DNP are created using line attribute DV.

When the STA objects are defined on-line, the STA attributes get the default values given in the attribute descriptions. Separate attribute writes is needed to complete the configuration.

17.2.3 Broadcast stations

Each time a NET unit is started, it creates automatically four STA objects with system object number 0. One STA object is of type RTU (S.P.I.D.E.R. RTUs and Collector), one is of type SPA (SPACOM), one is of type LCU and one is of type PLC. These STA objects are broadcast objects. For stations of type RTU, the broadcast object means all stations of this type connected to the same NET unit. For stations of type SPA, the broadcast object means all stations connected to chosen NET lines (see the BL attribute in Section 17.6). Provided that the broadcast stations have been mapped for the application, they can be accessed from SCIL.

17.2.4 Object notation

The STA attributes are accessed from SCIL with the object notation:

STAn:Sat

where

'n'	The logical station number, 0 ... 5000, as known to the application where the object notation is used. 'n' is translated to the communication system object number (0 or 1 ... 255) as described in Section 7
'at'	The attribute name

17.3 Common STA attributes

The attributes that are described in this section apply to all station types.

17.3.1 Basic attributes

17.3.1.1 IU In Use

The operational status of the station, whether it is in use or out of use. Taking the station out of use with this attribute stops all data communication with the station. All operations that would result in a data exchange are disabled. The station itself is not affected by the attribute, only NET unit's image of the station.

The station causes no system messages as long as it is out of use, only at the moment when it is taken out of use. Likewise, taking a station into use causes NET unit to send a system message (see the SE attribute).

Setting IU to 1 is allowed only if the station address is legal (the SA attribute) and the device is allocated by some application (the AL and AS attributes).

Regarding S.P.I.D.E.R. and Collector RTUs, NET unit sends an SCI (Status Check Instruction) to the station when it is taken into use by setting the IU attribute to 1. This is done unless the SC attribute has been set to 0 manually while the station was not in use. In this case, the polling will proceed from the state where the station was left when taken out of use.

Regarding REX and LMK type stations, the station objects should be taken into use not until the line object of the stations has been set to IU=1. Correspondingly, all station objects connected to a LonTalk protocol line should be taken out of use before the line object is taken out of use.

Data type:	Integer	
Values:	0	Out of use
	1	In use
Default value:	0	
Access:	No restrictions	

17.3.1.2 LI Line Number

The number of the NET line, to which the station is connected.

By writing a new value to the LI attribute, the station can be switched from one line to another. Both lines need to be defined with the same protocol and their original and destination lines have to be taken out of use. Changing the LI attribute, that is moving a station from one line to another, demands that the station as well as the old and new lines have been taken out of use (IU = 0).

This attribute is also used for setting the number of the back-up line if redundant lines are used. The indexes are used only when the redundant lines are used. Note that the indexes 1 and 2, i.e. main and back-up line numbers, are switched when a line switch operation is executed. The number of the back-up line is set to index 2 of the LI attribute.

Data type:	Integer
Value:	NET line number 1 ... 12
Default value:	The line number given when creating the STA object, see Section 15. For example, the default value is 1 for DNP V3.00.
Indexing:	None if redundant lines are not used
	Index 1 is for the number of the main line
	Index 2 is for the number of the back-up line
Access:	Read, conditional write

17.3.2 Device reservation

17.3.2.1 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 will be sent to this application.

The stations address all their messages to one single station address, which is the address of the communication unit. The NET unit forwards the received messages to the application, which has reserved the station (the AS attribute).

Although one application has reserved a station, other applications can send read commands to the station. The station will not transmit spontaneous messages to these other applications, unless the message split feature of NET unit is used. See the SP attribute in Section 17.4.

Data type:	Integer
Values:	0 or 1
Suggested value:	The AL attribute must always be = 1 when the station is in use
Access:	No restrictions

Example:

```
#SET STA1:SAL=1
```

17.3.2.2 AS Allocating Application

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

When AL is set to 1 on line, AS is automatically set to the number of the application from which AL is set. When AL is set to 0, AS also is automatically assigned the value 0.

The allocating application will receive all the spontaneous process data from the station (if message split, Section 17.4, is used, also other applications will receive the messages).

Data type:	Integer
Value:	0 ... 250. The application number as known to the communication unit. 0 = No application.
Access:	Read-only

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

17.3.3 System message handling

The attributes of this section affect the transmission of system messages that NET unit sends on various events related to the STA objects. The system messages are sent to the applications defined by the MS attribute and updated in the process objects defined by the MI attribute. Via the process objects event channels, loggings, alarms, etc. may be activated automatically.

Based on system messages from STA devices, the SYS600 base system automatically updates the validity stamp of the object values in the process database (the OS attribute). See the System Message Attributes in Section 15. When a system message of type "not valid" is received from a STA object, the main program automatically marks all process objects related to that station as not valid. The marking is done by setting the OS attribute (Object Status) to 2 (OBSOLETE_STATUS). The process objects, whose UN attribute (Unit Number) corresponds to the station in question, are to be marked. A system message from the same station, which tells that the connection is OK again, does not lead to any process object marking. The updated object values are subsequently marked valid (OS = 0).

Refer to Section 15 and the System Configuration manual to learn more about the system message handling.

The system message generation of the RTU type stations (S.P.I.D.E.R. RTU and Collector) differs from the system message handling of other stations.

Stations of other types than RTU cause the generation of system messages on the following events:

- The station is put into suspended state because it does not respond to poll packets or messages, or because IU has been set to 0.
- The connection to a station has been lost or re-established.
- The station connection recovers after a disturbance.

The S.P.I.D.E.R. RTUs generate system messages in the following situations:

- When the station is suspended or recovers from suspension.
- The station is stopped/restarted.
- Terminal message received.
- Terminal status received.
- Terminal event received.

Due to the differences in the generation of system messages, the MI attribute of the RTU type stations differs from the MI attribute of other stations. It is therefore described separately below.

17.3.3.1 MI Message Identification (other station types than RTU)

The message address used in system messages. The MI attribute is the address of the process objects (the OA attribute) where the system messages from the device are updated. At the generation of a system message the status code of the message is updated in the OV attribute of the process object with this object address.

The system message status code is stored in the process database of the receiving application or applications (defined by the MS attribute). If the NET node attribute SE (System messages enabled) has a value 4, a binary process object is updated at the same time as the analog process object defined with MI. See the MI attribute in Section 15 for more information.

Data type:	Integer
Value:	1 ... 65535, process object addresses
Default value:	1000 + STA object number (for STA, SPA, REX, LMK, PCL, SPI stations)
	1000* type code + STA number for exchangeable device types (e.g. IEC, DNP, PLC, ATR) See protocol dependent configuration manuals for more information.
Example:	See the MI attribute in Section 15
Access:	No restrictions

17.3.3.2 MI Message Identification, RTUs

The attribute has the same meaning as described above for other station types, but there may be several receiving process objects, one per message type.

The S.P.I.D.E.R. RTUs may cause four different types of system messages with different origins. The message types are numbered 1 ... 4 as follows:

1. NET internal messages. Codes 12601 ... 12699.
2. RTU terminal status message. The RTUs send status messages when there is a change in the RTU status. Codes 12701 ... 12749. See the TS attribute.
3. RTU terminal event message. The terminal events from the RTU are given a tag number 0 ... 999 in the communication unit. The tag number is sent as a system message to the base system. See the TE attribute.
4. RTU, terminal message (system message in RTU). The terminal messages from the RTU are given a tag number 0 ... 999 in the communication unit. The tag number is sent as a system message to the base system. See the TM attribute.

Data type:	Vector
Value:	Vector of four integers 1 ... 16380. The object addresses of the receiving objects.
Indexing:	System message type number. No index = index 1.
Default values:	Index 1: 8000 + station number
	Index 2: 8500 + station number
	Index 3: 9000 + station number
	Index 4: 9500 + station number
Access:	No restrictions

17.3.3.3 MS Message Application

Specification of the application or applications that will receive the system messages caused by the station. Each station may have up to six applications that receive system messages. The APL system object numbers as defined in NET unit specifies the application.

Data type:	Integer or vector
Value:	1 ... 250, or vector of six integers in the range 1 ... 250. APL system object numbers as defined to NET unit.
Indexing:	1 ... 6
	No index = Index 1
Access:	No restrictions

It is possible to send system messages to more than one application with a STA object. The System Configuration Tool does not support this, instead you have to create a user defined program if you need to do it.

17.3.3.4 OS Object Status

Indicates the state of the station. Writing to the OS attribute (OS = 1) of a station makes NET unit re-transmit the last system message caused by the station. Possible "Stopped" and "Suspended" messages cause old marking of process objects. By reading the OS attribute, the status code of the system message can be read. The attribute is available for Master and Slave for IEC stations.

Data type:	Integer	
Value:	When written:	
	1	Re-transmit system message
	When read:	
	A status code, for example:	
	0	OK (communication works properly)
	12337	ANSI station suspended
	12339	ANSI station taken out of use
	12602	RTU suspended
	12604	RTU taken out of use
	13251	SPA station suspended
	13252	SPA station taken out of use
	13801	PLC station suspended
	13802	PLC station taken out of use
	Etc., see the Status Codes manual	
Default value:	Suspended for IEC Stations	
Access:	No restrictions	

17.3.3.5 SE System Messages Enabled

Specifies whether system messages generated by NET unit and related to the station are sent to applications or not. Using this attribute, it is possible to disable the system messages related to the station. The attribute does not affect messages generated in the stations (terminal messages in S.P.I.D.E.R. RTUs).

Data type:	Integer	
Values:	0	System message generation disabled
	1	System message generation enabled (normal value)
Default value:	1	
Access:	No restrictions	

The value SE = 0 should be used only in special cases, for example if the base system application program often executes commands, which cause undesirable system messages. Undesirable system messages can be regular stopping and starting of a station.

17.4 STA attributes, ANSI stations

Besides the common attributes described in Section 17.3, the STA objects of type ANSI stations (STA) have the attributes described in this section.

17.4.1 Basic attributes

17.4.1.1 PH Phone Number

The phone number of the station.

The PH attribute is not directly used for dialling, but it can function as a memory for the phone number to be used when calling a station.

Data type:	Text
Value:	Text of max. 25 characters. The character string may contain a leading 0, "-", "/" characters, etc. Usually a delay in the dialling is marked by a comma (",").
Access:	No restrictions

Example:

```
#SET NET1:SCN3 = STA4:SPH
```

17.4.1.2 SA Station Address

The station address of the ANSI station. The value of this attribute must be the same as the corresponding station address value defined in the station.

Each station connected to a NET unit through the ANSI X3.28 protocol must have a unique station address. This demand for uniqueness also comprises the NET unit itself, see the SX attribute in Section 15. However, stations connected to separate NET units may have the same station addresses. When the station address is written on-line with SCIL, the communication program checks that the uniqueness is maintained.

Data type:	Integer
Value:	1...255
Access:	Read, conditional write

Example:

STA1 is given the address 20:

```
#SET STA1:SSA = 20
```

17.4.1.3 ST Station Type

The type of the ANSI station: SLC-500 or other types. The type specification is needed because the interpretation of object addresses in messages from SCL-500 differs from the other ANSI station types.

Data type:	Integer	
Value:	1	SRIO and other ANSI station types, except SLC-500
	4	SLC-500
Default value:	1	
Access:	No restrictions	

17.4.2 Polling attributes

The attributes in this section apply exclusively to stations on ANSI half-duplex lines.

17.4.2.1 CP Command Poll Count

The number of commands polled from one station until the next station is polled. The attribute is significant only to stations, which are connected to multidrop lines and which transmit command messages spontaneously.

The CP attribute states how many commands the station is allowed to transmit in succession, before the next station is polled for commands. The commands from applications have higher priority than the polling of stations.

CP gives a possibility to optimise the multidrop line. It is also a way to assign different priority levels to the stations for heavy-load situations.

Data type:	Integer
Value:	1 ... 255. Number of transmitted commands
Unit:	Transmitted commands
Access:	No restrictions

A high value may slow down the communication to other stations, for example in hardware fault situations.

17.4.2.2 PA Polling Address

Using this attribute, a station can be polled by using another address than its own station address (the SA attribute). The PA attribute makes it possible to connect PLC-5 and stations using Data Highway to a multidrop line of NET unit. In all other cases, the PA attribute should be equal with the SA attribute.

Data type:	Integer
Value:	1 ... 63 or 72 ... 254
Default value:	For stations defined off-line: the SA attribute
	For stations defined on-line: no default value
Access:	No restrictions

17.4.3 Station suspension

Suspension of a station means that the NET unit notices that the communication to the station does not work, and gives this information to an application as a system message. All process object values related to that station are marked as outdated, as they apparently are not properly updated. The reasons for a suspension may be:

- A reply message from a station does not arrive in time (REPLY TIMEOUT), see the RT attribute.
- A station on a multidrop line does not respond to polling packets (DEV STATUS IN signal generated in NET unit).
- A reply message from a station contains a severe error code.
- No acknowledgement (ACK) to a reply message.

Table 7 shows an overview of the reasons for suspension and the states that cause recovery from the suspension.

Table 7: Reasons for station suspension and recovery

Suspension Reason	Recovery Reason
No ACK to replay	ACK to command or reply
No replay in time	Command from substation or Reply in time
Severe error status in reply	Command from substation or Reply in time
No poll response in time	Response to poll (EOT or message)

17.4.3.1 FS Fast Select During Suspension

Determining which kind of commands from the application will be forwarded to a suspended station.

Data type:	Integer	
Value:	0	No read or write commands are forwarded to a suspended process unit
	1	Write commands will be forwarded to a suspended process unit
	2	Read and write commands will be forwarded to a suspended process unit
	3	Read allowed to suspended station
Default value:	0	
Suggested value:	If there is a great probability for data communication disturbances (e.g. at system installation or addition of new stations) the value 2 is recommended for the FS attribute.	
	In normal situations non-zero FS values are recommended mainly for multidrop connected stations, which do not transmit spontaneous command messages to the application. Such a station may be suspended because of a reply time out, although it responds to polling packets all the time. It can get out of the suspended state only if some kind of command message is transmitted to it.	
Access:	No restrictions	

17.4.3.2 RT Reply Time-out

The maximum time the NET unit will wait for a reply from the station.

Data type:	Integer
Value:	0 ... 655
Unit:	Seconds
Default value:	45
Suggested value:	For point-to-point connected stations, the recommended time is about 15 s. For multidrop connected stations the recommended time is about 45 s. To prevent unnecessary timeouts caused by data link layer re-transmissions, the value of RT should be significantly greater than the product of the TI and EN line attribute values (TI*EN) (Section 14).
Access:	No restrictions

17.4.3.3 SU Suspension Time

The time between diagnostic commands to a suspended station.

Unlike the DI attribute (see Section 14), this attribute is used when the station is in a suspended state. If DE = 0, SU has no meaning. If an acceptable reply is received to the diagnostic command, the station returns to the normal state, otherwise it stays in the suspended state for another SU period. If any other message is received from the station without errors during the period, the station enters the normal state.

Data type:	Integer
Value:	0 ... 655
Unit:	Seconds
Default value:	60
Suggested value:	60 ... 600 (the same as for the DI attribute)
Access:	No restrictions

17.4.4 Diagnostic and counter attributes

17.4.4.1 CT Counters and Timers

The values of the counters and timers situated in the station. Most station types have a number of diagnostic counters and timers. The CT attribute is used to read and reset these counters and timers. The exact number and format of diagnostic counters and timers may vary depending on the station type.

When reading the CT attribute, the word addresses of the counters are given as indices. The first counter address is obtained from the station by reading the DS attribute (see below).

The counters and timers are reset by a #SET command, by which the value 0 is assigned to the CT attribute. In the reset command the indices have no meaning. The reset command always concerns all the counters and timers.

Data type:	Integer, vector
Value:	Read: Vector of integers. The length of the vector and the value range depend on the station type Write: 0

Table continues on next page

Indexing:	Read: Word addresses of the counters to be read.
	Write: No index
Access:	No restrictions

17.4.4.2 DC Diagnostic Counters

The values of the diagnostic counters which NET unit keeps for the station.

To make the supervision and testing of the station communication easier, NET unit holds five diagnostic counters (numbered 1 ... 5) for each station. Each counter monitors a certain kind of events, according to the following list:

1. STATION SUSPENSION
   The counter is incremented each time the station is suspended. Depending on the reason for the suspension, one of the other counters is also incremented at the same time.
2. DEV STATUS RECEIVED
   The counter is incremented when a DEV STATUS IN signal is received from the ANSI X3.28 Half Duplex protocol data link layer. This signal indicates either that a station has ceased to respond to polling packets, or that it has started to respond again after a disturbance. The first situation will lead to a suspension of the station and increment counter 1 as well. From the function of this counter follows that generally an odd value indicates that the station does not respond to polling packets at present.
3. REPLY TIMEOUTS
   This counter is incremented when a command has been transmitted to a station and no reply arrives from the station within the time limit specified by the value of the RT attribute. The station is suspended.
4. STS NOT OK FROM RTU
   This counter is incremented each time a reply message from the station contains a non-zero value in the status code byte (STS). This does not necessarily lead to a suspension of the station. Suspension may occur depending on the severity of the status transmission responses are transmitted.
5. STS NOT OK FROM NET
   This counter is incremented when the NET unit transmits a reply message with a non-zero status code value (STS). Usually, the error code is caused by missing definitions in the NET unit or by the contents of the command message from the station, e.g. an unknown destination device, an undefined memory address or a memory area defined with a wrong data type or coding.

Data type:	Vector
Value:	Vector of integers in the range 0 ... 30000. Each element is the value of a counter. When the value 30000 is reached, the counters start over from 0, that is the following counter values will be 0, 1, 2 ... etc.
Default value:	All elements = 0
Indexing:	The DC attribute is indexed with counter number, 1 ... 5
Access:	Read-only, the values can be reset

17.4.4.3 DE Diagnostics Enable

Indicates whether the NET unit will transmit diagnostic commands to the station cyclically, and what type of diagnostic commands that will be used.

The time cycle always starts from zero when a message (command or reply) is received from the station. The length of the time cycle (in seconds) is normally the value of the DI attribute. When the station is suspended, the cycle length is obtained from the SU attribute.

The diagnostic commands used are Diagnostic Status and Diagnostic Loop. The Diagnostic Status command reads status information (see the DS attribute below) from the NET unit of the station. The status information received tells among others the type and operating mode of the station, and also gives an indication of possible errors. Diagnostic Loop transmits a byte sequence to the station, and only checks that the same byte sequence is received in the reply message.

If both commands are implemented into the station, Diagnostic Status is normally the command to use, but some station types have only the Diagnostic Loop.

If an RTU does not transmit messages spontaneously, it might stay in the suspended state although it answers to the polling packets. This situation can occur if the polling packets and their end-of-transmission responses are transmitted correctly, but line disturbances prevent the correct transmission of a whole message. After such a disturbance the station can return to the normal state if:

a) It transmits a command message as response to a polling packet

b) A diagnostic command is transmitted to the station

c) The value of the FS attribute is non zero, so that the commands from the application system are forwarded to the suspended station.

Data type:	Integer	
Value:	0	No diagnostic commands transmitted cyclically
	1	Diagnostic Status
	2	Diagnostic Loop
Default value:	0	
Suggested value:	On point-to-point lines, diagnostic commands should always be used. On multidrop lines, the use of Diagnostic Status can always be recommended if the extra load on the communication line is not a problem. The Diagnostic Loop command is not always to be recommended, because the station connection is usually tested anyway by the polling packets, and Diagnostic Loop gives no additional information. If the stations of a line are not polled cyclically, Diagnostic Loop can also be recommended.	
Access:	No restrictions	

17.4.4.4 DI Diagnostic Interval

The time between diagnostic commands to a station, which is not in suspended state (cp. the SU attribute, Section 14).

The DI attribute is meaningful only if DE has a non-zero value. Then DI gives the time in seconds from the last message (command or reply) reception from the station until the next diagnostic command will be sent. If some other message is received from the station during that time, the timer restarts from zero.

Data type:	Integer
Value:	0 ... 655
Unit:	Seconds
Default value:	60
Suggested value:	A typical value for a point-to-point connected station is 60 s. If the diagnostic commands are used on a multidrop line, the interval will normally be longer because of the low transmission rate. For the stations on the same multidrop line, the DI values should slightly differ from each other to smooth the load on the line especially after the communication program has been restarted.
Access:	No restrictions

17.4.4.5 DS Diagnostic Status

The status code of 10 bytes. A read command using the DS attribute returns 10 bytes of status information from the station or its communication unit. The format and exactitude of the status information depend on the station type. Usually information about station type, NET unit type, operating mode, error bits, counter and timer start address and station program version is included.

Data type:	Vector
Value:	Vector of 10 integer elements
Indexing:	1 ... 10, that is all information is always read with the same object notation. The start address needed to use the CT attribute is found in bytes 7 and 8. The exact meaning of each byte in the DS vector is described in the station manual.
Access:	Read-only

17.4.4.6 LS Last Error Status

The SYS600 error code for the last error that NET unit has discovered in a spontaneous message from a station.

Data type:	Integer
Value:	0 … 65535
	A status code, see the manual "Status Codes"
Access:	No restrictions

In the reply message to the station, NET unit sends the corresponding ANSI error code. This can be read in the station. The SYS600 error codes are translated to ANSI codes. See the Status Codes manual.

17.4.5 Station communication parameters

These attributes can be used only in connection with some station types (for example SPSC500M and Allen-Bradley PLC-2). The attributes are write-only, and their values are not stored in the communication unit, but directly transmitted to the station. Normally, the parameters are set in the process units.

17.4.5.1 EN Number of Enquiries

Maximum number of ENQs (response requests) per message from the station communication unit. The station will send ENQs in full duplex communication if it does not receive a response (ACK / NAK) to a command within its time limit (see the TI attribute below).

Data type:	Integer
Value:	0 ... 255
Suggested value:	Larger than 2
Access:	Write-only

17.4.5.2 NA Number of NAKs

Maximum number of NAKs (negative acknowledgement) the NET unit of the station accepts at the transmission of a message in full duplex communication. When this limit is reached, the message transmission has failed.

Value:	Integer, 0 ... 255
Suggested value:	3
Access:	Write-only

17.4.5.3 TI Length of Time-out

Time limit used by the station when it is waiting for a response to a message transmitted.

Data type:	Integer
Values:	0 ... 255, 255 = Infinite
Unit:	1/38 s in PLC-2 and SPSC500M
Access:	Write-only

17.4.6 Memory area definitions

The part of the station memory visible to other devices is divided into a number of memory areas (max. 30) for different types of data: binary input (BI), binary output (BO), and analog input (AI), and analog output (AO) and transparent data (TD). In addition, data may be transmitted with or without time stamps. Analog values may be coded as BCD numbers, floating-point numbers or binary numbers, etc.

The communication program needs definitions for each memory area of a station to know how each memory area is to be used, that is, to enable correct data interpretation and access checking. The memory area definitions specify the location of the data types in the data tables of the stations, the data coding, time stamping, message split (see Section 17.4.7), address format, and protected or unprotected write.

A memory area definition in the NET unit consists of a collection of eight attributes, namely DT, CO, AD, LE, AT, BF, TS and SL. The SL attribute is described in Section 14.

A memory area definition is added and removed using the MR attribute. When a new memory area is added, a memory area number that is used as identifier when referring to the area is defined. The entire memory area configuration for one STA can be copied from another STA with the MC attribute. The memory area defining attributes contain no process data. Usually, several memory areas are defined for a station. For this reason, the number of a memory area is given as indices to the attributes. Only one index is allowed. On-line changes in the memory area definitions are possible only if the IU attribute of the station has first been set to 0.

When a new STA object using the ANSI X3.28 protocol is created the following memory areas are automatically created:

No 1: BI, with default attribute values (see the MR attribute)

No 2: BO, with default attribute values (see the MR attribute)

No 3: AI, with default attribute values (see the MR attribute)

No 4: AO, with default attribute values (see the MR attribute)

No 5: AI, AD = 10 octal, BF = 2, LE = 1 (For spontaneous Commands from the station)

17.4.6.1 AD Start Address

The word address of the first word of the memory area.

Data type:	Integer
Value:	The NET unit allows values from 0 to 32767, the size of the station memory may set a lower maximum value.
Indexing:	Memory area number. Only one index is allowed.
Access:	Read, conditional write

17.4.6.2 AT Access Type

Defines if write commands directed to this memory area are protected or unprotected. The attribute is relevant only to Allen-Bradley stations.

With unprotected commands, any station can write anywhere in the data table of a PLC, if the unprotected commands are not disabled with a dipswitch. Concerning protected commands, the PLC program contains definitions stating which station is allowed to write in the memory locations. Write commands from undefined stations or to undefined areas not accepted.

Data type:	Integer	
Value:	0	Unprotected
	1	Protected
Suggested value:	For AO and BO memory areas of Allen-Bradley PLC stations, recommended value is AT = 1.	
Indexing:	Memory area number. Only one index is allowed.	
Access:	Read, conditional write	

17.4.6.3 BF Block Format

States if the spontaneous command messages from the station use the basic format of the protocol or if an additional address field is utilized.

The need for special formats is due to the implementation of spontaneous transmission into Allen-Bradley PLC-2 programs. In this programmable logic, a sent command includes a command line that contains a constant memory address. At transmission, the PLC adds this constant address to the word address field. However, sending data from several memory addresses may lead to a great number of command lines, which consume a lot of memory and programmer time. By adding an additional address into the data part of each message, the sending of commands requires only one or a few command lines. The additional address identifies the data elements the values of which are transmitted.

The constant address of the command line is chosen as the start address of a memory area with the BF value 2 or 3, depending on the coding of the second address. The additional address is defined in a memory area with BF = 1.

Data type:	Integer	
Value:	1	Allen-Bradley basic format
	2	Special format 1, the message contains a second word address, which is a BCD coded octal number
	3	Special format 2, the message contains a second, binary word address
	4	Multi-event format transmission for spontaneous event messages. This format allows transmission of many events with non-continuous addresses in the same telegram.
Suggested value:	The use of value 4 is suggested if it is supported by the station	
Indexing:	Memory area number. Only one index is allowed.	
Access:	Read, conditional write	

17.4.6.4 CO Coding

Coding of the data elements in the address interval defined by the memory area. The value of CO tells the communication program how to interpret the data of the memory area.

Data type:	Integer	
Value:	1 ... 12:	
	1	8 bit binary value
	2	12 bit binary value
	3	16 bit binary value
	4	32 bit binary value
	5	3 digit BCD value
	6	4 digit BCD value
	7, 8	Not in use
	9	32 bit floating point value
	10	ASCII data
	11	16 bit integer
	12	32 bit integer
	In a message, data elements of type 1 and 10 reserve 1 byte each. Elements of types 1, 2, 5 and 6 reserve 2 bytes and elements of types 4 and 9 reserve 4 bytes each.	
Indexing:	Memory area number. Only one index is allowed.	
Access:	Read, conditional write	

17.4.6.5 DT Data Type

The data type of the memory area. There are five types of memory areas BI, BO, AI, AO and TD. Memory areas of the types BO and AO can be used for both reading and writing data. There is also a certain memory area for the time synchronization area.

Data type:	Integer	
Values:	1 ... 6:	
	1	BI, binary input
	2	BO, binary output
	3	AI, analog input
	4	AO, analog output
	5	TD, transparent
	6	Time sync data
Indexing:	Memory area number. Only one index is allowed.	
Access:	Read, conditional write	

Example:

The type of the memory area number 1 of station 2 is set to binary input:

```
#SET STA2:SDT1=1
```

17.4.6.6 LE Length

Number of words in the memory area.

Data type:	Integer
Value:	0 ... 32767. In practice the station memory size and other memory areas that are used set the upper limit.
Indexing:	Memory area number. Only one index is allowed.
Access:	Read, conditional write

17.4.6.7 MC Memory Configuration

States the number of the station from which the configuration is copied. Using this attribute the whole memory area configuration of a station can be copied from another station.

When the configuration of station A is copied from station B, any old memory area definitions of station A will be overwritten. If an area nr x is defined for station A but not for station B, it will be removed.

Data type:	Integer
Value:	Station number
Indexing:	Memory area number. Only one index is allowed.
Access:	Read, conditional write

Example:

All memory area definitions of STA5 are copied (using the same memory area numbers) to STA1:

```
#SET STA1:SMC = 5
```

17.4.6.8 MR Memory Rung

Addition and removal of memory area definitions in a station data structure. Giving the MR attribute a string value with "C" as the first character creates a memory area. The NET unit then assigns default values to the memory area attributes. After the "C" may follow the two-character abbreviation of the data type wanted (AI, BI, AO, BO, and TD). By specifying the data type in the creation command, the application programmer can help the communication program to choose appropriate default values. Fewer SCIL commands are needed for completing the definition of the memory area. For example "CBI" means that a binary input memory area is created. If no other characters follow the "C" in the creation command, the NET unit will create a memory area of the type TD (Transparent Data). A memory area definition is removed by giving the value "D" to the MR attribute.

When a new memory area is added, the communication unit uses the following default values:

DT	BI (Binary Input)	Description
DT	1	Binary Input
CO	3	16 bit binary number
AD	0	
LE	2	
AT	0	Unprotected
BF	1	Basic Allen-Bradley format

Table continues on next page

TS	0	No time stamp
SL	0	All five elements

DT	BO (Binary Output)	Description
DT	2	Binary Output
CO	3	16 bit binary number
AD	0	
LE	2	
AT	1	Protected
BF	1	Basic Allen-Bradley format
TS	0	No time stamp
SL	0	All five elements

DT	AI (Analog Input)	Description
DT	3	Analog Input
CO	5	3 digit BCD number
AD	0	
LE	2	
AT	0	Unprotected
BF	1	Basic Allen-Bradley format
TS	0	No time stamp
SL	0	All five elements

DT	AO (Analog Output)	Description
DT	4	Analog Output
CO	5	3 digit BCD number
AD	0	
LE	2	
AT	1	Protected
BF	1	Basic Allen-Bradley format
TS	0	No time stamp
SL	0	All five elements

DT	TD (Transparent Data) or undefined	Description
DT	5	Transparent Data
CO	3	16 bit binary number
AD	0	
LE	2	

Table continues on next page

AT	0	Unprotected
BF	1	Basic Allen-Bradley format
TS	0	No time stamp
SL	0	All five elements

In any case, appropriate values must be assigned separately to the attributes AD and LE, before the new memory area is ready to be used.

At the creation of a new memory area, the attribute values can be copied from another memory area, within the same station or in another one. In this case, the attribute is assigned a coded integer value.

Data type:	Text, integer
Value:	Creating a memory area: Text of three characters. The first character is a C and the next two characters are the data types (AI, BI, BO, AO, and TD).
	Copying a memory area: Integer formed in the following way:
	100 x STA number + area number
	Removing a memory area: Text of one character: "D".
Indexing:	Memory area number, 1 ... 30. Only one index is allowed.
Access:	Read, conditional write

Example:

Area number 17 of station number 4 is copied to area number 5 of station number 1:

```
#SET STA1:SMR5 = 417
```

17.4.6.9 TS Time Stamp

States whether time tagged information is included in spontaneous commands from the station.

For the registration of signal sequences it is often desirable to "stamp" some data with the actual time already in the station. The time stamp is made by copying the minute, second and millisecond values from the station clock. If present in a message, a time stamp occupies 4 bytes, one for minute, one for second and two for milliseconds.

Data type:	Integer	
Value:	0	No time stamp
	1	Time stamp
Suggested value:	Most commonly, time tagging is used for binary input data and two bits ("double") indications (defined as AI areas in the communication unit).	
	If the station sends time tagged messages, the TS attribute must be 1, else TS = 0.	
Indexing:	Memory area number, only one index is allowed.	
Access:	Read, conditional write	

If time stamp is used, the station clock should be synchronized to the base system real time clock. This is accomplished with SCIL using station attribute SY Clock Synchronization.

17.4.7 Message split

Spontaneous messages from the station are sent to the application specified by the AS attribute (see Section 17.3). The split feature means that the NET unit copies the spontaneous messages from the station to other applications. The messages are also copied to the destination application defined by AS, see Figure 22. The feature must be activated for each STA individually. The receiving applications are memory area specific.

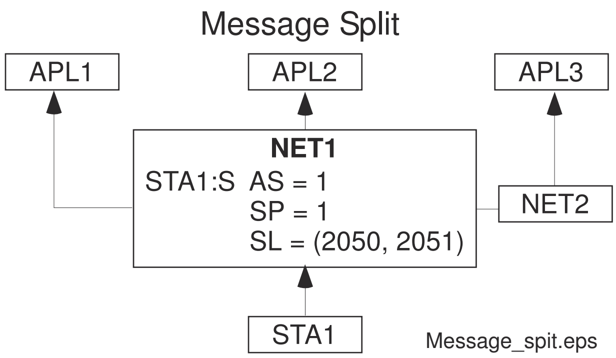


Figure 22: An illustration of message split

17.4.7.1 SL Split Destination List

A list of the applications, that will receive a copy of spontaneous messages with an address in a certain memory area. If the SP attribute of the station is <> 0, the NET unit copies an arrived message to all applications in the list. The maximum number of copy destinations is five.

Data type:	Vector
Value:	A vector of five integer elements, which can have the following values: 0 Element not used 2049 APL1 (2048 + 1) 2050 APL2 (2048 + 2) ... ... 2298 APL250 (2048 + 250) Any of the values above can be given to any element in the list. The attribute is found among the Memory Rung (MR) attributes. It is given as a number of five digits, where each digit is an application number. This means that only the applications with numbers 1 ... 9 can be defined as SPLIT applications.
Indexing:	The indices used in connection with the SL attribute are obtained from the formula: Index = 100 * (memory area no) + (SL element number) where 'memory area no' = see the MR attribute in Section 14 'SL element number' is a sequential number, 1 ... 5
Access:	No restrictions

Example:

Spontaneous messages from the 3rd memory area of station 1 will be copied to APL3:

```
#SET STA1:SSL304=2051
```

17.4.7.2 SP Message Split

Specifies if message split is used or not. It also specifies the error handling in those cases where one or several receiving applications do not reply.

The copy destination applications for different memory areas are the ones defined by the SL attribute.

Data type:	Integer	
Values:	0	No message split (copying)
	1	Message split activated. Non zero status code in reply message if one of the applications does not reply.
	2	Message split activated. Non zero status code in reply message if no application replies
	3	Message split activated. Non zero status code in reply message if the destination application defined in the message does not reply.
Default value:	0	
Access:	No restrictions	

17.4.8 Memory access

17.4.8.1 ME Memory

The data element(s) in the memory area(s). This attribute is used for reading from and writing data to the memory area of a station. The attribute is indexed with the station memory addresses (word addresses). For access to binary inputs or binary outputs, bit numbers may also be used.

Data type:	Integer
Value:	Binary inputs and outputs: 0 or 1
	Analog values: Depends on the data coding (the CO attribute, see Section 17.4.6)
Indexing:	Word address and possibly bit address. If bit address is used, the word address must be given as an octal number. Word addresses can be given as an address range, but not if bit addresses are used.
Access:	No restrictions

Example:

Memory address 1003 in STA1:

```
STA1:SME1003
```

All memory addresses in the given interval:

```
STA3:SME(3121..3127)
```

Bit number 5 in address 1234:

```
STA5:SME1234^5
```

1.4.9 Time synchronization

1.4.9.1 SY Clock Synchronization

Synchronizing the station time with the NET time. The time in the message is the NET time at transmission of the last bit of the first byte (DLE) in the message. Each station must be synchronized separately, broadcast is not supported. For stations that do not compensate for transmission time, the accuracy is not better than 50 ... 300 ms.

Using the SY attribute for synchronizing a station requires that a memory area with DT = 6 and LE = 9 has been defined, see Section 17.4.6. The address of the memory area is not significant to NET unit, but the station may require a specific address.

Data type:	Integer
Value:	1
Access:	Write-only

17.5 STA attributes, S.P.I.D.E.R. and collector RTUs

Besides the common attributes described in Section 17.3, stations of type RTU (S.P.I.D.E.R. RTUs and Collector RTUs) have the attributes described in this section.

17.5.1 Basic attributes

17.5.1.1 HR Host RTU

If the station is a sub-RTU, the HR attribute tells the station address of the host RTU one level up in the RTU hierarchy. For the uppermost RTU level, the HR attribute value is the same as the station address (the SA attribute). See Figure 23.

Data type:	Integer
Values:	1 ... 255
Default value:	The station address of the RTU. When the station address of the RTU is set, HR is automatically set to the same value.
Access:	No restrictions

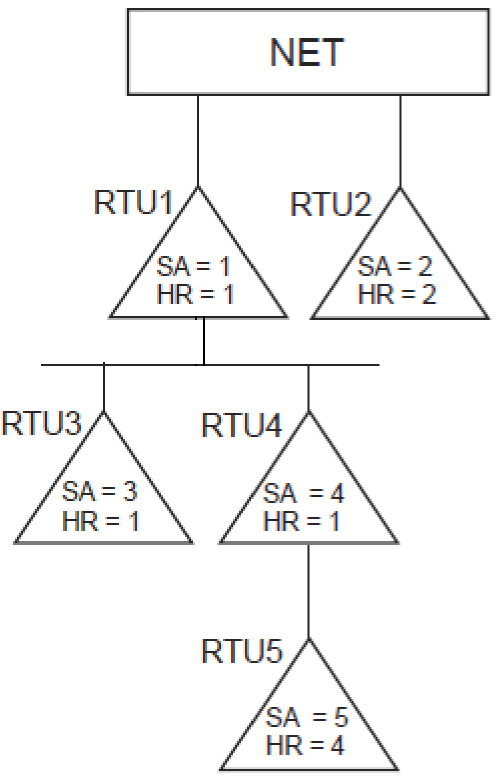


Figure 23: An illustration of the HR attribute

17.5.1.2 SA Station Address

The station address of the RTU. The address must be unique among all S.P.I.D.E.R. RTUs, Collector and P214 RTUs connected to the same NET unit. The address must coincide with the corresponding address in the RTU itself.

For S.P.I.D.E.R. RTUs legal addresses are 1 ... 255, except for the broadcast STA object, which has the address 0. The RTU can not be taken in use (see the IU attribute) unless it has a legal address.

Data type:	Integer
Value:	0 ... 255
	0 = Broadcast address
Access:	Read, conditional write

17.5.1.3 SO Synchronization Offset

This attribute is used to define a station specific time compensation to the synchronization message initiated with SY attribute. If the used hardware delays the transmission of the message to the RTUs, a value close to this delay should be assigned to this attribute. Some tuning work or a good knowledge of the used hardware is needed when this attribute is used.

Data type:	Integer
Value:	-32768 ... 32767
Access:	No restrictions
Default:	0
Unit:	10'th of milliseconds

Example:

If the average transmission delay to the station STA1 is known to be 60 milliseconds, the station object should be configured with the following SCIL command:

```
#SET STA1:SSO=600
```

The attribute can be modified while the system is running. It is possible for the SCIL application to retune, if the feedback of the synchronization accuracy is available.

Negative values will cause the RTU time to be behind the actual time. Also, too big a value, compared to the actual transmission delay, will cause the RTU time to be ahead of the actual time.

17.5.2 Diagnostic counters

17.5.2.1 DC Diagnostic Counters

The diagnostic counters for an RTU device monitors the telegram exchange to the specific RTU. The counters are:

1. SUSPENSIONS
   A S.P.I.D.E.R. RTU is suspended when the RTU line has got erroneous replies or no reply after the number of trials determined by the EN attribute of the line
2. TERMINAL STATUS RECEIVED
3. TERMINAL EVENTS RECEIVED
4. TERMINAL MESSAGES RECEIVED
   System messages from the RTU
5. PROCESS MESSAGES RECEIVED
   Indications, measurands and pulse counters
6. REPLY TIMEOUTS

Data type:	Vector
Value:	Vector of 6 integers, 0 ... 30000
	See the DC attribute in Section 14
Indexing:	Counter number
Access:	Read-only, the values can be reset

17.5.3 RTU configuration attributes

When these attributes are set, the NET unit sends an RP570 telegram to the RTU and, unless the telegram is in monologue mode (no response expected), wait for the response from the RTU. If the response is not a positive acknowledgement, an error code is returned.

17.5.3.1 FC Function Command

Enabling transmittance of function commands to the RTU. The number of the function command is given as index and additional information is given as the value of the attribute.

All function commands listed in the S.P.I.D.E.R. RTU manuals and in the RP570 protocol descriptions can be used, except the commands number 14, 15 and 16.

Example:

No	Meaning	Additional info
1	Cold start	-
2	Activate RTU after FTAB's	-
3	End of period/intermediate reading of PC's	1 = Intermediate 2 = End of period
4	Generate event with PROM version	-
5	Activate/deactivate local printer	1 = Activate 2 = Deactivate
9	Send system message queue	-
18	Deactivate RTU, accept complete set of FTAB's	-
19	Database time tag (configuration time)	Time, generated by the SCIL function RTU_ATIME

Data type:	Integer
Value:	The number of the additional info
Indexing:	One index, function command number
Access:	Write-only

Example:

To cause a cold start of an RTU, give the command:

```
#SET STA1:SFC1
```

To tag the database version with the current time, give the command:

```
#SET STA1:SFC19=RTU_ATIME
```

17.5.3.2 FT Function Table

Sending of a single function table to the RTU. The value written to the FT attribute should be the context of the function table that is stated in a text string. Note that sending function tables also demands sending the corresponding function commands depending on whether you are sending the complete set of function tables or just some alternating function table. The FT attribute is always set from a tool.

Data type:	Text
Value:	Text
Access:	Write-only

17.5.3.3 SY Synchronize

Makes an accurate time synchronization of the RTU(s). No value is necessary for this attribute because the time sent to the RTU(s) is taken from the internal clock of the computer. See also attribute SO Synchronization Offset.

When writing to the SY attribute of the broadcast object, all RTUs connected to the same NET unit are synchronized, one line at a time. The synchronization telegram is sent out as a broadcast telegram on each line with the RP570 protocol. Note! RTU200 and RTU210 substations do not support broadcast time synchronization commands, therefore each RTU200/210 station must be synchronized separately. The internal clock of the computer should be synchronized before synchronizing the RTUs.

Data type:	Any
Value:	Any or no value
Access:	Write-only

Example:

STA2 is synchronized:

```
#SET STA2:SSY
```

17.5.4 Process communication

17.5.4.1 RD Read Transparent Data

The response to transparent data, see the TD attribute below.

Data type:	Vector
Value:	Vector of max. 224 elements
Access:	Read-only

17.5.4.2 RT Reply Timeout

Determines the maximal time that the NET unit will wait for a telegram response from the RTU.

Data type:	Integer
Value:	0 ... 655
Unit:	Seconds
Default value:	20
Recommended value:	This time should be at least as long as the timeout for the corresponding communication line multiplied by the number of retries on that line. Note that when sending some function commands the response time from the RTU is longer than during normal communication.
Access:	No restrictions

17.5.4.3 SC Status Check Request

The SC attribute is used when the application desires to "force out" a status check instruction (SCI). By this command it is possible to update the process database completely from one RTU (for example at application start-up if the NET unit is not started at the same time). The SC attribute is automatically set to 1 when the station is taken out of use (IU=0). An SCI is sent when the station is taken into use again, unless the SC attribute has been set to 1 manually. The SC attribute is automatically set to 0 after an SCI.

The status check is sent automatically to all connected S.P.I.D.E.R. RTUs when NET unit is started. It is also sent automatically to suspended RTUs.

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

Example:

```
#SET STA2:SSC = 1
```

17.5.4.4 SM Sync Mode

Selecting the time synchronization mode, monologue or a dialog.

Data type:	Integer	
Value:	0	Monologue
	1	Dialog
Default value:	0	
Access:	No restrictions	

17.5.4.5 TA Transfer Address

Enables the registration of RP570 telegrams (transparent data telegrams (TDR), SYSM (terminal messages), terminal status (TSTA), ERMFD messages and ERMIR messages) in bit stream process objects. The TA attribute specifies the addresses of the receiving process objects when the telegram type is given as the index. Transfer address 0 for an index means that the telegram type is not updated as a bit stream.

Giving an address to transparent data (index 1) means that the whole transparent data telegram is updated in the bit stream object with the given transfer address, and no system message is generated. TA(1) = 0 means that a system message is generated but the telegram is not updated in a process object.

Giving an address to terminal messages (index 2) means that the whole terminal message content is sent into a bit stream object with the given transfer address, and additionally a system message is generated. TA(2) = 0 means that a system message is generated, but the message is not updated in a process object.

Giving an address to the terminal status messages (index 3), the messages are updated as a bit stream message in the process object with the given address. The messages are updated each time NET unit receives a TSTA message, which indicates the change of RTU terminal status. If TA(3) == 0 (default), NET unit uses the old transfer method, which means that NET unit sends a system message when a terminal status change has occurred. In that case, each bit change in the terminal status causes a system message, that is, one TSTA message may cause 16 system messages.

To make the bit stream messages readable, the SYS600 application must contain a command procedure that translates the messages to terminal status information. The TSTA bit stream contains 40 bits as follows:

IDENT	Bits 1 ... 8. Terminal status identity.
FLAGS Bits	9 ... 24. Flags for changed bits of TSTA status. True bit indicates which bits are changed in terminal status.
STATUS	Bits 25 ... 40. Current terminal status.

Giving an address to the ERMFD messages (index 4) means that the messages are updated as bit streams in process objects. If TA(4) = 0, the ERMFD messages are updated in analog process objects (RTU object type 11, RTU analog event recording object).

Giving an address to the ERMIR messages (index 4) means that the messages are updated as bit streams in process objects. If TA(4) = 0, the ERMIR messages are updated in analog process objects (RTU object type 10, RTU indication event recording).

Data type:	Vector	
Value:	Vector or 5 integers in the range 0 ... 4095	
	0 = the telegram is not updated in a bit stream process object	
Indexing:	One index, 1 ... 5. Omitting the index correspond to index 1.	
	1	Transfer address for transparent data
	2	Transfer address for terminal messages (SYSM)
	3	Transfer address for terminal status (TSTA) messages
	4	First process database address for ERMFD messages. Allowed address: 2304. When using TA(4) = 2304, the addresses 2304 ... 2559 are reserved for ERMFD data.
	5	First process database address for ERMIR messages. Allowed address: 1792. When using TA(5) = 1792, the addresses 1792 ... 2047 are reserved for ERMIR data.
Recomm. values:	Recommended value for TA(3) = 512	
Default values:	0 for indices 1 ... 3	
	TA(4) = 2304	
	TA(5) = 1792	
Access:	No restrictions	

17.5.4.6 TD Transparent Data

Writing transparent data (TDC) to the RTU. The RP570 protocol conveys the data directly. The interpretation and handling of transparent data are defined in the RTU. The response (TDR) can be read with the RD attribute a few seconds after the transparent data has been sent, or it can be updated in a process object specified by the TA attribute. When an answer arrives, the system message 12683 RTU_TRANSPARENT_DATA_PENDING) is generated.

Data type:	Text
Value:	Text of max. 224 bytes
Access:	Write-only

Example:

```
#SET STA1:STD = "ABCDEF" #ON RTU1:E1 #IF RTU1:P1 = 12683 #THEN -
@RESPONSE = STA1:SRD
```

17.5.5 Terminal reports

17.5.5.1 TE Terminal Event

Reading of the terminal events stored in NET unit. There is a ring buffer storage of 10 events for each RTU in the NET unit. Each time NET unit receives a terminal event it will send a tag number (1...999) as a system message (see the MI attribute). The corresponding event can be fetched from NET unit by reading the TE attribute indexed with the tag number.

If the event with the desired tag number is no longer found (due to buffer overflow), NET unit responds with the error code 12655 RTUC_TAGGED_EVENT_NOT_FOUND.

Data type:	Text
Value	A text of 7 bytes. The first byte is the event number and the rest of the bytes informational.
Indexing:	One index. The event tag number.
Access:	Read-only

17.5.5.2 TM Terminal Message

Reading of the terminal message (system message in the RTU) stored in the NET unit. There is a ring buffer storage of 3 messages for each RTU in the NET unit. Each time a terminal message from RTU, NET unit will send a tag number (1 ... 999) as a system message (see the MI attribute) to the base system. The corresponding terminal message can be fetched from the NET unit by reading the TM attribute indexed with the tag number.

If the event with the desired tag number is no longer found (due to buffer overflow), error code 12656 RTUC_TAGGED_MESSAGE_NOT_FOUND is returned.

Reading the TM attribute results in a 30 byte long text string. The text is used for system analysis of the RTU. Note that terminal messages are sent only by a function command request.

Data type:	Text
Value:	Text of 30 bytes
Indexing:	One index. The event tag number.
Access:	Read-only

17.5.5.3 TS Terminal Status

Reading the current (= last reported) terminal status stored in the NET unit as two 16 bit words. The terminal status is sent by the RTU after a status check (start-up) (see SC attribute) or at changes in the status during operation.

Data type:	Text
Value:	Two integers of 16 bits. The individual bits in the status words indicate different faults in the RTU, for example indication faults.
Indexing:	Index 1: word 1
	Index 2: word 2
	Only one index is allowed
Access:	Read-only

17.6 STA attributes, SPACOM

Besides the common attributes described in Section 17.3, the STA object of type SPA (SPACOM), have the attributes described in this section.

17.6.1 Basic attributes

17.6.1.1 BL Broadcast Lines

Choosing to which NET lines the broadcast messages, that is messages to station STA0, will be transmitted. SPA stations on LonTalk protocol lines (communicating via LSG device) must not be included in a broadcast. The attribute can only be used with STA0W

Data type:	Vector
Value:	Vector
	The elements of the vector determines which lines to be included in broadcast. A zero is always added first in the vector.
Default value:	0 = Broadcast messages are not sent to any NET line
Access:	No restrictions

Example:

Broadcasting a messages to line 4 and 7, sta0 is defined as station 0 to the NET:

```
#set STA0:SBL=(0,0,0,0,1,0,0,1,0)
```

To broadcast a message to only to line 8 the syntax would be

```
#set STA0:SBL=(0,0,0,0,0,0,0,0,1)
```

Now it should be possible to read the message with STA0:SBL. The answer should be the line number if only one line is specified. Otherwise it should be a vector containing the line numbers.

LonTalk protocol lines must not be included in the broadcast.

17.6.1.2 SA Station Address

The station address of the SPACOM unit used in the communication with NET unit. The station address must be unique among all SPA modules connected to the same NET line. Modules connected to different lines may be given the same station address. The station address of a STA object must coincide with the station address (slave number) defined in the corresponding SPACOM unit.

The broadcast telegrams always use the address 900 and need not be specified by this attribute.

Data type:	Integer
Value:	1 ... 899
Access:	No restrictions

17.6.1.3 UN Unit Number

Unit number of the SPA. Corresponds to the SPA station address (slave number).

Data type:	Integer
Value:	0 ... 65535
Access:	No restrictions

17.6.1.4 UT Unit Type

The type of the relay module: relay unit, alarm unit or SPA unit connected to LSG device.

Data type:	Integer	
Value:	0	Relay Unit
	1	Alarm Unit
	3	SPA unit connected to LSG device
Default value:	0	
Access:	No restrictions	

17.6.1.5 RL Router LMK

The RL attribute defines the object number (STA object number) of the LSG device to which the SPA station is physically connected and which acts as a router for the SPA station.

The attribute applies only for SPA units that are connected to the LONWORKS network via LSG devices (UT = 3). It has no meaning for the SPA units connected directly to the NET unit.

Data type:	Integer
Value:	1 ... 512
Access:	No restrictions

17.6.2 Event handling attributes

17.6.2.1 EC Event to Data Consistency Check Period

Event updated points are polled periodically with this interval to ensure that the value in the database is OK.

Data type:	Integer
Value:	0...60
	0 = no event check
Unit:	Minutes
Default value:	20 minutes
Recomm. value:	20 minutes is suitable in most cases
Access:	No restrictions

17.6.2.2 EL Event Buffer Length

The number of events stored in the station specific event buffer in NET unit. The suitable size is limited by the available free memory in NET unit.

Data type:	Integer
Value:	1 ... 65535
Default value:	20
Access:	No restrictions

17.6.2.3 EP Event Poll Priority Class

The event poll priority class of the station. Using the SPA line attribute PT (see Section 14), it is possible to define a ratio between event polls to stations of different priority classes.

Data type:	Integer
Value:	1 or 2. Event poll priority class.
Default:	1
Access:	No restrictions

17.6.3 Diagnostic attributes

17.6.3.1 DC Diagnostic Counters

Diagnostic counters keep count of various situations that can occur in the STA device. Each counter is associated with a descriptive name, but when it is accessed from SCIL the corresponding counter number (integer constant) must be used.

The diagnostic counters have the following meanings:

1. PROCESS DATA TELEGRAM RECEIVED
2. EVENT TELEGRAM RECEIVED
3. SUSPENSIONS
4. REPLY TIMEOUTS
5. BUFFER ALLOC FAILURES
6. PROCESS MESSAGES RECEIVED
7. ERROR IN E50 E51 RECOVERY
8. EVENT TO DATA DISCREAPANCY
9. UNEXPECTED SCM REPLY
10. ED REP FAIL NO BUFFS AVAIL

Data type:	Integer
Value:	0 ... 30000
Index:	Diagnostic Counter number, 1 ... 16
Access:	Read-only, the values can be reset

17.6.4 Station suspension attribute

17.6.4.1 RT Reply Time-out

Maximum time in seconds to wait for reply from a SPACOM unit. If the station does not answer within RT seconds, it will be suspended.

Data type:	Integer
Value:	0 ... 65535
Unit:	Seconds
Default value:	60 seconds
Recomm. value:	Do not use RT = 0
Access:	No restrictions

17.6.5 SPA point definitions

These attributes specify the handling of individual SPA points in NET unit. Station specific sequence numbers identifies the SPA points.

17.6.5.1 ED Event to Data

Defines SPA points that are updated by events. The attribute specifies which events that may update each SPA point, and how the event codes shall be interpreted.

Data type:	Vector	
Value:	Vector of 7 ... 13 elements.	
	See the explanations of the parameters in Table 8.	
	1	Object type
	2	Channel 1
	3	Channel 2
	4	Bits per channel
	5	Significant bits
	6	Event number
	7	Event value
	8 ... 13	Event number for odd elements and event value for even elements (optional)
Indexing:	SPA point number. See the SP attribute.	
Access:	No restrictions	

Example:

In a SPOC 110C unit, channels 1 ... 8 are defined as double indications, channels 9 ... 16 as single indications. Both are event updated. The double indications use the following event codes:

E1 = 01 (closed)

E2 = 10 (opened)

E3 = 11 (error)

E4 = 00 (error)

Single indications use the following codes:

E1 = input activated (closed)

E2 = input reset (open)

Defining the double indications:

```
#SET STA1:SSP1 = (0,1,8,"I",1,1,1,0,255,255,2)
#SET STA1:SED1 = (0,1,8,1,2,1,1,2,2,3,3,4,0)
```

Defining the single indications:

```
#SET STA1:SSP2 = (0,9,16,"I",1,1,1,2,0,0,2)
#SET STA1:SED2 = (0,9,16,0,1,1,0,2,1)
```

Defining SACO 16D indication (1/16I1) as event updated:

```
#SET STA1:SSP1 = (0,1,16,"I",1,1,1,4,0,0,2)
#SET STA1:SED1 = (0,1,16,1,1,1,1,2,0)
```

Table 8: Explanations of the SPA point and event updating definition parameters (the SP and ED attributes)

Parameter	Explanation
bit transpose mask	Integer, 0 ... 65535. The bits in the bit mask of the integer specify in pairs a possible change of bit order in double indications. "00" = no change of order. "11" = change of order. In SYS600 the first bit in a double indication is supposed to be "closed" and the second bit "open". If the SPACOM unit uses another order, the bits must change order.
bit type mask	Integer, 0 ... 65535. The bits in the bit mask of the integer specify in pairs the type of indication: "11" = double indication, "0" = single indication.
channel 1	Integer, 0 ... 999. The lowest channel which updates the point.
channel 2	Integer, 0 ... 999. The highest channel which updates the point.
data format	1 = bits 2 = hexadecimal 3 = real 4 = long integer
data category	The data category as defined in the SPA protocol (v.2.4) given as a text: "I", "O", "S", "V", "M", "C", "F", "T", "D", "L", "B".
data nr 1	Integer, 0 ... 65535
data nr 2	Integer, 0 ... 65535
event number	An event number that updates the point.
event value	The value that the point is updated to when the event specified by 'event number' occurs.
filter (deadband)	Real positive decimal value 0 ... 0.999 (less than 1). The smallest change in input value that is reported to the process database.
object type	Integer 0 ... 7. 0 = indication, 1 = digital input, 2 = analog input, 3 = digital setpoint, 4 = analog setpoint, 5 = object command, 6 = pulse counter, 7 = event code parsing (for internal use only)
process object address	Integer, 1 ... 255. The block address of the process object corresponding to the SPA point (as defined in the process object definition.
bits per channel	Integer, 0 ... 15. The number of bits per channel.
significant bits	Integer, defines the count of the bits that are affected by the event to data conversion. 1 - single indication, 2 = double indication.
updating method	1 = cyclical polling 2 = event update 3 = event consume. Events are used for the updating of the corresponding process object, but not for updating of the event handling object.

17.6.5.2 SP SPA Point

Defines the SPA points to NET unit. It ties together the SPA identifications and the corresponding process objects. Each SPA point, independent of updating method, must be defined by this attribute. A SPA point number identifies each SPA point, which must be unique among all SPA points within the SPA module.

If the object type is analog input or pulse counter and the amount of the configured data points is greater than 1 (data2 - data1 > 1), the entered SYS600 process object address is understood as a starting address and is updated with value 'data 1'. The deadband value given as element 9 is meaningless in this configuration. The maximum amount of values in one definition is 15. See the example at the end of this description for more information.

When writing to this attribute, all parameters must be present. See the parameter explanations in Table 8. The SPA points in SPA units connected via LSG device are defined mainly in the same way as SPA points connected via the SPA protocol. However, there are some differences in the analog point definition.

Data type:	Vector		
Value:	Vector of 7 ... 11 elements. The meaning of the elements depends on the type of the SPA point as described below. For an explanation of the parameters, see Table 8.		
	Indications:		
	Element	1:	Object type
		2:	Channel 1
		3:	Channel 2
		4:	Data category
		5:	Data 1
		6:	Data 2
		7:	SPA data format
		8:	SYS600 process object address
		9:	Bit type mask
		10:	Bit transpose mask
		11:	Updating method
	Analog input, SPA points in stations connected via SPA protocol:		
	Element	1:	Object type
		2:	Channel 1
		3:	Channel 2
		4:	Data category
		5:	Data 1
		6:	Data 2
		7:	SPA data format
		8:	SYS600 process object address (starting address if item count > 1)
		9:	Filter (deadband, meaningless if item count > 1) (Note! When SP is read, value multiplied by 1000 is returned)
		10:	Updating method
	Analog input, SPA points in stations connected via LSG device:		
	Element	1:	22
		2:	Network variable index
		3:	LON base type:
			1 = LBT_UNSIGNED_16
			2 = LBT_SIGNED_16
			3 = LBT_UNSINGED_8
			4 = LBT_SIGNED_8

Table continues on next page

			5 = LBT_SIGNED_32
			7 = LBT_FLOAT_IEE754
			8 = LBT_STRUCTURE
		4:	Self-documentation or comment text
		5:	SNVT type according to LONMARK specifications
		6:	SYS600 process object address
		7:	Deadband (filter)
	Command point:		
	Element	1:	Object type
		2:	Channel 1
		3:	Channel 2
		4:	Data category
		5:	Data 1
		6:	Data 2
		7:	SPA data format
		8:	Process object address
		9:	Updating method
Indexing:	SPA point number, 1 ... 4095		
Access:	No restrictions		

Example:

Defining an analog point that contains the measured current on phase 3 (SPA item: channel 0, "I", data 3), in a SPAC 310 C/SPTO 1D unit. Filtering is set to 0.1*In. The SYS600 process object address is 200:

```
#SET STA1:SSP1 = (2,0,0,"I",3,3,3,200,0.1,1)
```

Defining three pulse counter points of active energy in the SPAC 330C unit (SPA item: channel 0, "V", data 8 ... 10). The delta value 0 is meaningless in this situation. The updated SYS600 process objects are 100 .. 102:

```
#SET STA1:SSP2 = (6,0,0,"V",8,10,4,100,0,1)
```

17.6.6 Miscellaneous SPACOM STA attributes

17.6.6.1 DA Data

This attribute is used for process database communication. It may not be used in SCIL programs.

17.6.6.2 PR Parameter Reservation

By writing to this attribute (value 1), the writing application reserves the right to read and write the SPA parameters using the STAn:SSM attribute of the station. By writing a zero (0) to the PR attribute, the reservation is released. Only the reserving application, or the AS application can release the reservation. By reading the attribute you get information of the reserving application.

When no reservation is active, only the AS application is allowed to access the SM attribute.

Spontaneous data (events etc.) is always sent to the application defined by the AS attribute.

Data type:	Integer, vector	
Value:	Write value, 0 or 1:	
	0	Release. Allowed always for the AS application. Allowed always if no reservation is active. Allowed for the application holding the reservation. Not allowed in other cases.
	1	Reserve. Allowed for all applications, but only if no reservation is active. When receiving this value, NET unit stores the node nr and translated object nr of the reserving application, which can be seen from the message
	Read value: A vector with three integer elements:	
	1	Node nr of application holding the reservation (0 if free).
	2	Translated application nr of application holding the reservation (0 if free).
	3	1 if the asking APL is the AS (Allocating Application), otherwise 0.
Default value:	No reservation active	
Indexing:	When read: 1 ... 3	
	When written: None	
Access:	No restrictions	

17.6.6.3 SM SPA Message

Makes it possible to communicate with a SPACOM unit by sending any SPA message and reading the reply as a text. No check of the message is performed in SCIL, or in NET unit, that is, even faulty messages are sent to the SPACOM unit.

When a SPA message has been sent from an application, the reply to the message can only be read once from the same application.

Unless a reservation has been made with the PR attribute, only the application specified by the AS attribute has access to the SM attribute.

Data type:	Text
Value:	The contents of a valid SPA-message, not including the message frame (start character, unit address, checksum, and message trailer), ending with a : character
Access:	No restrictions

Example:

Requesting SPACOM unit identification using data category "F", from a 16D alarm unit:

```
#SET STA1:SSM = ("RF:")
```

The message ">1RF:XXcr" is sent on the SPA bus.

Reading the result:

```
@R = STA1:SSM
```

%R could now be for example "<1D:SACO 16D1: XXcr"

17.6.6.4 ST State

The status of the station: OK (answering to poll messages) or suspended.

Data type:	Integer	
Value:	0	Suspended
	1	OK
Indexing:	None	
Access:	Read-only	

17.6.6.5 UP Update Points

Starts an updating of all SPA points. When the attribute is set to 1, NET unit starts to poll all defined SPA points once (including event updated points) and sends the data to the application, whether the data had changed or not. Filter values for analog points are ignored. When all points have been polled once, NET unit resets the UP attribute to 0 and sends a system message (SPAP_DATABASE_UPDATE_COMPLETE).

When reading the attribute, it tells the on-going operation.

Data type:	Integer	
Value:	0	No on-going updating (read)
	1	Start updating (write), on-going updating (read)
Index:	None	
Access:	No restrictions	

17.6.7 SPACOM attributes for TCP interface

These attributes are used only with the TCP/IP communication mode of the SPA protocol. The SPA protocol is configured to the TCP mode by giving the SD attribute of the line object the value "TCP". See also the LD line attribute.

The other parts of the SPACOM attribute interface are the same for serial and TCP modes of the SPA protocol.

17.6.7.1 CT Connecting Timeout

The maximum time of the TCP connect operation. The value of this attribute depends on the speed of LAN, remote station and the possible routers between SYS600 and the substation. It should be smaller than the HT attribute of the line but it should be big enough to enable reliable reconnecting of the substation. In a multidrop configuration, too big a value may cause communication disturbances if some of the stations is not available.

Data type:	Integer
Value:	0 ... 60000
Unit:	Milliseconds
Default:	500 ms
Access:	No restrictions

17.6.7.2 ET Reconnecting Timeout

The interval of reconnecting attempt while communication is not established.

Data type:	Integer
Value:	1 ... 255
Unit:	Seconds
Default:	30 s
Access:	No restrictions

17.6.7.3 IA Internet Address

The IP address or the host name of the remote host. The connection is established with a device in this address using port number 7001 (defined in [2]). The line must be taken into use at least once before the writing to this attribute.

Data type:	Text
Value:	Any string, max 29 characters
Access:	No restrictions

This attribute accepts the IP address in form

```
#SET STA1:SIA = "10.0.0.51"
```

or as an alias name

```
#SET STA1:SIA = "rtu51"
```

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

If the remote slave device uses a non-standard port for the communication, it can be specified followingly:

```
#SET STA1:SIA="10.0.0.51;7002" ; remote device uses port 7002
```

No space characters are allowed between the address and the port number. The port number must be in range 1 ... 65535.

17.7 STA Objects, P214 RTUs

17.7.1 Definition

Each station of type Procontrol P214 (Indactic) must be defined as a STA object (of type "PCL") in the NET to which it is directly connected. The STA object can be defined with the PC attribute, see Section 15.4.3 .

When the P214 STA objects are defined on-line (with the PC attribute), the STA attributes get the default values listed in appendix A and mentioned in the attribute descriptions.

A broadcast STA object with system object number 0 is automatically created each time the communication unit starts up. The broadcast object notates all P214 RTUs connected to the same NET.

17.7.2 Object Notation

The attributes in this section are valid only for stations of type P214. The attributes are accessed from SCIL with the object notation:

STAn:Sat

where

'n'	is the station number, 0 ... 1000, as known to the application by the station mapping, see section 12.3.4. The number is translated to system object number, 0 ... 100, as illustrated in figure 12-5.
'at'	is attribute name.

17.7.3 Attributes

The STA attributes in this section are valid only for stations of type P214. The attributes are described in the following subsections:

Section 17.7.4	Section 17.7.4: IU, LI, SA
Section 17.7.5	Section 17.7.5:AL, AS
Section 17.7.6	Section 17.7.6: DC, RT
Section 17.7.7	Section 17.7.7: MI, MS, OS
Section 17.7.8	Section 17.7.8: DA, EC, FC, FE, GP, NR, TV
Section 17.7.9	Section 17.7.9: PC, PM

17.7.4 Basic Attributes

17.7.4.1 IU In Use

This attribute states whether the station connection is in use or not. The attribute tells the state of use as known to the communication unit. It does not affect the station itself, only its image in the communication unit.

The station sends no system messages as long as it is out of use. At the moment when the station is taken out of use a system message is sent.

Values:	0	Not in use
	1	In use
Default value:	0	
Access:	No restrictions	

17.7.4.2 LI Line Number

The number of the NET line, to which the RTU is connected. The station is switched from one line to another by writing a new value to the LI attribute. Change of line in this way is possible only if both the previous and the new lines are defined with the same protocol and have been taken out of use.

Value:	NET line number, Integer 1 ... 12
Default value:	The NETn:SPC attribute

17.7.4.3 SA Station Address

The station address of the RTU. The address must be unique among all S.P.I.D.E.R. RTUs and P214 RTUs connected to the same NET. The attribute must have the same value as the station address in the corresponding RTU.

Value:	Integer, 0 ... 255. 0 = Broadcast address
Access:	Read, conditional write

17.7.5 Device Reservation

17.7.5.1 AL Allocation

The attribute tells whether or not the RTU is reserved by a certain application (see the AS attribute).

Rec. value:	For P214 connections AL should always be 1, i.e. the allocation is always active (AL = 1).

17.7.5.2 AS Allocating Application

The number of the application which has reserved the RTU. The spontaneous messages from the station are sent to this application. Other applications can send read commands to the station but do not get any spontaneous messages.

Value:	Integer, 0 ... 250. The application number as known to the communication unit. 0 = no application.
Access:	No restrictions

17.7.6 Diagnostics and Suspension

RTUs of type P214 are suspended in the following situations:

- When the RTU line has got erroneous replies or no reply after the number of trials determined by the EN attribute of the line (Section 16.4).
- When a reply message from a station does not arrive in time (REPLY TIMEOUT), see the RT attribute.

17.7.6.1 DC Diagnostic Counters

P214 type stations have the following diagnostic counters:

1.	STATION_SUSPENSIONS COUNTER
2.	DEV_STATUS_RECEIVED COUNTER
3.	REPLY_STATUS COUNTER
4.	STS_NOT_OK_FROM_PCL COUNTER
Value:	Vector of four integers in the range 0 ... 30000. Each element is a counter value.
Indexing:	Counter number.
Access:	No restrictions.

17.7.6.2 RT Reply Time-out

The maximum time (number of seconds) that the communication unit will wait for a reply from the station.

Value:	Integer, 0 ... 655. Number of seconds.
Default value:	45
Rec. value:	For point-to-point connected stations, the recommended value is approximately 15 s, and for multidrop connected stations approximately 45 s. To prevent unnecessary time-outs caused by data link layer retransmissions, the value of RT should be significantly greater than the product of the TI and EN line attribute values (TI*EN).
Access:	No restrictions.

17.7.7 System Message Handling

P214 RTUs send system messages, e.g., in the following situations:

- The station is suspended or recovers from suspension.
- At start-up.
- When TERMINAL_STATUS occurs.

Based on the system messages from STA devices, the SYS600 base system automatically updates the validity stamp of the object values in the process database (the OS attribute), see Figure 16 and Section 15.3.4.

Refer to Section 15.3.4 to learn more about the system message handling.

17.7.7.1 MI Message Identification

The object address (the POA attribute) to which the system messages from the device are sent. See the MI attribute in Section 15.3.4.

The P214 system messages are of two types:

1:	Codes generated in NET
2:	Terminal status codes
Value:	Vector of two integers, 1 ... 16380
Indexing:	Message type number, 1 or 2
Default value:	Index 1: 1000 + station number (NET messages)
	Index 2: 1500 + station number (terminal status codes)
	These default values can be used as such (copied to the process object address), or they can be changed.
Example:	See the MI attribute in Section 15.3.4

17.7.7.2 MS Message Application

The MS attribute is the system object number of the application which will receive the system messages from the station.

Value:	Integer, 1 ... 250. The APL object number as known to the communication unit.
Default value:	1
Access:	No restrictions

17.7.7.3 OS Object Status

Writing to the OS attribute (OS = 1) of a station makes NET retransmit the last system message caused by the station. Possible Stopped and Suspended messages cause old-marking of process objects. By reading the OS attribute, the status code of the system message can be read.

Value:	Integer	
	When written: 1 = retransmit system message	
	When read: a status code:	
	0	OK
	12803	Station not in use
	12801	Station suspended

17.7.8 Data Communication Attributes

17.7.8.1 DA Data Value

The data of the group.

This attribute is used for reading and writing group data.

Value:	Depending on the group datatype:
	Command output: 1 or 0
	Set point: 16 bit integer
	Counter and simple data:16 bit word/cardinal
	Measurand: 16 bit integer
Indexing:	Depending on the group datatype:
	Command output: group and bit address in the form:
	group^bit_no
	where
	'group' = group number + 2000H
	'bit_no' = bit number 0 ... 15
	Set point: Group number + 5000H
	Counter and simple data: Group number
	Measurand: Group number
Access:	Depending on the group datatype:
	Command output: No restrictions
	Set point: No restrictions
	Counter and simple data: Read-only
	Measurand: Read-only

Examples:

```
#SET STA2:SDA500^1 = 1
```

Writing to a command output.

```
#SET STA3:SDA600 = 1000
```

Writing to a Set Point.

```
@V = STA4:SDA300
```

Reading a Counter, Simple data or Measurand.

17.7.8.2 EC Event Control

The event generation in the RTU can be enabled or disabled for one class at a time using the EC attribute. When the EC attribute is read, NET returns a 16 bit mask, but the attribute is written one bit at a time by indexing the attribute.

Value:	When read: Integer of 16 bits
	When written: Vector of 16 integers, 0 or 1
Indexing:	When read: No
	When written: Bit number
Access:	No restrictions

Example:

```
#SET STA2:SEC5 = 1    ;Bit number 5 is set on.
```

17.7.8.3 FC Freeze Counters

Freezes the counters of all P214 RTUs connected to NET. NET will send one freeze counter command per P214 line.

Value:	1
Access:	Write-only, only for broadcast station

Example:

```
#SET STA0:SFC = 1
```

17.7.8.4 FE Flush Events

Setting this attributes clears all the event buffers of the station.

Value:	1
Access:	Write-only

Example:

```
#SET STA1:SFE = 1
```

17.7.8.5 GP Group Parameters

Group parameters (at present, only deadband) can be read and written with this attribute. Because the number of parameters varies from group to group, the parameters should always be read first, then edited and written. When reading the GP attribute, NET returns a vector. When writing, a string variable should be used.

Value:	When read: vector of integers
	When written: text string
Indexing:	Group number
Access:	No restrictions

Example:

```
@V = STA2:SGP200
```

... editing ...

```
#SET STA2:SGP200 = string
```

17.7.8.6 NR Normalize

After start-up the RTU is normalized by writing to this attribute.

Value:	1
Access:	Write-only

Example:

```
#SET STAn:SNR = 1
```

17.7.8.7 ST Set Time

Synchronizes the clocks of all P214 RTUs to the NET clock using one broadcast command to each P214 line. The internal clock of the computer should be synchronized before synchronizing the RTUs.

Value:	1
Access:	Write-only, only for broadcast station

Example:

```
#SET STAn:SST = 1
```

17.7.8.8 TV Type Value

Writing of group type code to the attribute TV means a request for updating objects of that type in the process database.

Value:	Integer, 1, 2 or 5, the datatype	
	1	Simple data
	2	Counters
	5	Measurands
Access:	Write-only	

Example:

```
#SET STAn:STV=5
```

Requests current measurand values.

17.7.9 Priority Control Attributes

17.7.9.1 PC Priority Control Counter

This attribute controls the polling relation between the priority levels. The value of the PC attribute tells how many times in a sequence the events of the high priority level can be read, before the data of the low priority level will be read once. The attribute value is significant only if there are events on both levels in every polling cycle.

Value:	Integer 0 ... 255, number of readings
Access:	No restrictions

17.7.9.2 PM Priority Mask

With this mask the event classes of the RTU can be grouped into two priority levels. The mask is a 16 bit word. Each bit in the mask controls the corresponding event class. The ones in the mask tell which classes will be polled with a higher priority. The internal event classes (13, 1 and 15) are, however, always polled with high priority, and will always be returned as ones from NET when PM is read. When writing the PM attribute, NET ignores the contents of bits 13 ... 15.

Value:	Integer of 16 bits
Access:	No restrictions

17.8 STA attributes, REX stations

Besides the common attributes described in Section 17.3, the STA object of type REX (REF, RED, REC, etc. relays) have the attributes described in this section.

17.8.1 Basic attributes

17.8.1.1 NN Node Number

The LONWORKS node number of the station.

Data type:	Integer
Value:	1 ... 127
Access:	No restrictions

17.8.1.2 SN Subnet Number

The subnet number of the station.

Data type:	Integer
Value:	1 ... 127
Access:	No restrictions

17.8.1.3 UN Unit Number

Unit number used in transparent SPA messages (both messages resulting from commands and messages generated with the SM attribute.

Data type:	Integer
Value:	0 ... 65535
Access:	No restrictions

17.8.1.4 UT Unit Type

This attribute is obsolete, it has no functional meaning. For compatibility reasons, it has not been removed.

Data type:	Integer
Value:	0 = REx device
	1 = REC 561
Index:	None
Default value:	0
Access:	No restrictions

17.8.2 Session handling

17.8.2.1 SC Session Nack Timeout

The timer (Terr) for controlling the cyclic sending of NACK after a message sequence error. This timer is active only when the network congestion occur, and should be a bit less than the retransmit timer (Retr).

Data type:	Integer
Value:	1 ... 60000 (ms)
Default value:	750 (ms)
Unit:	Milliseconds
Indexing:	None
Access:	No restrictions

17.8.2.2 SH Session Setup Handling

Controlling and monitoring of REX device Session Setup. The session Setup mode must be configured with SH attribute before the device is taken in use.

Data type:	Integer	
Value:	When written:	
	0	No download of substituted information
	1	Download substituted information when device is started
	When read:	
	0	Download of substituted information not configured
	1	Download of substituted information configured
Default value:	0	
Indexing:	No indexes	
Access:	Read, conditional write	

Example:

```
#SET STA1:SIU = 0                    ;Stop device
#SET STA1:SSH = 1                    ;Setup Session with download substituted info
#SET STA1:SIU = 1               ;Start device. Wait for Session Startup system message
#SET STA1:SGO = %SUBSTITUTION_INFORMATION
#SET STA1:SGO = (0,0,0,0,0,0,0) ;End of commands
```

17.8.2.3 SI Session Idle Timeout

The idle ACK message interval timer (Tidle) is used to keep channel alive. It also retransmits ACK messages in case of ACK loss. In that situation the flow will be driven by the retransmission timer. The Session Idle Timeout needs to be smaller than the Session Keepalive Timeout (SK).

Data type:	Integer
Values:	1 ... 60000 (ms)
Default value:	10000 (ms)
Indexing:	No
Access:	No restrictions

17.8.2.4 SK Session Keepalive Timeout

The connection timer (Tconn) that supervises the operation of the remote node. On the idle channel both of the transmission partners send frequently so called keepalive messages. This transmission should happen in the range of 1 minute. Otherwise the connection timeouts.

Data type:	Integer
Values:	1 ... 60000 (ms)
Default value:	60000 (ms)
Unit:	Milliseconds
Indexing:	None
Access:	No restrictions

17.8.2.5 SR Session Retransmit Timeout

The retransmit timer (Tretr) is used to trigger a retransmission of the unacknowledged message if the message or ACK / NACK was lost. The Session Retransmit Timeout should be greater than the time to send a full window (max Credit).

Data type:	Integer
Value:	1 ... 60000 (ms)
Default value:	5000 (ms)
Unit:	Milliseconds
Indexing:	None
Access:	No restrictions

17.8.2.6 SS Session in Sequence Response Delay

The time that the receiver of the message waits before responding. The timer is activated after every received message. If the channel is idle the timer will timeout. During obstruct of traffic the sender will lose the Credit and flag the message for immediate ACK (TranAck flag). In such circumstances, the Tseq timer will not expire.

Data type:	Integer
Value:	1 ... 60000
Default value:	300
Unit:	Milliseconds
Indexing:	None
Access:	No restrictions

17.8.3 Process communication

17.8.3.1 DA Data

This attribute is used for process database communication. It is not used from SCIL programs.

17.8.3.2 RQ Receive Quota

Receive quota for the station. This attribute defines the maximum amount of incoming data messages received from the device but not yet acknowledged and transmitted to the SYS600 process database. This value is not used in the transparent SPA communication. Generally, the default value is suitable. A bigger value increases the buffer consumption from the pool defined with the line attribute PS.

Data type:	Integer
Values:	1 ... 10
Default value:	10
Indexing:	None
Access:	No restrictions

17.8.3.3 SM SPA Message

Sending of any SPA message to the REX station. The reply that is received can be read as a character string using the SM attribute and processed in SCIL. When sending a SPA message, SYS600 does not check the correctness of the message syntax.

Data type:	Text
Value:	When written: The contents of a valid SPA-message, not including the message frame (start character, unit address, checksum, and message trailer).
	When read: The reply on a SPA message
Access:	No restrictions

Example:

Requesting SPAOM unit identification using data category "F", from a 16D alarm unit:

```
#SET STA1:SSM = ("RF::") ;This result "<1RF::XXcr" (XX=CHECKSUM) on the SPA BUS, read the result
@R = STA1:SSM                  ;%R could now be "<1D:SACO 16D1:cr"
```

17.8.3.4 TC Transaction Check

This attribute can be used to report for too high speed of issuing the SPA messages using attribute SM. If the value of TC is 1, STA object may return an error code 13356 REXC_SM_WRITE_BUSY indicating that the amount SPA messages is too big for the device. With most devices, this transaction check is not needed.

Data type:	Integer	
Value:	0	No transaction check
	1	Transaction check used
Default:	0	
Access:	No restrictions	

17.8.3.5 TQ Transmit Quota

Transmit quota for this device. This attribute defines the maximum amount of outgoing data messages transmitted to the device but not yet acknowledged by the device. This value is not used in the transparent SPA communication. Generally, the default value is suitable.

Data type:	Integer
Values:	1 ... 10
Indexing:	None
Default value:	10
Access:	No restrictions

17.8.3.6 GI General Interrogation

An application may at any time force a complete update of point data by mean of this attribute. Setting this attribute to 1 makes the NET unit send a general interrogation command to the REX unit that then reads its process connections and sends the data to NET unit. NET unit resets the GI attribute to 0 when the general interrogation termination message is received from the unit.

Data type:	Integer
Value:	0 or 1
Index:	None
Access:	No restrictions

Example:

The command activates an updating of the process objects:

```
#SET STA4:SGI = 1
```

17.8.3.7 GO General Object Handling

Sending of the general object handling commands to a REX device.

Data type:	Vector	
Value:	Vector of 7 elements:	
	1	UN Unit Address of the handled object or end of commands

Table continues on next page

	2	OA Object Address of the handled object
	3	OG Originator Address
	4	TOH Type of Handling
	5	TOV Type of Value
	6	LOV Length of Attribute Value
	7	AVA Attribute Value in bytes as defined in LAG
Indexing:	No indexes	
Access:	Write-only	

Example:

Substitute & block double point information 1 to unit number 7, object address 1342 in REX device 4, length 1 byte, originator address 3:

```
#SET STA4:SGO = (7,1342,3,4,2,1,1)
```

UN	Unit Address of the handled object. This is an end of commands flag, session start-up sequence can continue, rest of the elements are ignored. Value: 1 ... 65535
OA	Object Address of the handled object
OG	Originator Address
TOH	
Value:	Description:
0	Substitute
1	Desubstitute
2	Block
3	Deblock
4	Substitute & block
5	Desubstitute & deblock
6	Set (parameter)
7 ... 255	Reserved for future use
TOV	
Value:	Description:
0	Value not present
1	SPI Single Point Information
2	DPI Double Point Information
3	SVAF Short Floating Point Number
4	BSI Binary State Information
5	BCR Binary Counter Reading
6	VAI Signed Integer Information (16 bit)
7	VAI32 Signed Integer Information (32 bit)
8	VTI Value with Transient State Indication
9	CP16 Two Octet Binary Time
10	Time Tag Information
LOV	Length of attribute value (AVA) in bytes
AVA	Array of bytes according to TOV as defined in LAG

17.8.3.8 IL Interlocking Data

Downloading the interlocking data to a bay unit when substitution concept is used.

Data type:	Vector	
Value:	An array with five elements:	
	1	Object Address
	2 … 5	4 bytes of data holding the value
Indexing:	No indexes	
Access:	Write only	

Example:

Send interlocking data 8,1,3,6 to object address 1342 in REX device 4:

```
#SET STA4:SIL = (1342,8,1,3,6)
```

17.8.4 Event handling

17.8.4.1 EF Event Filter Number

Filter number for event sessions. This attribute value specifies, which filter is going to be used. It is specified when the PC-NET is configured when opening a session between the relay and the PC-NET. The lower the filter value is, the more signals are sent by the relay.

At the moment the default value for PC-NET is 0, which means that the REF relay sends all signals without filtering them. In this case, it would mean having a great amount of events. Therefore, it is recommended to use event filter number 2 with SYS600.

Data type:	Integer
Value:	0 ... 4
Default:	0
Suggested value:	2
Access:	No restrictions

17.8.4.2 HI Historical Events

Specifies whether history events are requested at event session start-up or not. History events are events collected before event session start-up time.

If this attribute is set to 1, all history events registered in the station since the time specified by the HS attribute (see below) are reported at the beginning of an event session.

Data type:	Integer	
Value:	0	History events are not updated
	1	History events are updated
Access:	No restrictions	

17.8.4.3 HS Event History Start Time

The start time of the history events, which will be reported at the beginning of an event session when HI = 1 (see the HI attribute above).

Data type:	Vector	
Value:	Vector of two elements:	
	1	Time in seconds as time data
	2	The milliseconds as integer
Index:	None	
Default:	1.1.1996 00:00:000 (all stored events reported) as time data and integer	
Access:	No restrictions	

Example:

Assuming NEWEST_EV is a process object whose time is used to specify the start time of history events:

```
#SET STA1:SHS=(NEWEST_EV:PRT1, NEWEST_EV:PRM1)
```

17.8.4.4 RM Running Mode

Determines if an event session is opened when the IU attribute is set to 1.

Data type:	Integer	
Values:	2 or 7:	
	2	No event session. Transparent SPA messages are possible but not as commands.
	7	Event session opened, transparent SPA messages and commands possible.
Indexing:	None	
Access:	No restrictions	

17.8.5 Suspension attributes

17.8.5.1 RT Reply Timeout

The maximum time in seconds that NET unit waits for reply from the REX unit when sending commands and transparent SPA messages.

Data type:	Integer
Value:	0 ... 655
Unit:	Seconds
Default:	20 seconds
Access:	No restrictions

17.8.6 SPA point definition

The binary output objects of the REX stations must be defined in NET unit as SPA points using the SP attribute.

17.8.6.1 SP SPA Point

The binary output objects (SPA commands) as SPA points to NET unit. It ties together the SPA command identifications and the corresponding process objects. A SPA point number identifies each SPA point, which must be unique among all SPA points within the same REX module.

Data type:	Vector	
Value:	Vector of 8 integer and text elements. When writing to the attribute, all elements must be present.	
	1	10
	2	Channel 1, integer 0 ... 999
	3	Channel 2, integer 0 ... 999
	4:	Data category
		When writing:
		Text: "I", "O", "S", "V", "M", "C","F","T","D","L" or "B"
		When reading:
		ASCII code of the character above
	5	Data nr 1, integer 0 ... 999999
	6	Data nr 2, integer 0 ... 999999
	7	Data format, integer 1 ... 4. 1 = bits, 2 = hexadecimal, 3 = real, 4 = longint
	8	Process object address, integer 0 ... 65535
Indexing:	1 ... max point def	
Access:	No restrictions	

Example:

Defining a binary output at channel 1 in a SACO16D unit (SPA items: 1O1) at OA 666:

```
#SET STA1:SSP1 = (10, 1,1,"O",1,1,1, 666)
```

17.8.7 File transfer handling attributes

17.8.7.1 FO File Transfer Timeout

With REX device File Transfer Timeout handling attribute timeout value can be changed if necessary.

Data type:	Integer
Value:	0 … 655
Unit:	Seconds
Indexing:	No indexes
Default value:	6 sec
Access:	No restrictions

Example:

```
#SET STA1:SFO = 10
```

17.8.7.2 FP File Transfer Progress

With REX device File Transfer Progress handling attribute user can follow the processing of file transfer. Value is the amount of transferred bytes. This attribute cannot be read by SCIL because the data transfer is not known by SCIL. User gets the value of progressed file transfer from FP process object attribute.

Data type:	Integer (signed 32 bit)
Value:	0 … 2 GB
Indexing:	Transfer ID
Access:	Read-only

17.9 STA attributes, LMK stations

Besides the common attributes described in Section 17.3, the STA object of type LMK (LSG devices and other LONWORKS devices, but not REX relays) has the attributes described in this section.

17.9.1 Basic definition attributes

17.9.1.1 NN Node Number

The LONWORKS node number of the station.

Data type:	Integer
Values:	1 ... 127
Indexing:	None
Access:	No restrictions

17.9.1.2 SN Subnet Number

The subnet number of the station.

Data type:	Integer
Values:	1 ... 127
Indexing:	None
Access:	No restrictions

17.9.1.3 UT Unit Type

The type of the LMK station:

- LSG device
- Multiple LONMARK devices (devices which take input from many physical devices)
- Other devices using the standard LONWORKS interface (for example Weidmüller)

Data type:	Integer	
Value:	1	LONMARK device (not multiple and not LSG device)
	2	Multiple LONMARK device (this device takes input from many physical devices)
	3	LSG device
Default value:	1	
Access:	No restrictions	

17.9.2 Polling attribute

17.9.2.1 CT Consistency Check Time

Defines the period of time that the LMK device polls network variables (each CT minutes). This ensures that the data in the local LMK database is consistent with the data in the physical device. (The LMK needs a local database to be able to handle deadband supervision). The consistency polling is initiated from PC-NET for configured LON Points of types analog input and digital input.

Data type:	Integer
Value:	0 ... 255 0 = No consistency checking
Unit:	Minutes
Indexing:	None
Access:	No restrictions

17.9.3 Process communication

17.9.3.1 DA Data

The attribute for process communication database. It is not used from SCIL programs.

17.9.3.2 GI General Interrogation

An application may at any time force a complete update of point data by mean of this attribute. Setting this attribute to 1 makes the NET unit send a general interrogation command to the LMK unit, which then reads its process connections and sends the data to NET unit. NET unit resets the GI attribute to 0 when the general interrogation termination message is received from the unit.

Data type:	Integer
Value:	0 or 1
Index:	None
Access:	No restrictions

Example:

This command activates an updating of the process objects:

```
#SET STA4:SGI = 1
```

17.9.3.3 LM LON Message

Sending any LonTalk message to the LMK station. The reply that is received can be read back from the LM attribute (as a character string) and processed in SCIL.

Data type:	Text
Value:	Write value: LON message
	Read value: The reply to the LON message
Access:	No restrictions

17.9.3.4 RT Reply Timeout

Maximum time in milliseconds to wait for reply from a LONWORKS node when sending commands and transparent SPA messages.

Data type:	Integer
Value:	0 ... 65535
Unit:	Milliseconds
Indexing:	None
Default value:	5000 milliseconds
Access:	No restrictions

17.9.4 Diagnostic attributes

17.9.4.1 DC Diagnostic Counters

Keeping count of various situations that can occur in the STA device. Each counter is associated with a descriptive name, but when it is accessed from SCIL the corresponding counter number (integer constant) must be used. LMK stations have 8 diagnostic counters.

The diagnostic counters have the following meanings:

1. LKM_PROCESS_DATA_TLG_RECEIVED
2. LMK_SENT_SPA_MESSAGES
3. LMK_SUSPENSIONS
4. LMK_REPLY_TIMEOUTS
5. LMK_BUFFER_ALLOC_FAILURES
6. LMK_TRANSPARENT_SPA_TIMEOUTS
7. LMK_UNEXPECTED_REPLY_RECEIVED
8. LMK_REPLIES_RECEIVED

Data type:	Vector
Value:	Vector of 10 integers in the range 0 ... 30000
Index:	1 ... 8, counter number
Access:	Read-only, the values can be reset

17.9.4.2 DI Diagnostic Interval

Defines the period of time that the LMK device polls node status from the physical device (each DI seconds) to make sure that the connection is alive. A failed status poll suspends the device. The diagnostic message is initiated from PC-NET to the LSG device and it uses REQUEST/RESPONSE service provided by LON.

Data type:	Integer
Values:	0 ... 65535
	0 = No connection check polling
Unit:	Seconds
Indexing:	None
Access:	No restrictions

17.9.5 LON point definition

17.9.5.1 LP LON Point

Ties together the LONWORKS Network Variable indices with process objects in the process database. A LON point number identifies each LONWORKS point in NET unit, which must be unique among all LON points referring to the same LONWORKS module.

Data type:	Vector	
Value:	Vector of 6...7 elements depending on the LONWORKS point type. When writing to this attribute, all parameters must be present	
	Analog input definition:	
	Element 1:	2
	Element 2:	Network variable index. Integer, 0 ... 4095.
	Element 3:	LON base type, see Table 9.
	Element 4:	Self-documentation text. Text of max 30
	Element 5:	SNVT type. According to LonMarc spec. (The SNVT Master List ... 1995).
	Element 6:	Process object address. Integer, 0 ... 65535.
	Element 7:	Deadband. Real, 0 ... 0.9999.
	Structure input point definition:	
	Element 1:	3.
	Element 2:	Network variable index. Integer, 0 ... 4095
	Element 3:	Self-documentation text. Text of max 30 characters
	Element 4:	SNVT type. According to LonMarc spec (The SNVT Master List ... 1995).
	Element 6:	Process object address. Integer, 0 ... 6553
	Digital output definition:	
	Element 1:	4
	Element 2:	Network variable index. Integer, 0 ...4095.
	Element 3:	LON base type, see Table 9.
	Element 4:	Self-documentation text. Text of max 30 characters.
	Element 5:	SNVT type. According to LonMarc spec (The SNVT Master List ... 1995).
	Element 6:	Element 6: Process object address. Integer, 0 ... 65535
	Analog output definition:	
	Element 1:	5
	Element 2:	Network variable index. Integer, 0 ... 4095.
	Element 3:	LON base type, see Table 9.
	Element 4:	Self-documentation text. Text of max 30 characters
	Element 5:	SNVT type. According to LonMarc spec (The SNVT Master List ... 1995).
	Element 6:	Process object address. Integer, 0 ... 65535.
	Digital input point definition:	
	Element 1:	6
	Element 2:	Network variable index. Integer, 0 ... 4095.
	Element 3:	LON base type, see Table 9.
	Element 4:	Self-documentation text. Text of max 30 characters.
	Element 5:	SNVT type. According to LonMarc spec (The SNVT Master List ... 1995).

Table continues on next page

Element 6: Process object address. Integer, 0 ... 65535.

Indexing: LONWORKS point number, 1 ... 512

Access: No restrictions

Example:

Defining an analog input with process database address OA= 6666, LON NV index = 234 and deadband 1.123:

```
#SET STA1:SLP1 = (2, 234, 2, "Phase 1current on bay 3",1, 6666, 1.123)
```

Table 9: The LON base types

Type Code	Type Name
1	LBT_UNSIGNED_16
2	LBT_SIGNED_16
3	LBT_UNSIGNED_8
4	LBT_SIGNED_8
5	LBT_SIGNED_32
7	LBT_FLOAT_IEE754
8	LBT_STRUCTURE
9	LBT_LSG_CONTROL
10	LBT_LSG_BIT_WRITE

17.10 STA attributes, SPI stations

SPI type stations correspond to SCADA systems and other control systems, which communicate with NET unit through the RP570 slave protocol in a master-slave relation where NET unit is the slave. NET unit sees the control system as a station (a STA object), and the control system (from now on referred to as 'master station') sees NET unit as a S.P.I.D.E.R. RTU200.

The RP570 slave interface in NET unit emulates a RTU200. It is parameterized like RTU200 with FTABs, which must be loaded at each NET unit start-up. FTABs can be loaded either from the master station or from the SYS600 base system.

The attribute interface of Modbus Slave station resembles the process communication attributes of SPI stations.

Besides the common attributes described in Section 17.3, the STA object of type SPI has the attributes described in this section.

17.10.1 Basic attributes

17.10.1.1 SA Station Address

The station address of the master station. The address must be unique among all SPI type stations connected to the same NET unit. The address must coincide with the corresponding address in the master station.

Data type:	Integer
Value:	1 ... 255
Access:	Read, conditional write

17.10.2 Configuration attributes

17.10.2.1 FT Function Table (FTAB)

Loading FTABs to NET unit. By using this attribute, NET unit can be parameterized with FTABs loaded from the SYS600 base system, and FTABs need not be downloaded from the master station.

The FT attribute accepts FTABs in the same format as the RTU 200 devices (RTU type STA objects) in NET unit. This format is a character string whose ASCII value represents the corresponding FTAB byte in a certain FTAB. Use the SCIL function RTU_BIN to convert hex values to ASCII. Refer to the RP-570 protocol manual to find details about the FTAB contents. The supported FTAB fields are listed in the document Functional specification for RP-570 Slave protocol in NET. Unsupported fields are ignored.

When all FTABs have been loaded, bit 8 (RTU active) in the ST1 attribute must be set to 1. This has the same effect as "FCOM ACTIVATE" from the master.

Data type:	Text
Value:	Text
Index:	None
Access:	Write-only

Example:

Writing a simple AMV FTAB for block number 100 with priority 1 and deadband 10:

```
#SET STA1:SFT=RTU_BIN("0A64000000000000010000000A00"); WRITE FTAB(s) #SET
STA1:SST1=(8,1). ; ACTIVATE WHEN ALL FTABS ARE WRITTEN
```

17.10.2.2 FV FTAB Database Version

This attribute is used to define and store the version of the RP570 database in the SPI station object. The version string consists of six info bytes with the following meaning:

Info1 ... Info2 = Days since 1.1.1980

Info3 ... Info6 = Time in 0.1 msecs since midnight

RP570 master may define this string using the function command 19 = Database version or it can be defined from SCIL using this FV attribute. The SPI station object uses terminal event 23 to report the version string which is currently active e.g. after a cold start.

Data type:	Text
Value:	String of 6 characters
Indexing:	None
Access:	No restrictions

Example:

The version string is cleared:

```
#SET STA1:SFV="000000"
```

17.10.2.3 OL Overflow Limit

This attribute defines the low and high limits for valid analog input values. If the value is outside the range defined by this attribute, the data is sent with faulty status.

Value:	Integer, -2047 ... 2047	
Indexing:	1	Lowest valid value
	2	Highest valid value
Default:	1	-2000
	2	2000
Access:	Read/write	

17.10.2.4 RM Running Mode

Consists of a set of flags that control the behavior and functionality of the SPI station. Each flag is represented by one bit of this attribute.

Data type:	Integer		
Value:	0 ... 65535, consisting of bits as follows:		
	Bit 0	Application controlled commands	
		0	The slave device always responds to incoming commands with OK status and the CS attribute is meaningless. This is the default behavior.
		1	The slave device expects that the response status is returned by the application using the attribute CS. The timeout for writing CS is defined by the attribute AT.
	Bit 1	Terminal event retransmission	
		0	Latest terminal events (TEV-messages) may be retransmitted if an SCI or a cold start (FCOM 1) is received.
		1	Terminal events (TEV-messages) are never retransmitted.
	Bit 2	RTU Deactivation (FCOM18) handling�	
		0	Deactivation is handled, database is cleared for new FTABs.
		1	Deactivation is not handled.
Default value:	0		
Access:	Read, conditional write		

17.10.3 Process data communication attributes

The following attributes are used for sending data from SYS600 base system to the master station via the NET databases. Each SPI type station defined in NET unit has its own database for data transfer to the master station. The data transmission and the activation of the data transmission must be handled on application level using cross references in the process database (the FX and FI process object attributes), event channels and command procedures. For this purpose, there are ready-made tools and procedures, which can be used as such or modified.

The data sent from the master station to SYS600 are updated in the SYS600 process database as input data.

The attributes described in this section apply also to Modbus Slave station. For more information on Modbus Slave stations, see the Configuring SYS600 for Modbus Slave Manual.

17.10.3.1 AV Analog Value

Update of changes in analog measured values in the master station. The values assigned to this attribute are transferred to the master station as AVM/AVS telegrams. The AV attribute is assigned a vector consisting of a time stamp, the analog value and a status indicator. The current value of the AV attribute can be read back, but a read operation only returns the analog value (not a vector).

The attribute is indexed by the block address of the value as defined in the master station and in NET unit.

Data type: Vector

Value: A vector of three elements:

(time, value, status)

where

'time' Is the time stamp of the data given in RP570 format as a text. The SYS600 time can be transformed to RP570 format using the SCIL function RTU_ATIME

'value' Is the analog value given as an integer number scaled to the range -2000 ... +2000. If a value outside this range is given, the value is sent to the master station with faulty status according to the RP570 protocol. This behavior can be modified with the configuration attribute OL. If the value is outside -2047 ...+2047, it is truncated to lie within this limit.

'status' Is a status indicator given as an integer in the range 0 ... 15. When status is changed using this attribute, NET unit reports it automatically as AVS to the master. The value of status has the following meaning:

Bit nr in status	Meaning
B3 ...... B0	
* 0 0 0	Normal
* 0 0 1	Low alarm zone
* 0 1 0	High alarm zone
* 0 1 1	Low warning zone
* 1 0 0	High warning zone
* 1 0 1	Spare
* 1 1 0	Spare
* 1 1 1	Faulty value

Bit B3 = 0 means Normal trend

Bit B3 = 1 means High trend

B3 is combined with B2 ... B0 to form the status word. For example 'status' = 0 means "normal trend, normal value". STATUS = 7 means "normal trend, faulty value". STATUS = 15 means "faulty value, high trend".

Indexing: 0...255, block address of the analog value

Access: No restrictions

Example:

Setting the analog value with block address 100 to an object value with status = OK (= 0):

```
#SET STA99:SAV(100) = (RTU_ATIME(%RT,%RM),%OV,0)
```

17.10.3.2 DD Double Indication

Send of changes in double indications to the master station. The values assigned to this attribute are transferred to the master station as IDS/IDM telegrams. The attribute is assigned a vector containing a time stamp, bit number, double indication value, and status indicator.

The contents of the attribute can be read back to a SCIL variable, but the result of a read operation is a single integer rather than a vector. The integer variable is a 16-bit bitmask that represents the state of all bits in this block.

The attribute is indexed by the block address of the indication.

Data type:	Vector	
Value:	A vector with 4 ... 6 elements given as follows:	
	(time, bit_nr, value, status [,ermi_enabled [,time_quality]])	
	where	
	'time'	Is a time stamp given in the format returned by the SCIL function RTU_ATIME
	'bit_nr'	Is the bit address, 0, 2, 4, 6, 8, 10, 12 or 15, of the double indication
	'value'	Is the value of the double indication, 0, 1, 2,or 3
	'status'	Is the status reported to the master system in IDS telegrams. See the ‘status’ parameter for the ID attribute.
	The rest of the parameters are optional. If they are omitted then default values will be used.	
	'ermi_enabled'	See the 'ermi_enabled' parameter for the ID attribute
	'time_quality'	See the 'time_quality' parameter for the ID attribute
Indexing:	0...255, block address	
Access:	No restrictions	

17.10.3.3 EI Ermi from Indications

Generating ERMI messages from SCIL.

Data type:	Vector		
Value:	A vector with four integer elements given as follows:		
	(time, bit_nr, value, time_quality)		
	where		
	'time'	Is a time stamp given in the format returned by the SCIL function RTU_ATIME	
	'bit_nr'	Is the bit number, 0 ... 15, of the changed bit.If the object is a double indication, only the bit specified by the ‘bit_nr’ parameter is considered to have changed, and is affected by the FTAB setup of the bit in question.	
	'value'	Single indications:	
		0	Bit changed state to off
		1	Bit changed state to on
		ERMI is generated according to the FTAB configuration for this indication block and bit number.	

Table continues on next page

		Double indications:
		In the case of double indications, the value of both bits must be sent to NET unit, since both bits must be reported in the ermi message. It is not possible to read values from the internal database in NET unit, because the indication block may have been updated several times before the EI attribute is written.
		Value range: integer 0...3
		0 Bit values 00
		1 Bit values 01
		2 Bit values 10
		3 Bit values 11
		Only the bit specified by 'bit_nr' is considered to have changed.
	'time_quality'	See the 'time_quality' parameter of the ID attribute below
Indexing:	0 ... 255, block address of the indication	
Access:	Write-only	

Example:

Single indication handling. Assume that we have a single indication at block 55 bit 8, a time stamp in the variable %TIME, and time_quality = 5. To send an ERMI message to the master station when bit 8 changes state to 0, execute the following SCIL statement:

```
#SET STA1:SEI55 = (%TIME, 8, 0, 5)
```

To send an ERMI message when bit 8 changes state to 1, execute the following SCIL statement:

```
#SET STA1:SEI55 = (%TIME, 8, 1, 5)
```

Example:

Double indication handling. Assume that we have a double indication at block 55, bits 2 and 3, a time stamp stored in the TIME variable, time_quality = 5, bit 2 = 1 and bit 3 = 0 (the double indication bits are thus 01).

To send an ERMI message when bit 2 changes state to 0 (switch opens, but has not reached end position yet), that is, the double indication bits are 00, execute the following SCIL statement:

```
#SET STA1:SEI55 = ( TIME, 2, 0, 5)
```

To send an ERMI message when bit 3 changes state to 1 (switch is open and has reached end position), i.e., the double indication bits are 10, execute the following SCIL statement:

```
#SET STA1:SEI55=( TIME, 3, 2, 5)
```

To send an ERMI message when the switch is closed again but a fault keeps bit 3 activated, while bit 2 changes state to 1, i.e. the double indication bits are 11, execute the following SCIL statement:

```
#SET STA1:SEI55 = ( TIME, 2, 3, 5)
```

17.10.3.4 EX Event Recording Message

Generating ERMx messages. By writing a vector of bytes to the EX attribute, NET unit generates ERMx message according to the data of the vector. The first element in the vector determines the type of the ERMx message.

The ERMx messages are stored in a queue in NET unit that can contain max. 100 ERMx messages. This queue is common for all ERMx messages, also the ERMI messages generated using the EI and ID attributes. If the ERMx queue overflows, the EX attribute returns en error value and NET unit sends the systems status message to the SYS600 base system and a TEV (2) message to the master station.

Data type:	Vector
Value:	Each index (block number) is a vector of 9 ... 18 elements depending on the message type. The first element specifies the type of the ERMx message. The table below shows the meaning of the elements for different types of messages. Below the table, some parameters are explained.

The vector has the following elements:

Vector element	ERMI	ERMIR	ERMA	ERMD	ERMFD
1	32	41	33	34	43
2...7	Time	Time	Time	Time	Time
8	Time quality	Time quality	Time quality	Time quality	Time quality
9	Bit number	Bit number	Limit info	Limit info	Format
10	Bit value	Bit value			Value msb
11		Rel.time msb			Value
12		lsb			Value
13		Number msb			Value lsb
14		lsb			Rel.time msb
15		Cause of transmission			lsb
16					Number msb
17					lsb
18					Cause of transmission

'time'	Vector of 6 text elements	
'time quality'	Bit0 ... Bit2	As T_QUALITY part in ERMx messages
	Bit3	OF (overflow flag)
	Bit4	NS (clock not synchronized flag)
'bit number'	0 ... 15 with single indication	
	128 … 142 with double indication	
'bit value'	0,1 with single indication	
	0,1,2,3 with double indication	
	0	intermediate position
	1	on
	2	off
	3	error
'limit info'	Bit0 ... Bit2	As STATUS part in ERMx messages
	Bit3	LLD (level limit direction)

Table continues on next page

		Bit4	LLP (level limit passed flag)
Indexing:	Block number		
Access:	Write-only		

17.10.3.5 ID Indications

Updating changes in indications (binary input data) in the master station. The values assigned to this attribute are transferred to the master station as IDS/IDM telegrams. The attribute is assigned a vector containing a time stamp, bit number, bit value, and status indicator.

The contents of the attribute can be read back to a SCIL variable, but the result of a read operation is a single integer rather than a vector. The integer variable is a 16-bit bitmask that represents the state of all bits in this block.

The attribute is indexed by the block address of the indication.

Data type:	Vector		
Value:	A vector with 4 ... 6 elements given as follows:		
	(time, bit_nr, value, status [,ermi_enabled [,time_quality]])		
	where		
	'time'	Is a time stamp given in the format returned by the SCIL function RTU_ATIME	
	'bit_nr'	Is the bit number, 0 ... 15, of the changed bit	
		The bit can be converted to a double-indication by adding 12810 to ‘bit_nr’, when ‘bit_nr’ is regarded as the number of the lower bit (0,2,4, ...14). This feature is included for backward compatibility but should not be used.	
	'value'	Is 0 or 1. 0 = bit changed state to off, 1 = bit changed state to on.	
	'status'	Is the status reported to the master system in IDS telegrams. The status value refers only to ‘bit_nr’ and can thus be set individually for all bits. Value range: integer 0...3:	
		0	OK, status reported to master only when it has changed
		1	ERROR, status reported to master only when it has change
		2	OK, status always sent to master
		3	ERROR, status always sent to master
		Values 2...3 are not really necessary since NET unit automatically sends IDS telegrams when status bits have changed. However, using values 2...3 forces IDS telegrams to be sent even in cases where neither the indication bit nor the status bit has changed.	
	Rest of the parameters are optional. If they are omitted then default values will be used.		
	'ermi_enabled'	Enables/disables automatic ermi generation in NET unit. Value range: integer 0 ... 1.	
		0	Automatic ermi generation disabled regardless of FTAB set-up
		1	Automatic ermi generation enabled
		Default = 1	
	'time_quality'	Time quality as reported in ermi messages. Value range: integer 0...5:	
		5	Seconds * 10 and below invalid

Table continues on next page

		4	Seconds and below invalid
		3	Milliseconds * 100 and below invalid
		2	Milliseconds * 10 and below invalid
		1	Milliseconds and below invalid
		0	Time is valid
		Default = 0	
Indexing:	0 ... 255. Indication block address		
Access:	No restrictions		

Example:

Setting single indication bit 0 at block 25 to 1, assuming that status is OK:

```
#SET STA99:SID(25)=(RTU_ATIME(%RT,%RM),0,1,0)
```

Setting double indication bits 2 and 3 at block 5 to 10 (= OFF), assuming that status is OK:

```
#SET STA99:SID(25)=(RTU_ATIME(%RT,%RM),2,0,0)
#SET STA99:SID(25)=(RTU_ATIME(%RT,%RM),3,1,0)
```

Setting bit 8 in block 55 to 1 without generating ERMI, status = 0:

```
#SET STA1:SID55 = (TIME, 8, 1, 0, 0)
```

Setting bit 8 in block 55 to 1 and generating ermi, status set to 1, time quality = 5:

```
#SET STA1:SID55 = (TIME, 8, 1, 1, 1, 5)
```

17.10.3.6 PC Pulse Counter

Updating changes in pulse counters in the master station. The values assigned to this attribute are transferred to the master station as PCM telegrams. The PC attribute is assigned a vector consisting of time stamp, end of period flag and pulse counter value.

The content of the attribute can be read back to a SCIL variable, but the result of a read operation is a single integer rather than a vector. The integer variable contains the last PC-value that was written to this block.

A previous value that has not yet been sent to the master station cannot be overwritten by a new value. If an attempt is made to write such a value, the SCIL STATUS SPIC_PREV_PC_VALUE_NOT_YET_REPORTED is generated. The pulse counter value may be saved and sent later.

Data type:	Vector		
Value:	A vector of three elements, given as:		
	(time, status, value)		
	where		
	'time'	Is a time stamp given in the format returned by the SCIL function RTU_ATIME	
	'status'	Is the status byte reported in RP-570 telegrams. The status bits have the following meanings:	
		Bin number	Meaning
		B0	Spare (not used)
		B1	EPR, 1 = end of period reading

Table continues on next page

	B2	IR, 1 = intermediate reading
	B3	LS, 1 = Local storage implemented
	B4	IT, 1 = Invalid Time
	B5	CT, 1 = Changed Time
	B6	IV, 1 = Invalid Value
	B7	RC, 1 = Restarted Counter
		Note CT (bit 5), and RC (bit 7) should only be set to 1 once when the error situation is detected, and to 0 thereafter. Logic for the rest of the status bits must be handled in the SCIL application, because NET unit has no way of knowing the actual state of the pulse counters. Please refer to S.P.I.D.E.R. protocol descriptions for more details.
	'value'	Is the current pulse counter value
Indexing:	0 ... 255	
Access:	No restrictions	

Example:

```
@S = STATUS                          ;Clear status by reading it once
#SET STA99:SPC(55)=(RTU_ATIME(%T),2,1234)           ;Try to write EOP value
@S = STATUS                          ;Read resulting status
#IF %S == 0 #THEN #BLOCK ;Code for clearing pulse counter here
#BLOCK_END
#ELSE BLOCK
#BLOCK_END                           ;Code for saving pulse counter here
```

17.10.3.7 TA Transparent Data Address

The TA attribute used with index 1 specifies the address of the bit stream type process object where the data part of incoming TDC messages from the master station will be sent. Address 0 disables transparent data handling and the NET unit responses with NXR to TDC messages.

NET unit sends some incoming FCOM messages directly to a bit stream type process objects in the base system. The TA attribute used with index 2 specifies the address of this process object. The following FCOM numbers are handled this way:

	FCOM20 FCOM21 FCOM22 FCOM24
Data type:	Integer
Value:	Each index of the attribute is a process object address given as an integer
Indexing:	1 or 2. TA(1) = TDC address. TA(2) = FCOM address.
Default:	Both indices = 0
Access:	No restrictions

17.10.3.8 TR Transparent Data Response

The base system is responsible to reply to a received TDC message with a TDR message. A TDR message is created by writing a byte string to the TR attribute. The NET unit adds the necessary frames to the message and sends it with priority 3 to the master station.

NET unit has a queue for 5 TDR messages, that is, NET unit can store up to 5 unsent TDR messages.

Data type:	Vector
Value:	A vector of at a maximum 230 byte strings
Indexing:	None
Access:	Write-only

17.10.4 Function Control Attributes

17.10.4.1 CB Command Blocked from Master Station

The command blocking in NET unit. When NET unit detects an exclusive command (IXC, CBXC, EXC, IHC or SPM message when EC >0) from the master station, NET unit sets the command blocked flag to 1. The commands can also be blocked by setting the CB attribute = 1 in a SCIL program. The blocking is automatically released when the time defined by the EC attribute (see below) has elapsed. The blocking can also be released by setting CB = 0.

If the base system replies with an error reply to a command, which would have caused a command blocking in NET unit, no command blocking takes place.

Data type:	Integer	
Value:	0	Not blocked
	1	Blocked
Default:	0	
Indexing:	None	
Access:	No restrictions	

17.10.4.2 CT CBXC Timeout

Time limit in minutes between CBXC and EXC telegrams. If this time limit is exceeded, the command selection is aborted.

Data type:	Integer
Value:	Integer
Unit:	Minutes
Indexing:	None
Default value:	1 minute
Access:	No restrictions

17.10.4.3 DC Diagnostic Counters

Keeping count of various situations that can occur in the STA device. Each counter is associated with a descriptive name, but when it is accessed from SCIL the corresponding counter number (integer constant) must be used. The following counters are updated:

1. ERMI_BUFFER_OVERFLOW
2. SCI_RECEIVED
3. CBXC_TIMEOUT
4. EXC_FAILED
5. IHC_FAILED
6. IHC_RECEIVED

7. SCS_REPLY_TIMEOUT
8. STATUS_NOT_OK_FROM_NET
9. TEV_BUFFER_OVERFLOW
10. TR_BUFFER_OVERFLOW
11. BLOCKED_COMMANDS
12. EXCLUSIVE_COMMAND_TIMEOUT

Data type:	Integer
Value:	Integer
Indexing:	Counter number. One index or an index range.
Access:	Read-only, the values can be reset

17.10.4.4 DI Database Initialized

Indicating whether or not the database in NET unit has been initialized and can be polled from the master station. At NET unit start-up, the DI attribute has the value 0. When all values in the NET database have been initialized (either given a value or a faulty status), this attribute must be set to 1. NET unit does not answer to polls from the master station until DI = 1.

Data type:	Integer
Value:	0 or 1
Indexing:	None
Access:	No restrictions

17.10.4.5 EC Exclusive Commands from Master Station

Specifies the command blocking time in seconds. Command blocking means that after a command from the master station, no new command is allowed until the blocking is released. The command blocking is automatically released when the time defined by the EC attribute has elapsed. It can also be released by writing a 0 to the CB attribute (see above). During command blocking the NET unit replies to incoming commands with NXR.

Data type:	Integer
Value:	0 ... 60
Unit:	Seconds
Indexing:	None
Default:	0
Access:	No restrictions

17.10.4.6 MM SYS Command Multiplier

When the data communication attributes AV, ID, EI, EX and TV are updated, NET unit updates the data in the databases of all SPI type stations, which have MM = 1. This means that the same data is sent to all these master stations. The MM attribute does not affect the transmission of transparent data.

Data type:	Integer	
Value:	0	Data not sent to this station
	1	Data sent to this station
Indexing:	None	
Access:	No limitation	

17.10.4.7 RT SYS Reply Timeout

The maximum time in seconds for waiting for a reply from the base system before regarding it as communication timeout. When NET unit receives an EXC, IXC or TXI telegram from the master station, it sends it to the receiving SYS600 application. If NET unit does not receive an acknowledgement from the base system within RT seconds, it is regarded as a communication timeout and an NXR message is sent to the master station.

Data type:	Integer
Value:	1 ... 65
Default value:	15
Unit:	Seconds
Indexing:	None
Access:	No restrictions

17.10.4.8 TI Time Initialized

When all relevant parts of the SYS600 system have received time synchronization as a result of a TSI telegram (Time Sync Instruction) from the master station, NET unit should be notified by setting this attribute to 1. As long as TI is 0 (default), the bit with the information "RTU is synchronized" is 0 in all applicable RP-570 messages.

Data type:	Integer
Value:	0 or 1
Indexing:	None
Access:	No restrictions

17.10.5 Terminal messages

17.10.5.1 ST Terminal Status

Corresponds to terminal status messages (TSTA) generated by the RP-570 interface. Each time the ST attribute is set, a new terminal status message is sent to the master station (if new status <> old status).

The attribute is used indexed. Index 1 of the ST attribute corresponds to TSTA, ID=1 messages, and index 2 to TSTA ID=2. Each index is assigned a vector value where the first element indicates the number of the bit to be operated (set or cleared). The second element determines the operation (set or clear) to be performed on the bit.

The lists below shows the status information that the master station associates with the different bits.

ST attribute, index 1, (TSTA ID=1):

Bit number	Meaning if the bit is set (=1)
0	At least one indication faulty
1	At least one analog input faulty
2	At least one digital measured value faulty
3	At least one pulse counter faulty
4	At least one object or reg. command output is faulty

Table continues on next page

5	At least one setpoint faulty
6	At least one general output. Faulty
7	RTU is faulty
8	RTU is active
9	RTU is synchronized
10	* spare *
11	Local printer is off-line
12	Command outputs are blocked due to config Error
13	* spare *
14	* spare *
15	Local printer is out of service
ST attribute, index 2, (TSTA ID=2):	
Bit number	Meaning if the bit is set (=1)
0	* spare *
1	Plain texts are loaded
2	Backup medium for plain texts faulty
2-15	* spare *
Data type:	Vector
Value:	Write: Each index is a vector of two integers, where
	Element 1 = Bit number to set or clear, 0 ... 15
	Element 2 = Operation, 1 = on, 0 = off
	Read: Each index is a 16 bit word
Indexing:	The attribute is always used with index 1 or 2
	Index 1 = TSTA, ID = 1
	Index 2 = TSTA, ID = 2
Access:	No restrictions

Example:

```
#SET STA99:SST1 = (9,1)
```

Setting the TI attribute (see Section 17.10.4) is equivalent to setting ST(1) bit 9.

17.10.5.2 TV Terminal Event Message

Activates the transmission of a terminal event message to the master station.

Data type:	Vector	
Value:	Each index is a vector with six elements. The written vector has following structure:	
	Element	1: INFO1
		2: INFO2
		3: INFO3
		4: INFO4
		5: INFO5

Table continues on next page

	6: INFO6	
Indexing:	Terminal event number. Only TEV 13 is allowed at the moment.	
Access:	Write-only	

17.10.6 Loop control attributes

17.10.6.1 LC Loop Control

Activating and de-activating the loop control. Set the IU attribute of both the line and the STA object to 0 before changing this attribute.

Data type:	Integer	
Value:	0	No loop control
	1	Loop control activated
Indexing:	None	
Access:	Read, conditional write	

17.10.6.2 LT Loop Timeout

The maximum time for NET unit to wait for reply from the master station before sending the system message to the base system. Loop switching is done if no RA or RB polls are received within LT number of seconds.

When the DI attribute is set to 1, that is, NET unit starts to wait for messages from the master station, it waits LT number of seconds. If no valid RP-570 message is received within this period, the system message SPIP_COMMUNICATION_WITH_CS_LOST is sent to the SYS600 base system. If loop control is used, the loop direction is toggled.

After each valid RP-570 message, NET unit waits LT + 30 second before SPIP_COMMUNICATION_WITH_CS_LOST is sent.

When communication has been down, and a valid RP-570 message is received, the system message SPIP_COMMUNICATION_WITH_CS_ESTABLISHED is sent to the SYS600 base system.

Data type:	Integer
Value:	Integer
Unit:	Seconds
Indexing:	None
Default value:	20 seconds
Access:	No restrictions

17.10.7 Redundant line attributes

The following attributes can be used for defining redundant lines for RP-570 Slave and IEC 60870-5-101 Slave protocols.

17.10.7.1 LI Line Number

The number of the back-up line is set to index 2 of the LI attribute. More about LI attribute can be found in Section 17.3.2.

Example:

```
#SET STA1:SLI(2) = 5
```

17.10.7.2 RU Redundant Line Station

This attribute defines the number of the STA object connected to redundant RP-570 lines. This attribute should be set both for the main and back-up lines. The information provided by this attribute is needed when a line switch operation is executed. Value 0 indicates that redundant lines are not used.

Data type:	Integer
Value:	0...255
Index range:	NET line number
Default value:	0
Access:	Read, conditional write

17.10.8 Application based command controlling

The attributes RM (bit 0), AT and CS can be used to form a negative or a positive response to an incoming control command from the SCIL application.

With setting RM bit 0 = 0, all command messages (CBXC, EXC, IXC and IHC) are always responded by PC-NET with CBR/EXR, meaning OK. This is the default behavior and if it is used the information of this section is not needed.

With setting RM bit 0 = 1, PC-NET is configured to use the enhanced mode and it waits until the SCIL application has written to the CS attribute. No automatic responses are sent in this mode.

When a control command is received by the SPI station object, it updates a process object with the same address as the object number in the incoming command message. The updated process object is of type ANSI/Digital Input.

With setting RM bit 0 = 1, the incoming command updates the attribute RA of the mentioned process object as follows:

1	check back before execute (= select)
2	execute
3	cancel
4	immediate execute

With setting RM bit 0 = 0, RA is updated with value 0 and the application need not write to the CS attribute. When a control command is received, the operation of the COM500i signal routing procedure COM_DSBO is based on the value of the RA attribute of the updated process object.

The actual communication will be as in the following table:

Command	Response when CS <> 0	Response when CS = 0
CBXC (RA -> 1)	NXR	CBR
EXC (RA -> 2)	NXR	EXR
IHC (RA -> 3)	NXR	EXR
IXC (RA -> 4)	NXR	EXR

The application should write to the CS attribute within the time specified by the SPI station attribute AT.

17.10.8.1 AT Application Status Timeout

Timeout value for the command confirmation from application. The value specifies how long the application is waited for to write to the CS attribute. This timeout should be longer than the response header timeout multiplied by the retry count in the master end. In normal situation, this attribute has no effect.

This attribute is meaningful only when the bit 0 of attribute RM is 1.

Data type:	Integer
Value:	1 ... 60
Unit:	Seconds
Indexing:	None
Default value:	20 seconds
Access:	No restrictions

17.10.8.2 CS Command Confirmation

Manual confirmation of the received messages. In RP570, the commands received by the slave station are confirmed by using specific response messages and the response message is selected using the status value given as the first parameter.

This attribute should be used only when the bit 0 of attribute RM is 1.

Data type:	Integer	
Value:	0	Previous command should be responded with status "OK".
	non-zero	Previous command should be responded with status "not OK".
Access:	Write only	

Section 18 PRI objects for communication system

18.1 About this section

This section describes the communication system PRI objects and their attributes:

Section 18.2	General: printer object definition and object notation.
Section 18.3	PRI attributes: Basic PRI Attributes (IU, LI, PT), Device Reservation (AL, AS), System Messages Handling (MI, MS, OS), Diagnostic Counter (DC), Printer Control (CD, CS, DA, PE), Printout Properties (CC, CT, PX).

18.2 General

18.2.1 Definition

Each printer connected to the process communication system must be defined as a PRI object in the NET unit to which it is directly connected. PRI objects are defined with the NET object attribute PR (Section 15). Defining a PRI object requires that the line has been defined as a printer line (Section 16). The printers must also be defined as PRI base system objects in the base systems that will use the printers.

When the PRI objects are defined with SCIL (with the PR attribute), the PRI attributes get the default values given in the attribute descriptions.

18.2.2 Object notation

From SCIL, the PRI attributes are accessed with the notation:

PRIn:Sat

where

'n'	The logical printer number, 0 ... 20, as known to the application according to the printer mapping (see the mapping attributes in Section 7). The number is translated in the base system (through the application mapping and the corresponding PRIn:B object) to system object number, 1 ... 8.
'at'	An attribute name

18.3 PRI attributes

18.3.1 Basic PRI attributes

18.3.1.1 IU In Use

Specifies whether the printer connection is in use or not. The attribute determines the state of use as known to the communication unit. It does not affect the printer itself, only its image in the communication unit.

If the printer is off-line or its IU = 0 when a printout message is sent to the printer, the message is saved in the spool queue of the base system. The message is stored until IU is set to 1.

The printer sends no system messages as long as it is out of use, only at the moment when it is taken out of use.

Data type:	Integer	
Values:	0	Not in use
	1	In use
Default value:	0	
Access:	No restrictions	

18.3.1.2 LI Line Number

The number of the NET line to which the printer is connected. The line is determined when the printer is created with the NETn:SPR attribute.

Data type:	Integer
Value:	1 ... 12
Access:	Read-only

18.3.1.3 PT Printer Type

The type of the printer as defined to NET unit. There are six types of printers which product different type or printout:

- Character based black and white printout (ASCII). The base system sends CR and LF characters within the print messages.
- Transparent printout. Printers defined as transparent can print the printout commanded by the SCIL function PRINT_TRANSPARENT.
- Pixel based, black and white printout. All EPSON FX compatible printers can be used for this type of printout.
- Character based black and white printout (ASCII). The base system does not send CR and LF characters, but it sends color information, which is not printed. Graphical characters can be replaced by printer characters using the CT attribute (Section 18.3.6).
- Character based color printout. This type is used on FACIT 4544 printers. The characters CR and LF are generated by the communication unit.
- Pixel based color printout. All EPSON JX compatible printers can be used for this type of printout.

The CR character causes a carriage return and the LF character a line feed.

The corresponding PRI base system objects must be defined as the same printer type in the base systems.

Data type:	Integer	
Value:	1, 2, 3, 5, 6, 7:	
	1	Character based black-and-white printer (ASCII)
	2	Transparent printer. Printer for the full-graphic printout.

Table continues on next page

3	Pixel based, black and white printer. All EPSON FX compatible printers can be used
5	Character based black and white printer (ASCII)
6	Character based colour printer of type FACIT 4544
7	Pixel based colour printer. All EPSON JX compatible printers can be used.

Access: Read, conditional write

NET uses the control sequence ESC ? K for setting printers of type 3 and 7 in graphical mode. This sequence is not supported by all EPSON FX-80 and EPSON JX-80 printers.

18.3.2 Device reservation

18.3.2.1 AL Allocation

Specifies whether the printer is allocated to a certain application or not.

Reservation of a printer is needed to prevent mixing of printer outputs from different program processes (in the same or in separate base systems). Therefore, the main program always reserves a printer before sending print messages (if all the data to be printed does not fit into one message). After sending the print messages, the main program releases the printer (AL = 0) automatically.

Normally, there is no need to write the AL attribute from application programs. Manual printer reservation with SCIL is needed only when using DA and CD attributes (see Section 18.3.5).

If the printer has been reserved by setting the AL attribute or it is reserved for some other reason, it must be released by a program in the same application as reserved the printer. When releasing a printer, the printer must first be taken out of use with the IU attribute (see below) before the AL attribute can be reset.

Data type:	Integer	
Values:	0	Free, not allocated
	1	Reserved, allocated
	A reservation (AL = 1) means that the printer is logically connected to one application, and that other applications have no or limited access to it.	
Default value:	0	
Suggested value:	0	
Access:	Read, conditional write	

18.3.2.2 AS Allocating Application

The number of the application that reserved the printer. NET unit automatically updates the AS attribute by the number of the writing application when the AL attribute is set to 1 on-line. When the AL attribute is set to 0 on-line, the AS attribute is set to 0 as well.

Data type:	Integer
Value:	0 ... 250. The application number as known to the communication unit. 0 = No allocating application.
Default value:	0
Access:	Read-only

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

18.3.3 System message handling

A PRI object generates system messages, for example in the following situations:

- The printer has been offline or busy for more than 10 seconds.
- The printer connection (DCD) has been lost.
- The printer accepts data again after any of the above situations.

Concerning the system message handling, see System Message Attributes in Section 15 and the System Configuration manual.

18.3.3.1 MI Message Identification

The process object address (the OA attribute of the process object) to which the system messages from the device are sent.

Data type:	Integer
Value:	1 ... 16380, receiving object address
Default value:	3000 + printer number
	This default value can be used as such (copied to the process object address), or it can be changed.
Example:	See the MI attribute in Section 14

18.3.3.2 MS Message Application

Specifies the application that will receive the system messages caused by the PRI object. Its value is the system object number of the application as known to NET unit.

Data type:	Integer
Value:	1 ... 250. The APL object number as known to the NET unit (defined by the NETn:SSY attribute).
Access:	No restrictions

18.3.3.3 OS Object Status

When written the attribute causes a re-transmission of the latest system message (write value 1), or cancels a possible OFF_LINE state (write value 0). When read, the attribute returns the current printer status.

If the printer does not send XON after it has been turned on, NET unit will regard it as offline. Writing to the OS attribute cancels the off state, provided that the printer is on.

After an application has been started, the process object that receives the system messages caused by the printer (the MI attribute) has no value (OS = 10, not sampled). An updating of the process object is achieved by writing to this attribute, for example in the command procedure started by APL_INIT_1.

Data type:	Integer
Value:	Read:
	Printer status
	(for example 0 = ON_LINE, 13103 = OFF_LINE, 13120 = OFF_LINE_AND_NO_CONNECTION)
	Write:
	1 Re-transmission of latest system message
	0 If printer is off, writing value 0 does nothing. If printer is on, writing value 0 cancels the OFF_LINE (XOFF) state. This causes a system message with the status code 13126 PRIC_PRINTER_OFF_LINE_STATE_CANCELLED.

Example:

The latest system message is retranslated and updated in the process object defined by the MI attribute of the PRI object:

```
#SET PRI3:SOS = 1
```

18.3.4 Diagnostic counter

18.3.4.1 DC Diagnostic Counter

Contains a diagnostic counter value that counts the occurrences of certain events related to the PRI objects. The counter is updated by the NET unit.

The diagnostic counter is incremented in the following situations:

- Each time a flow stop request (XOFF) character has been received from the printer, and no flow start request (XON) character has followed within 10 seconds. As an example this situation occurs when the printer runs out of paper or when a user has left the printer in off-line mode.
- When DCD(Data Carrier Detect) has been low for more than 10 seconds and the OS attribute of the printer line is 2 or 3.

Data type:	Integer
Value:	0 ... 30000. The event count(modulo 30001). Modulo 30001 is the operation that returns the remainder after division by 30001. The remainder is an integer that increments by 1 as the event count increments by 1.
Initial value:	0
Access:	Read-only, the values can be reset

18.3.5 Printer control

18.3.5.1 CD Control Data

An application program transmitting control sequences to the printer as character strings. Any 8-bit character value can be used. For example, CD can be used to choose a new font, initiate form feeds, choose a national character set or change horizontal or vertical spacing. Unlike the DA attribute (described later), CD makes it possible to send non-printable characters to the printer. Control sequences may be sent to the printer, for example when the base system is started up, or from the start programs of format pictures.

Data type:	Text
Value:	Maximum 228 characters. Any bit combinations are allowed in the characters.
Access:	Write-only

18.3.5.2 CS Control Data Store

Storing control sequences in the printer data structure and sending them automatically to the printer. The information is sent when a new printer is added or the IU attribute of a printer is changed from 0 to 1. The CS attribute can be used for printing control sequences at the initialization of printers, which is sometimes necessary.

Data type:	Vector
Value:	Max. 10 character elements. Any bit combinations are allowed in the characters.
Default value:	Carriage Return + Form Feed
Access:	No restrictions

18.3.5.3 DA Data

Transmitting data to the printer without using a format picture (for example for test purposes). (NET unit uses the DA attribute also for other purposes). By writing a character string to the DA attribute, the application program can transmit the string to the printer. In more extensive use of the DA attribute, the application programmer must provide for the printer allocation using the AL attribute described earlier. To type short printouts that go into one message, the AL attribute need not be set.

Data type:	Text
Value:	Text
Access:	Write-only

Example:

Transmitting a line with the text HI PRINTER to printer 1:

```
#SET PRI1:SDA = "HI PRINTER" + ASCII(13) + ASCII(10)
```

18.3.5.4 PE Print Enable

In some special application cases there is need to stop a printer temporarily. The stopping is done so that the messages meant for the printer are not stored in the spool queue of the base system (what is the case if IU is set to 0). This is possible by setting the PE attribute to 0.

Data type:	Integer	
Value:	0	NET unit destroys the printer messages when they arrive.
	1	NET unit transmits the printer messages to the printer when they arrive (normal operation state).
Default value:	1	
Access:	Read, conditional write	

18.3.6 Printout properties

18.3.6.1 CC Color Conversion

Enables an automatic color conversion for color pixel based printers (PT = 7). The attribute contains a programmable color transformation table that translates the video monitor colors to printer colors.

Data type:	Vector
Value:	Vector of 64 integer elements. Each element is calculated according to the formula:
	16 * (BG printer color) + (FG printer color)
Indexing:	0 ... 63 calculated according to the formula:
	8 * (BG video color) + (FG video color)
	BG = background and FG = foreground
	The video colors (background and foreground) are numbered as follows:
	0 Black
	1 Magenta
	2 Blue
	3 Cyan
	4 Red
	5 Amber
	6 Violet
	7 White
	The printer colors are numbered as follows:
	0 Black
	1 Magenta
	2 Cyan
	3 Violet
	4 Yellow
	5 Orange
	6 Green
	7 White
Access:	No restrictions

18.3.6.2 CT Character Transformation Table

This attribute applies to printers of type 5 (PT = 5).

An array of 256 elements that translates the SYS600 specific character codes to corresponding printer codes. By using this attribute it is possible to utilise graphic characters from the character set of the printer.

Data type:	Vector
Value:	Vector of 256 integers 0 ... 255. Printer character code
Indexing:	0 ... 255. SYS600 character code number
Default value:	CT (0 ... 64) = 43

Table continues on next page

	CT (65) = A
	CT (66) = B, etc.
Access:	No restrictions

Example:

Characters with code values 1 ... 20 will be printed as "A":s:

```
#SET PRI1:SCT(1..20) = 65
```

18.3.6.3 PX Pixel Characters

The attribute applies to pixel printers (PT = 3 or 7).

The printed bit pattern for each character code can be decided by the application. There is only one pixel printer character set in each communication unit. Therefore, all printers connected to the same NET unit use the same character set and a change in the set will appear in all printers.

Data type:	Vector
Value:	Vector of 8 elements. Bit patterns for each column of the character
Indexing:	0 ... 255. ASCII character code (the code received from a the application)
Access:	No restrictions

Example:

Exchange of the Swedish Å to the German Ü for character code 93 (dec):

```
#SET PRI1:SPX93 = (0,30,65,1,1,65,30,0)
```

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 511 482-UEN