MicroSCADA Pro
SYS 600 *9.1.5
System Objects

Technical Description
1. About this manual .................................................................9
   1.1. Copyrights ........................................................................9
   1.2. Trademarks ......................................................................9
   1.3. Related documents ...........................................................9
   1.4. Document revisions ..........................................................10

2. Introduction ........................................................................11

3. System object handling in SCIL ...........................................13

4. Base system objects, overview ............................................15
   4.1. Base system object types ..................................................15
   4.2. Common naming attributes ..............................................18
   4.3. Defining base system objects ............................................19

5. SYS objects for base system .................................................21
   5.1. General ............................................................................21
   5.2. Basic SYS attributes .......................................................22
      5.2.1. Basic configuration attributes ..................................22
      5.2.2. Hardware and software information .......................23
      5.2.3. Communication attributes ......................................25
      5.2.4. Time handling attributes .......................................27
      5.2.5. Memory handling attributes ..................................29
      5.2.6. Global paths and representation libraries .................33
   5.3. Software configuration attributes ....................................34
      5.3.1. Shadowing attributes .............................................34
      5.3.2. DDE Server attributes ...........................................34
      5.3.3. OPC Server attributes ..........................................36
   5.4. Device attributes ............................................................36
      5.4.1. Audio alarm device ...............................................37
      5.4.2. External clock .......................................................37
      5.4.3. SPA device attributes ...........................................40
   5.5. Operating system event handler attributes ....................41
   5.6. Miscellaneous attributes ...............................................42

6. APL objects for base system .................................................43
   6.1. General ............................................................................43
   6.2. Basic APL attributes .....................................................44
      6.2.1. Basic configuration ...............................................44
      6.2.2. Supervision configuration attributes .......................46
      6.2.3. Paths, representation libraries and text databases ......46
      6.2.4. External and alias applications .................................47
   6.3. Mapping attributes ........................................................49
      6.3.1. Application mapping ...............................................49
6.3.2. Device mapping attributes ........................................50
6.4. Shadowing attributes ..................................................53
6.5. Resource handling attributes .........................................58
6.6. Application control attributes .........................................64
   6.6.1. Functional definitions ...........................................64
   6.6.2. User interface related attributes ...............................66
   6.6.3. Operation control attributes ...................................67
   6.6.4. SCIL loop control attributes ....................................68
   6.6.5. Revision compatibility ............................................69
6.7. Application diagnostic attributes .......................................75
6.8. Mirroring attributes ....................................................76
   6.8.1. Mirroring configuration ...........................................76
   6.8.2. Mirroring diagnostics .............................................78
6.9. Miscellaneous APL attributes ............................................79

7. MON and IND objects for base system ............................81
   7.1. MON objects .................................................................81
      7.1.1. General .................................................................81
      7.1.2. Basic MicroSCADA monitor definition attributes .......82
      7.1.3. Informative MON attributes ....................................83
      7.1.4. MicroSCADA monitor control attribute .....................84
      7.1.5. Miscellaneous MON attributes ................................87
   7.2. IND objects .................................................................88
      7.2.1. General .................................................................88
      7.2.2. IND attributes ........................................................88

8. LIN objects for base system ............................................91
   8.1. General .................................................................91
   8.2. LIN attributes ..........................................................92
      8.2.1. Basic LIN definition attributes ..............................92
      8.2.2. RAM link attributes .............................................93
      8.2.3. Diagnostic counters .............................................94
      8.2.4. Miscellaneous LIN attributes ...............................95

9. NOD objects for base system ..........................................97
   9.1. General .................................................................97
   9.2. NOD attributes ........................................................98
      9.2.1. Basic node definition attributes ............................98
      9.2.2. Node diagnostic attributes ....................................100
      9.2.3. Node communication attributes ............................101
      9.2.4. Miscellaneous NOD attributes ..............................102

10. STA and STY objects for base system ............................103
   10.1. STA objects ..........................................................103
      10.1.1. General ..........................................................103
10.1.2.STA attributes .........................................................103
10.2.STY objects ...............................................................109
  10.2.1.General .............................................................109
  10.2.2.STY attributes ....................................................111

11. PRI objects for base system ........................................113
  11.1.General .................................................................113
  11.2.PRI attributes ........................................................114
    11.2.1.Common printer attributes ................................114
    11.2.2.Printer connection attributes ............................116
    11.2.3.Printout attributes ..........................................117
    11.2.4.Printer queue attributes ....................................119
    11.2.5.Printer log attributes ......................................119
    11.2.6.Printer control attributes ..................................122
  11.2.7.Miscellaneous PRI attributes ..............................123

12. Base System Object Navigator .................................125
  12.1.General .................................................................125
  12.2.Recognizing of the base system objects ....................125
  12.3.Viewing and editing of the base system objects ........126
    12.3.1.Base system objects .........................................126
    12.3.2.Base system object attributes ..........................126
    12.3.3.Base system object attribute values ...................127
  12.4.Adding of a new base system object ........................127
  12.5.Starting of the tool ..............................................128
  12.6.How to handle the object and attribute trees ............130
  12.7.Tools in the tool ...................................................130

13. Communication system objects, overview ..................133
  13.1.General ...............................................................133
  13.2.Communication system object types ..........................134
  13.3.Defining communication system objects ....................137
  13.4.Attributes ...........................................................139

14. NOD (NET) objects for communication system ..........141
  14.1.General ...............................................................141
  14.2.Basic NET attributes ............................................142
    14.2.1.Basic definitions ..........................................142
    14.2.2.Functional specifications ..................................143
    14.2.3.NET information ..........................................144
    14.2.4.System message attributes .............................145
    14.2.5.Connected nodes ...........................................148
  14.3.Object definition attributes ..................................149
    14.3.1.External nodes ..............................................149
    14.3.2.Applications ................................................150
14.3.3. Station definition attributes ........................................ 151
14.3.4. Adding devices of exchangeable device types .......... 157
14.3.5. Printer definition attributes .................................... 158
14.4. Application attributes .................................................. 159
14.5. Frontend node attributes .............................................. 161
14.6. Redundancy attributes .................................................. 163
14.7. Miscellaneous NET device attributes ......................... 165

15. NET lines for communication system .......................... 169
15.1. General ........................................................................ 169
15.2. Basic line attributes .................................................... 172
  15.2.1. Basic definition ..................................................... 172
  15.2.2. Other basic attributes ........................................... 176
  15.2.3. System message handling ..................................... 178
15.3. Data transmission attributes ........................................ 179
15.4. Communication control attributes ............................. 182
15.5. Polling attributes ....................................................... 191
15.6. Autodialling attributes ............................................... 197
15.7. RP570 communication loop attributes ....................... 201
15.8. LON configuration attributes ....................................... 205
15.9. Miscellaneous NET line attributes ............................. 207
  15.9.1. Diagnostic counter ............................................... 207
  15.9.2. Clock synchronization attributes .......................... 210
15.10. DNP V3.00 protocol .................................................. 212
  15.10.1. Link layer attributes .......................................... 212

16. STA objects for communication system ...................... 215
16.1. General ........................................................................ 216
16.2. Common STA attributes ............................................. 217
  16.2.1. Basic attributes .................................................... 217
  16.2.2. Device reservation ................................................ 218
  16.2.3. System message handling ..................................... 219
16.3. STA attributes, ANSI stations ..................................... 223
  16.3.1. Basic attributes .................................................... 223
  16.3.2. Polling attributes .................................................. 224
  16.3.3. Station suspension ............................................... 225
  16.3.4. Diagnostic and counter attributes ......................... 226
  16.3.5. Station communication parameters ....................... 230
  16.3.6. Memory area definitions ....................................... 231
  16.3.7. Message split ..................................................... 237
  16.3.8. Memory access .................................................... 239
  16.3.9. Time synchronization ............................................ 240
  16.4.1. Basic attributes .................................................... 240
16.4.2. Diagnostic counters ...................................................... 242
16.4.3. RTU configuration attributes ........................................... 242
16.4.4. Process communication .................................................. 244
16.4.5. Terminal reports ........................................................... 248
16.4.6. Communication loop attributes ....................................... 249
16.5. STA attributes, SPACOM ....................................................... 250
   16.5.1. Basic attributes ......................................................... 250
   16.5.2. Event handling attributes .......................................... 252
   16.5.3. Diagnostic attributes .................................................. 253
   16.5.4. Station suspension attribute ....................................... 253
   16.5.5. SPA point definitions ................................................ 254
   16.5.6. Miscellaneous SPACOM STA attributes ............................ 258
16.6. STA Objects, P214 RTUs ................................................... 260
   16.6.1. Basic Attributes ......................................................... 260
   16.6.2. Device Reservation ..................................................... 261
   16.6.3. Diagnostics and Suspension ......................................... 262
   16.6.4. System Message Handling .......................................... 262
   16.6.5. Data Communication Attributes .................................... 263
   16.6.6. Priority Control Attributes ........................................ 266
16.7. STA attributes, REX stations ............................................. 267
   16.7.1. Basic attributes ......................................................... 267
   16.7.2. Session handling ....................................................... 268
   16.7.3. Process communication .............................................. 270
   16.7.4. Event handling .......................................................... 273
   16.7.5. Suspension attributes ............................................... 274
   16.7.6. SPA point definition .................................................. 274
   16.7.7. File transfer handling attributes .................................. 275
16.8. STA attributes, LMK stations ............................................ 276
   16.8.1. Basic definition attributes ........................................... 276
   16.8.2. Polling attribute ....................................................... 277
   16.8.3. Process communication .............................................. 277
   16.8.4. Diagnostic attributes ................................................ 278
   16.8.5. LON point definition .................................................. 279
16.9. STA attributes, SPI stations ............................................. 281
   16.9.1. Basic attributes ......................................................... 281
   16.9.2. Configuration attributes ............................................. 281
   16.9.3. Process data communication attributes ............................ 282
   16.9.4. Function Control Attributes ........................................ 291
   16.9.5. Terminal messages ..................................................... 294
   16.9.6. Loop control attributes .............................................. 296
   16.9.7. Redundant line attributes .......................................... 296
16.10. STA attributes, IEC station ............................................ 297
   16.10.1. The application layer ................................................. 297
16.10.2. Data in monitoring direction with IEC 60870-5-101 ..... 298
16.10.3. Data in monitoring direction with IEC 60870-5-103 ..... 298
16.10.4. Data in controlling direction ..................................... 298
16.10.5. Data in controlling direction with IEC 60870-5-101 ..... 299
16.10.6. Data in controlling direction with IEC 60870-5-103 ..... 300
16.10.7. Basic attributes of the application layer ....................... 300
16.10.8. Process communication attributes ............................. 300
16.10.9. Device configuration attributes .................................. 307
16.10.10. Other device handling attributes .............................. 310
16.10.11. Redundant Line Attributes ...................................... 315
16.11. STA attributes, PLC stations ........................................... 315
   16.11.1. Basic attributes ........................................................ 316
   16.11.2. Configuration attributes ........................................ 316
   16.11.3. Communication attributes ..................................... 320
16.12. STA attributes specific to DNP V3.00 ............................. 322
17. PRI objects for communication system ............................... 331
   17.1. General ................................................................. 331
   17.2. PRI attributes ......................................................... 331
      17.2.1. Basic PRI attributes ............................................. 331
      17.2.2. Device reservation ............................................. 333
      17.2.3. System message handling .................................... 334
      17.2.4. Diagnostic counter .............................................. 335
      17.2.5. Printer control ................................................ 336
      17.2.6. Printout properties ............................................. 337
18. Index ............................................................................. 341
1. About this manual

1.1. Copyrights

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

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

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

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

Copyright © 2005 ABB Oy
All rights reserved.

1.2. Trademarks

Registrations and trademarks used in this document include:

Windows: Registered trademark of Microsoft Corporation.
Echelon: Registered trademark of Echelon Corporation Inc.
LonWorks: Registered trademark of Echelon Corporation Inc.

1.3. Related documents

<table>
<thead>
<tr>
<th>Name of the manual</th>
<th>MRS number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Language SCIL</td>
<td>1MRS751849-MEN</td>
</tr>
<tr>
<td>System Configuration</td>
<td>1MRS751846-MEN</td>
</tr>
<tr>
<td>Application Objects</td>
<td>1MRS751848-MEN</td>
</tr>
<tr>
<td>Status Codes</td>
<td>1MRS751850-MEN</td>
</tr>
</tbody>
</table>
1.4. Document revisions

<table>
<thead>
<tr>
<th>Version</th>
<th>Revision number</th>
<th>Date</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8.4.4</td>
<td>15.03.2002</td>
<td>Document created</td>
</tr>
<tr>
<td>B</td>
<td>8.4.4</td>
<td>26.03.2003</td>
<td>Changes and additions in Sections 4.3, 5.4, 5.5, 5.8, 8.2.2., 9.1, 13.2.2.</td>
</tr>
<tr>
<td>C</td>
<td>8.4.5</td>
<td>15.01.2004</td>
<td>OM attribute added in Chapter 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OPC Server attributes added in Section 4.3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Application diagnostic attributes added in Section 5.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Windows NT changed to Windows</td>
</tr>
<tr>
<td>D</td>
<td>9.0</td>
<td>30.06.2004</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>9.1</td>
<td>09.03.2005</td>
<td>Additions, DO_NOT_SYNCHRONIZE_PICTURE_UPDATE, COUPLE_AUDIO_ALARMS_AND_PRINTOUTS, Minor corrections</td>
</tr>
<tr>
<td>F</td>
<td>9.1.5</td>
<td>28.11.2005</td>
<td>OPC stations and nodes</td>
</tr>
</tbody>
</table>
2. **Introduction**

**About this chapter**

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

**System objects**

System objects are programmable units that specify the system configuration and communication of the MicroSCADA 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) (named system objects in earlier MicroSCADA revisions), 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 PC-NET units and communication frontends determine the MicroSCADA configuration. The communication system objects give the communication units an image of the devices connected to them. The base system objects give the MicroSCADA main program an image of the devices and the software used by the base system.

When connecting a device to the MicroSCADA 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.

**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 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.

**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 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, printers. Each process NET unit has its own configuration file containing a SCIL program.
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”

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 Chapter 3.
3. System object handling in SCIL

About this chapter
This chapter describes how to handle the system objects in SCIL.

System object notation format
The system object notation has the following format:

\[ \text{name} : \{ \text{application} \} \{ \text{type} \} \{ \text{attribute} \} \{ \text{index} \} \]

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.

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 MicroSCADA distributed system.

Alternatively, a freely chosen alias name may be given for a system object by specifying its BN attribute, see Chapter 4.

The predefined base system object names are listed and described in Chapter 4 and the communication system objects in Chapter 13.

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 Chapter 6.). A prerequisite for using application number is that the applications recognise 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.
Type
The type is a letter specifying the system object type as follows:

- S Communication system objects
- B Base system objects

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, Chapter 3) 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.

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.

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:SP04 The protocol of line 4 of NET1
- APL3:BAS The application state of application
4. Base system objects, overview

About this chapter

This chapter introduces the base system objects:
4.1. Base system object types. An overview of the base system object types.
4.2. Common naming attributes. Attributes common to all base system object types.
4.3. Defining base system objects. The principles for defining base system objects, the start-up configuration, on-line configuration, attribute access, etc.

4.1. Base system object types

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
INDn Input Devices for semi-graphic workstations, 'n' = 1 ... 100
PRIn Printers. 'n' = 1 ... 20
LINn Links: connections to adjacent nodes. 'n' = 1 ... 20
NODn Nodes: base systems and frontends. 'n' = 1 ... 250
STAn Stations: Remote Terminal Units, PLCs, Protective Equipment, etc. 'n' = 0 ... 5000
STYn Station type defining objects, 'n' = 1 ... 31

The base system recognises 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 .

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 4.2.

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

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 node number of the base system
- Memory spaces and queue lengths
- Properties and addresses of connected devices
Fig. 4.1.-1  *The base system objects defined in the shaded MicroSCADA base system*

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

IND objects
The IND objects define the user-input devices of semi-graphic workstations (MicroWORKSTATION). Only one IND object for a workstation is needed.

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.

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 (RAM, serial line or LAN)
- Communication properties: Redundancy check, time-out length, etc.
- Diagnostic counters, etc.

NOD objects
Each base system and NET unit known to the base system must be defined as a NOD object, also 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 MicroSCADA system.

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.
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.

4.2. 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.

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.

BT Base System Object Type
The type of the object.
Data type: Text keyword
Values: "SYS", "APL", "STA", "STY", "MON", "PRI", "NOD", "LIN" or "IND"
Access: Read-only, configurable

BM Base System Object Number
The object number of the object.
Data type: Integer
Value: 0 ... 5000
Access: Read-only, configurable

Example:
Suppose we want 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:

\[
\begin{align*}
\text{STA5:BBN} & \equiv \text{"PICCADILLY_STATION"} \\
\text{PICCADILLY_STATION:BBT} & \equiv \text{"STA"} \\
\text{PICCADILLY_STATION:BBM} & \equiv 5
\end{align*}
\]

4.3. Defining base system objects

**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.

**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. At least the following definitions must be included in SYS_BASCON.COM:

- A SYS object representing the base system itself. It must always be created as the first base system object in the SYS_BASCON.COM file.
- An APL object corresponding to an application in the base system. At least one application must be given a name (APLn:BNA) and set to HOT (APLn:BAS).
- A MON object connected to the HOT application.

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.

**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 Application Objects), or with the Base System Configuration tool. These on-line changes are not stored, unless they are included in the base system configuration file SYS_BASCON.COM.

**Required definitions**

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

About this chapter

This chapter describes the SYS objects and their attributes:

5.1 General: the definition of SYS objects and the SYS object notation.

5.2 Basic SYS Attributes:

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

5.3 Software Configuration Attributes:

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

5.4 SYS Device Attributes:

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

5.5 Operating System Event Handler Attributes:

- Enable Operating System Event Handler (OE)
- Operating System Event Handler Filter (OT)

5.6 Miscellaneous SYS Attributes: comment (CX) and system variables (SV, UV).

5.1. General

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.

Example:

```plaintext
#CREATE SYS:B = List(-
   SA = 209,-       ;Station address of base system
   ND = 9,-        ;Node number of base system
   DN = 1,-        ;Default NET node number
   DS = "STA",-    ;Default STA type: E.G. STA,RTU,SPA,REX
   DE = 0,-        ;DDE server 0=disabled, 1=enabled
   PC = 6000,-     ;Picture Cache (kB)
   RC = 1000,-     ;Report Cache (kB)
   - ;MS-STOOL Settings
```
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 Chapter 6)

An alternative way of using freely chosen object names is described in Chapter 4.

5.2. Basic SYS attributes

Common naming attributes BN, BT and BM are described in Chapter 4.

The SYS specific attributes are described in the following sections.

5.2.1. Basic configuration attributes

These attributes define the node number and address of the base system. They are obligatory for all base systems.

**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
MicroSCADA Pro
System Objects
Technical Description

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

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

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

5.2.2. Hardware and software information
The base system computer hardware, the operating system or the MicroSCADA kernel settles these attributes. They are set automatically and can not be configured, nor changed.

HW  Hardware
The type of the base system computer.

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

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 4.0 or Windows 2000
       1  Windows XP
       2  Windows Server 2003
Access: Read-only

See the OV attribute (Operating System Minor Version) for more details of the version numbering.
**OS**  **Operating System**
The operating system running in the base system computer.

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

**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
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 MicroSCADA:

<table>
<thead>
<tr>
<th>Version</th>
<th>OV</th>
<th>OM</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT 4.0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>XP</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Server 2003</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

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.

**PR**  **Product Name**
The name of the product can be read from this attribute.

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

**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
MicroSCADA Pro
System Objects
Technical Description

5.2.3. Communication attributes

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

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 Chapter 6.
**DS**  
**Default STA Type**

The default value for the station type, the attribute STAn:BST, if not explicitly defined (see the STA attributes in Chapter 10).

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Text keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&quot;STA&quot;</td>
</tr>
</tbody>
</table>

Stations using the Allen-Bradley ANSI X3.28 protocol

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;RTU&quot;</th>
</tr>
</thead>
</table>

S.P.I.D.E.R. RTUs

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;SIN&quot;</th>
</tr>
</thead>
</table>

Sindac stations using the ADLP80 protocol.

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;SID&quot;</th>
</tr>
</thead>
</table>

Sindac stations using the APLP180 protocol

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;PAC&quot;</th>
</tr>
</thead>
</table>

PAC-5, LED & Keyboard interface

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;SAT&quot;</th>
</tr>
</thead>
</table>

SATTCON Central Station through the COMLI Slave connection

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;RCT&quot;</th>
</tr>
</thead>
</table>

PROCOL Station Interface

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;LCU&quot;</th>
</tr>
</thead>
</table>

Load Control Unit (Load Management)

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;SPA&quot;</th>
</tr>
</thead>
</table>

SPACOM

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;REX&quot;</th>
</tr>
</thead>
</table>

REx type relays

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;IEC&quot;</th>
</tr>
</thead>
</table>

Stations using International Electrotechnical Comission standard protocol

<table>
<thead>
<tr>
<th>Value:</th>
<th>&quot;DNP&quot;</th>
</tr>
</thead>
</table>

Stations using Distributed Network Protocol

Default value:  "STA"
Access: No restrictions

**ER**  
**Enable Routing**

Enabling and disabling message routing. The base system has the capability to route messages to other nodes, see the example in Fig. 5.2.3.1.

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.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>0</td>
</tr>
</tbody>
</table>

Routing disabled

| Value:           | 1             |

Routing enabled

Default value:  0
Access: No restrictions
Fig. 5.2.3.-1 If routing is enabled in base system 1 it can route messages addressed to base system 2

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 time-out can be temporarily and locally changed by the SCIL function TIMEOUT (see the Programming Language SCIL manual, Chapter 8).

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

5.2.4. Time handling attributes

TF Time Format
The format of text representation of dates. The TF attribute affects the time functions TIME, TIMES and DATE and the #SET_TIME command, see the manual "The Programming Language SCIL".

- Data type: Integer
- Value: 0 The format yy-mm-dd hh:mm:ss
- Value: 1 The format dd-mm-yy hh:mm:ss
- Default value: 0
- Access: No restrictions

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)
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.

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 (MicroSCADA 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 MicroSCADA 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 MicroSCADA 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.
### TS  Time Season
The time season - summer time / winter time - of the PC31/32 clock, if such is used for synchronising 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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Text keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>&quot;SUMMER&quot;, &quot;WINTER&quot; or &quot;UNKNOWN&quot;</td>
</tr>
<tr>
<td>Default value</td>
<td>&quot;UNKNOWN&quot;</td>
</tr>
<tr>
<td>Access</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>-12.0 ... +12.0</td>
</tr>
<tr>
<td>Unit</td>
<td>Hours</td>
</tr>
<tr>
<td>Default value</td>
<td>0</td>
</tr>
<tr>
<td>Access</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

### 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.

### 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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Text keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>&quot;NEVER&quot;</td>
</tr>
<tr>
<td>Access</td>
<td>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.</td>
</tr>
</tbody>
</table>
### 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).

"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
MicroSCADA Pro
System Objects

Technical Description

**MF Memory Blocks Free**

The number of free memory blocks in each category of memory block sizes. The size of the blocks in each category is defined by the MS attribute, see below.

- **Data type:** Vector
- **Value:** Vector of 24 integer elements. Each element is the number of free memory blocks in the actual category.
- **Indexing:** The entire vector is always read without index. Individual elements cannot be accessed.
- **Access:** Read-only

**Example:**

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

**MP Memory Pool Sizes**

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

- **Data type:** List
- **Value:** List value with following integer attributes
  - GLOBAL: Size of the global memory pool
  - 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

**MS Memory Block Size**

The size of the memory blocks in bytes. The memory blocks are grouped into 24 categories with different sizes. This attribute contains the size of the memory blocks in each category.

MS is a vector of 24 elements, where each element represents a certain block size. MF (see above) tells how many blocks of each size are free. MU (see below) tells how many blocks of each size are used.

- **Data type:** Vector
- **Value:** Vector of 24 integer elements in the range 8 ... 67 108 864. 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
MU Memory Blocks Used
The number of memory blocks used in each category of memory block sizes, see MS above.

Data type: Vector
Value: Vector of 24 integer elements. Each element is the number of used memory blocks in the actual category.
Indexing: The entire vector is always read without index. Individual elements cannot be accessed.
Access: Read-only

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

PC Picture Cache Size
The maximum memory space allowed for picture cache, that is, pictures and library representations kept in the primary memory. When the occupied space in the picture cache memory (the PU attribute) rises to this value, the oldest elements are dropped out. Pictures that are in use (shown on screen) are not kept in the picture cache.

If the primary memory used by the program is running out, memory space is occasionally borrowed from the picture cache and report cache memories.

If the picture handling is slow, the reason may be too small picture cache size.

Data type: Integer
Value: 0 ... 65 535
Unit: Kilobytes
Default value: MIN (6 000,(Global_Memory_Pool / 4)), described in SCIL syntax
Access: No restrictions

PU Picture Cache Used
The cache memory space used by pictures and library representations in the primary memory. When the PU attribute rises over the PC attribute, the oldest pictures or representations are dropped out from the picture cache.

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

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.
If the memory used by the program is running out, memory space is occasionally borrowed from the picture cache and report cache memories.

If the report database handling is slow, the reason may be too small RC value.

### Data type: Integer
### Value: 0 .. 65 535
### Unit: Kilobytes
### Default value: MIN(3 000,(Global_Memory_Pool / 4)), described in SCIL syntax
### Access: No restrictions

**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

**5.2.6. Global paths and representation libraries**

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

**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 be either a text value defining one directory or a text vector defining one or more directories.

The attribute can only be set by the "#CREATE SYS:B" command in SYS_BASCON.COM.

**Example:**

```plaintext
#CREATE SYS:B=LIST(PH=LIST(PICG="/LAN/TEST/PICG","/LAN/ACTIVE/PICG"),-
TOOLS = "D:\VS_TOOLS"),-
```

**Representation Libraries (RL)**

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 by the "#CREATE SYS:B" command in SYS_BASCON.COM.

**Software configuration attributes**

**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 Chapter 6.

**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

**DDE Server attributes**

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

**DE DDE Server Enabled**

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

- Data type: Integer
- Value: 0 Disabled
Example:

Enabling the SYS:BDE attribute in SYS_BASCON.COM

#CREATE SYS:B = LIST(-
SA = 209,- ;STATION ADDRESS OF BASE SYSTEM
ND = 9,-  ;NODE NUMBER OF BASE SYSTEM
DN = 1,-  ;DEFAULT NET NODE NUMBER
SV = %SV,- ;PRINTER SETUP
DS = "RTU",- ;STA TYPES: STA,RTU,PCL,SIN
DE = 1)   ;DDE SERVER ENABLED

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 after MicroSCADA started
3   Cumulative value of successful request transactions after MicroSCADA started
4   Cumulative value of successful poke transactions after MicroSCADA started
5   Cumulative value of successful execute transactions after MicroSCADA started
Access: No restrictions

DU  DDE Server in Use

The status of use of the DDE server. The value is 0 if DDE server is not started. If the DDE server is started, the value is 1 while someone is logged in. If there is nobody logged in, the value is 0.

Data type: Integer
Value: 0  Not in use
       1  In use
Access: Read-only
5.3.3. OPC Server attributes

The OPC Server attributes are used to start and stop the MicroSCADA OPC Server and to configure its name space.

**OPC Server**
The state of the MicroSCADA OPC 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 MicroSCADA OPC Server. If set by the #CREATE SYS:B command (in SYS_BASCON.COM), the OPC Server is started immediately after the application(s) has 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.

**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.

5.4. 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.

5.4.1. Audio alarm device

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

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

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

5.4.2. External clock

CA  External Clock Address
The use of external clock (radio clock), see the CL attribute.
Data type: Integer
Value: 0 = No external clock
Any other value = External clock is in use

Default value: 0

Access: No restrictions

**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
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

**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 physical clock of an integrated frontend is synchronized with the same frequency, whether there is an external clock or not. 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

**CL External Clock**

The type of the external clock, if any. The external clock synchronizes the MicroSCADA 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
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 synchronising 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:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>'1'</th>
<th>'0'</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0</td>
<td>Clock state</td>
<td>Free running</td>
<td>DCF77 controlled</td>
</tr>
<tr>
<td>D1</td>
<td>Daylight saving</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>D2</td>
<td>Sync'ed since reset</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Dayl.sav. is going to change</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>D4</td>
<td>UTC time</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>D5</td>
<td>Leap second announced</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>D6</td>
<td>Board time from</td>
<td>Serial iface</td>
<td>DCF77</td>
</tr>
<tr>
<td>D7</td>
<td>On-board time invalid</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

5.4.3. SPA device attributes

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

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

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
5.5. Operating system event handler attributes

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

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

OE Enable Operating System Event Handler

Enabling the Operating System Event Handler.

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

OT Operating System Event Handler Filter

Information about which event types is passed on to MicroSCADA. 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
```
5.6. **Miscellaneous attributes**

**CX**  
**Comment Text**  
A freely chosen comment text.

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

**SV**  
**System Variables**  
The attribute can be used as global variable. It is reserved for ABB and should not be used in application programs.

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

**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:

```plaintext
*SET SYS:BUV1 = SYS:BUV1 + 1
```
6. APL objects for base system

About this chapter

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

6.1 General: APL Objects and APL Object Notation.
6.2 Basic APL attributes: Common naming attributes (BN, BT, BM), Basic Configuration (AN, AS, NA, TT), Supervision Configuration (ME, QE), Paths, Representation Libraries and Text Databases (PH, RL, TD), External and Alias Applications (ND, TN).
6.3 Mapping attributes: Application Mapping (AP) and Device Mapping Attributes (MO, PR, ST).
6.4 Shadowing attributes (SC, SD, SF, SI, SL, SN, SP, SQ, SR, SS, SW, SY).
6.5 Resource handling attributes (AA, AU, EM, EU, HB, PM, PQ, PU, RO, QD, QL, QM, QO, QP, QU).
6.6 Application control attributes: Functional Definitions (AQ, EE, HP, PP), User Interface definitions (CP, LA, MS), Operation control (AC, AT, HT, UC), SCIL Loop Control Attributes (PS, RS), Revision Compatibility (RC).
   Application diagnostic attributes (DI, DT, DS).
6.7 Mirroring attributes: Configuration attributes (EP, HE, IE, IS), Diagnostic attributes (HD, ID).
6.8 Miscellaneous APL attributes: Comment (CM), Global Variables (SV, UV).

6.1 General

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.

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 = 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:

\[ \text{APL}_n:m\text{Bat} \]

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 Chapter 4.

6.2. Basic APL attributes

Common naming attributes BN, BT and BM are described in Chapter 4. The APL specific attributes are described in the following sections.

6.2.1. 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.

**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, Chapter 8).

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

**Example:**

The object number of the current application is shown:

!SHOW AN APL:BAN

**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 shut-down 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.
MicroSCADA Pro
System Objects
Technical Description

"WARM"  The application is not running, but the databases are loaded and accessible.
"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"

All local applications should be set to "COLD" before system shut-down. When an application is set to "HOT", the event channels APL_INIT_1 and APL_INIT_2 (Chapter 9) are activated.

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 characters
Access: Read, conditional write

TT  Translation Type
Determines how the base system regards the application and where the application is found, see example in Fig. 6.2.4.-1.

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.

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"
6.2.2. Supervision configuration attributes

**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).

**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).

6.2.3. Paths, representation libraries and text databases

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

**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 only be set by the "#CREATE APL:B" command or the "#SET APLn:BPH" command when the application is cold and its shadowing state is "NONE".
**MicroSCADA Pro**  
**System Objects**  
**Technical Description**

### 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 only be set by the "#CREATE APL:B" command or the "#SET APLn:BRL" command when the application is cold and its shadowing state is "NONE".

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

### 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:** Read, conditional write

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 only be set by the "#CREATE APL:B" command or the "#SET APLn:BTD" command when the application is cold and its shadowing state is "NONE".

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

### 6.2.4. 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").
Fig. 6.2.4.-1 Communication between applications that are in different base systems

**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 Fig. 6.2.4.-1 and the TN attribute below.

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

**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 6.3.), 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")
   ```

6.3. Mapping attributes

6.3.1. Application mapping

Application mapping is required if you wish the application to communicate with other applications, in the same or other base systems.

**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

The application recognises other applications by 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 Fig. 6.3.1.-1.

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 Fig. 6.3.1.-1.

```plaintext
APL:BAP5 = 2
```
6.3.2. 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 Fig. 6.3.2.-1. 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, fixed monitor number mapping is required only for semi-graphic monitors, such as MicroWorkstations. In other cases, value -1 is normally used instead of the physical device number, indicating that any physical monitor number will do.
MicroSCADA Pro
System Objects
Technical Description

Fig. 6.3.2.-1 An illustration of the station mapping attribute (ST). The PR attribute works in a similar way.

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

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 MicroSCADA monitors will be available for the application, and which are the logical and physical numbers of monitors that the application may use.

Only semi-graphic (MicroWorkstation) monitors require a fixed monitor mapping. When a VS or an X monitor is opened, any free physical monitor number is allocated for it. Physical monitor number -1 is configured in MO for such monitors. 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 logical monitor numbers 1 and 2 for MicroWorkstations (mapped to physical monitors 11 and 12) and numbers 3 to 7 for VS (or X) monitors:

```
#CREATE APL1:B = LIST(-
 ............
 MO = (11,12,-1,-1,-1,-1,-1), -
 ............)
```

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.

ST          Station Mapping
Mapping of logical station numbers to physical station numbers.

Data type: Vector
Value: Up to 5001 integers in the range 0 ... 5000. Physical station numbers
Indexing: Logical station number, 0 ... 5000
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 Fig. 6.3.2.-2.

Because one-to-one mapping is the default, ST attribute does not always have to be specified.
6.4. 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 the run-time only changed data items are shadowed.

**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. The SYS:BSH and APLn:BSS attributes do not get the shadowing values.
The attribute is also the maximum time allowed for file destruction in the receiving application.

**SD 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 BYTES
5. RECEIVED MESSAGES
6. RECEIVED COMMANDS
7. RECEIVED TRANSACTIONS
8. RECEIVED BYTES
9. RAM DUMP TIME
10. FILE DUMP TIME

**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.
MicroSCADA Pro
System Objects
Technical Description

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td><strong>Shadowing Diagnostic Interval</strong>&lt;br&gt;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).&lt;br&gt;Data type: Integer&lt;br&gt;Value: 0 ... 65 535&lt;br&gt;Unit: Milliseconds&lt;br&gt;Default: The value of the SF attribute&lt;br&gt;Access: No restrictions</td>
</tr>
<tr>
<td>SL</td>
<td><strong>Shadow Dump Slowdown</strong>&lt;br&gt;Shadowing dump phase control. In some applications, the file dump phase of shadowing startup 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.&lt;br&gt;Data type: Integer&lt;br&gt;Value: 0 ... 65 535&lt;br&gt;The slow down time in milliseconds per kilobytes of transferred data&lt;br&gt;Default: 0&lt;br&gt;Access: No restrictions&lt;br&gt;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.</td>
</tr>
<tr>
<td>SN</td>
<td><strong>Shadowing Number</strong>&lt;br&gt;The logical application number of the shadowing partner application (usually an external application).&lt;br&gt;Data type: Integer&lt;br&gt;Value: 0 ... 250&lt;br&gt;Logical application number&lt;br&gt;Access: Read-only, configurable&lt;br&gt;Example: See the example of the SS attribute below&lt;br&gt;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.</td>
</tr>
</tbody>
</table>
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.

**SP**  
**Shadowing Phase**

The shadowing phase in progress.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Text keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td></td>
</tr>
<tr>
<td>&quot;NONE&quot;</td>
<td>No shadowing</td>
</tr>
<tr>
<td>&quot;HOT_RC&quot;</td>
<td>Normal operation of the stand-by application in Hot Stand-by systems</td>
</tr>
<tr>
<td>&quot;HOT_SD&quot;</td>
<td>Normal operation of the primary application in hot stand-by systems</td>
</tr>
<tr>
<td>&quot;TO_HOT_SD&quot;</td>
<td>On-going file dump or RAM dump (primary application)</td>
</tr>
<tr>
<td>&quot;TO_HOT_RC&quot;</td>
<td>On-going file dump or RAM dump (stand-by application)</td>
</tr>
<tr>
<td>&quot;WARM_RC&quot;</td>
<td>Normal operation of the stand-by application in warm stand-by systems (file shadowing)</td>
</tr>
<tr>
<td>&quot;WARM_SD&quot;</td>
<td>Normal operation of the primary application in warm stand-by systems</td>
</tr>
<tr>
<td>&quot;TO_WARM_RC&quot;</td>
<td>On-going file dump or RAM dump (stand-by application) in warm stand-by systems</td>
</tr>
<tr>
<td>&quot;TO_WARM_SD&quot;</td>
<td>On-going file dump or RAM dump (primary application) in warm stand-by systems</td>
</tr>
</tbody>
</table>

Access: Read-only

**SQ**  
**Shadow the Event Channel Queue**

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

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Event channel shadowing disabled</td>
</tr>
<tr>
<td>1</td>
<td>Event channel shadowing enabled</td>
</tr>
</tbody>
</table>

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).

**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.

**SS Shadowing State**

The role of the application in the shadowing relation.

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

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

**SW  Shadowing Watchdog**
The logical application number of the watchdog application.

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

**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 |

### 6.5. Resource handling attributes

**AA  APL-APL Server Count**
The number of APL-APL communication servers that handles the APL-APL communication.

| Data type: | Integer |
| Value:     | 0 ... 10 |
| 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 time-outs 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.

**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.

**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 communication frontend (NET). The behavior is specified for not losing events in any circumstances (However, if the event buffers in NET 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 6.7.

**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
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

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. MicroSCADA printer despooler (PRNC) also supervises the length of the print queue of the connected NT printer and waits while the length is more than 100 jobs.

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

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
MicroSCADA Pro
System Objects
Technical Description

**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.

**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.
QL  Process Query Length

Maximum length of process database query performed with the function PROD_QUERY, see the Programming Language SCIL manual, Chapter 9.

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

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 MicroSCADA 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 maximum (QM(2)) is not honoured, because otherwise events would be lost. When the execution of the primary object of the event channel is finished, 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 command procedures (the primary object of the event channel plus 4 secondaries) and the value of the EM attribute is 500, the length of the event channel queue (QU(2)) is likely to be about 2500 in a rush situation, regardless of the value of QM(2).
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:

```plaintext
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.

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

RO    Running Objects
The names of the data objects or command procedures currently under execution in
REPR queues (c.f. attribute QO).
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.

### 6.6. Application control attributes

#### 6.6.1. Functional definitions

The following attributes specify some functions of the application.

**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 MicroSCADA monitor basis. An alarm picture is not removed from the queue until it has been displayed on the MicroSCADA monitor. This is the default.
  - "BY_APL": The queue is maintained on application basis. When an alarm picture is displayed on a MicroSCADA monitor, it is removed from the alarm picture queues of all MicroSCADA 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

**EE**  
**System Event Enabled**

The state of the system event channel SYS_EVENT:A – is it in use or not. If the system event channel is in use, and if it exists, it is activated in the application at the events described in Chapter 9 "Predefined Event Channels" of the Application Objects manual.

- **Data type:** Integer
- **Value:**
  - 0: SYS_EVENT:A is not in use
  - 1: SYS_EVENT:A is in use
- **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.

**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.

**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 tells when the status change is considered to fulfill the citerion NEW VALUE.

Data type: Text keyword
Value:  
- "NEVER" Never
- "WHEN_SET_TO_2" When OS is set to 2 (but not vice versa)
- "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
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 Chapter 10) 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 considered.

### 6.6.2. User interface related attributes

The following attributes affect the user interface of the application.

**CP**  
**Color Allocation Policy**

The default color allocation policy used by the MicroSCADA monitors of the application.

- **Data type:** Text keyword
- **Value:** "NONE" (= undefined), "PRIVATE" or "SHARED"
- **Default:** "PRIVATE"
- **Access:** No restrictions

The .COLOR command allocates an entry from the color palette of the MicroSCADA monitor. If there are several MicroSCADA windows (or other applications requiring many colors) open on a MicroSCADA monitor, it may happen that the palette runs out of entries (there are normally 256 slots available). To avoid this from happening, the color allocation policy can be defined as "SHARED". The other option is "PRIVATE".

Private colors allocate a dedicated palette entry for the color, shared colors do not. Therefore, a change to a private color appears immediately on screen, that is, all the pixels drawn with the color are immediately affected. Changes to a shared color, on the other hand, affect only pixels drawn with the color after the change. As a summary, private colors are needed for color animation (done for example in current color chooser pictures).

The color allocation policy for individual MicroSCADA monitors can be changed by the monitor attribute CP. The color allocation policy for individual color definitions can be changed by the .COLOR command.

**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.

**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.

### 6.6.3. Operation control attributes

The following attributes keep track of events in the application. They are useful in application programs.

**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

**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.

**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 display so that a change of the attribute causes an updating of the display.

**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

### 6.6.4. SCIL loop control attributes
The following attributes enable a stop of running loops in SCIL programs (the #LOOP command).

**PS Printer Spool Stop**
Stopping the printer spool. Setting this attribute to one (1) stops a running loop in the picture that is printed at the moment.

- **Data type:** Vector
- **Value:** A vector of two integers
- **Indexing:**
  - 1 Process printout
  - 2 Report printout
- **Access:** Read, conditional write

**RS Report Task Stop**
Stopping the report task. Setting the value of this attribute to one (1) stops a running loop in a command procedure under execution.

- **Data type:** Vector
- **Value:** Vector of 32 integers (1)
- **Indexing:** The number of the process that executes the loop.
  - See Fig. 6.3.2.-1.
- **Access:** Read, conditional write
**Revision compatibility**

**RC**  
Revision Compatibility

Allowing the application engineer to choose the behaviour of SCIL language elements when upgrading to a new MicroSCADA 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 MicroSCADA 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"
- "KEEP_FILE_VERSION_1_DATABASE_FILES"
- "DEFAULT_DAYLIGHT_POLICY_ISCALENDAR"
- "844_COMPATIBLE_MIRRORING"
- "CREATE_VERSION_2_SCIL_DATABASES"
- "DO_NOT_SYNCHRONIZE_PICTURE_UPDATE"
- "COUPLE_AUDIO_ALARMS_AND_PRINTOUTS"
- "ALLOW_CONFLICTING_F_ATTRIBUTE_NAMES"

**Access:** Read-only, configurable

**Example:**

The application APL1 uses the compatibility issue ON_COMMAND_EXPANSION:

```scil
#CREATE APL1:B = LIST{
```

69
Compatibility issues:

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 behaviour in case of a single line #ON command, for example:

```scil
@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:

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

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.

The base system configuration file (SYS_BASCON.COM) delivered by the supplier has this compatibility feature ON.

If a tool or a SCIL program should be independent of the RC attribute value of the application, use the REVISION_COMPATIBILITY function to temporarily disable the compatibility issue. Then reset the issue using the same function, see the Programming Language SCIL manual, Chapter 9.

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.

**NO_QUALITY_ATTRIBUTE_SEMANTICS**

In MicroSCADA revision 8.4.2 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 behaviour of the process object in any way.

In newer MicroSCADA 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 pseudoattribute. 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.

**NO_ALIAS_CHECKING**

In 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 “Installation” manual, chapter “Upgrading from Previous revisions”.

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.
NO_ALARM_BY_OR_AND_OF

In 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 behaviour.

DONT_RECALCULATE_AL_AFTER_ALARM_BLOCKING

In 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 behaviour.

CREATE_VERSION_1_FILES

In revision 8.4.4 of MicroSCADA, the implementation of keyed files is 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 manual “Programming Language SCIL.”.

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.

KEEP_FILE_VERSION_1_DATABASE_FILES

In revision 8.4.4 of MicroSCADA, the implementation of keyed files is 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.
In revision 8.4.4 of MicroSCADA, the implementation of time handling is 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 behaviour 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 behaviour 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 behaviour, see manual “Application Objects”.

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 behaviour of new time channels, however.

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 MicroSCADA 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.
CREATE_VERSION_2_SCIL_DATABASES
The internal implementation of SCIL databases (SDB) has been optimized for faster access in MicroSCADA 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 MicroSCADA 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.

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.

COUPLE_AUDIO_ALARMS_AND_PRINTOUTS
Generation of audio alarms has been changed in MicroSCADA 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.

ALLOW_CONFLICTING_F_ATTRIBUTE_NAMES
When an old (Rev. 8.4.3 or older) application was upgraded to 8.4.5 or newer, 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.
### 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.

#### 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

#### 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:

```plaintext
#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
```
**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 ... 65535
    - 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

---

**6.7. Mirroring attributes**

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

---

**6.7.1. Mirroring configuration**

**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. Actually, it 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.

**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.

**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.

**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.

6.7.2. Mirroring diagnostics

**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:
  - 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

**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
6.8. Miscellaneous APL attributes

**CX** Comment Text

A freely chosen comment text.

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

**SV** System Variables

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

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

**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
7. MON and IND objects for base system

About this chapter
This chapter describes the base system objects related to the operator workstations:

7.1 The MON objects and their attributes:
- General.
- Basic monitor definition attributes (BN, BT, BM, DT, TT)
- Informative monitor attributes (AN, AP, DC, LI, SD, SG, SZ)
- Monitor control attribute (BP, CP, DI, ED, IL, LA, MS, PC, WC)
- Miscellaneous monitor attributes: comment (CM) and global variable attributes (SV, UV)

7.2 The IND objects and their attributes (BN, BT, BM, CX, DC, DT, MO, TT).

7.1. MON objects

7.1.1. General

MON objects
The MON objects correspond to the MicroSCADA monitors opened on screens by the operator or automatically. A screen - the base system screen or a workstation - can contain one or more MicroSCADA monitors connected to the same or different applications in the same or in different base systems. Semi-graphic workstation correspond to one MON object.

Each MicroSCADA monitor which will be used by a base system and its applications must be defined as a MON object. The MicroSCADA monitors are reserved by an application with the monitor mapping attribute (APLn:BMO, Chapter 7.).

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

The MON attributes of printers 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, Chapter 6)
'n' The object number of the MicroSCADA monitor in the other base system

An alternative way of using freely chosen object names is described in Chapter 4.

7.1.2. Basic MicroSCADA monitor definition attributes

Common naming attributes BN, BT and BM are described in Chapter 4.
The MON specific attributes are described in this and next sections.
The following attributes must be defined for all MON objects:

**DT**  
Device Type

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

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Text keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&quot;VS&quot; MicroSCADA monitor supporting the display of Visual SCIL dialogs. Motif widgets are not supported.</td>
</tr>
<tr>
<td></td>
<td>&quot;X&quot; Full graphics MicroSCADA monitor using X and Motif software. Visual SCIL is not supported</td>
</tr>
<tr>
<td></td>
<td>&quot;MW&quot; Semi-graphic workstation</td>
</tr>
<tr>
<td></td>
<td>&quot;NONE&quot; MicroSCADA monitor is not defined</td>
</tr>
<tr>
<td>Default value:</td>
<td>&quot;VS&quot;</td>
</tr>
<tr>
<td>Suggested value:</td>
<td>Use the &quot;VS&quot; type unless there is some reason for using another type. The &quot;X&quot; type must be used for monitors that should be able to display Motif widgets.</td>
</tr>
<tr>
<td>Access:</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>

**TT**  
Translation Type

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

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Text keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&quot;NONE&quot; Off, out of operation</td>
</tr>
<tr>
<td></td>
<td>&quot;LOCAL&quot; Normal operating state</td>
</tr>
<tr>
<td>Default value:</td>
<td>&quot;NONE&quot;</td>
</tr>
<tr>
<td>Access:</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>
## Informative MON attributes

### AN  Application Monitor Number

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

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

### AP  Application Number

The number of the application that controls the MicroSCADA monitor according to the monitor mapping (the attribute APln:BMO, Chapter 6).

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

### DC  Device Connection

The type of the MicroSCADA monitor connection. The attribute is mainly informative and need not be set.

- **Data type:** Text keyword
- **Value:**
  - "LINE"  Serial port or LAN connection
  - "RAM"  Direct connection to the base system computer
  - "NET"  Connected to a NET unit
- **Default value:** The default value depends on the device type (see the DT attribute) as follows:
  - "MW"  "NET"
  - "X" or "VS"  "RAM"
- **Access:** Read, conditional write

### LI  Logged In

Informs whether a user has logged in on the MicroSCADA monitor. When a user logs in or out, an event channel named MON_EVENT (if it exists) is started in the application. See Application Objects, Chapter 9.

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

**System Objects**

**Technical Description**

---

**SD**  
**System Device Name**

The identifier of the system device. This attribute is set automatically when the operator opens a MicroSCADA monitor of type VS or X.

- **Data type:** Text
- **Value:** Node name and display number. When the operator opens a MicroSCADA monitor of type VS or X, the SD attribute gets a value as follows:
  - "" Local VS monitor, not using Exceed
  - "Id:0.0" Remote VS or X monitors, where id is node name or ip address. Requires Exceed.
  - "localhost:0.0" Remote monitor (VS or X) on local PC. Requires Exceed.
- **Default value:** No
- **Access:** Read-only

---

**SG**  
**Semi-graphic**

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

- **Data type:** Integer
- **Value:**
  - 0 Full-graphic MicroSCADA monitor
  - 1 Semi-graphic MicroSCADA monitor
- **Access:** Read-only

---

**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

---

**7.1.4. MicroSCADA monitor control attribute**

**BP**  
**Blink Policy**

Specifies how to handle the blink behaviour 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)
MicroSCADA Pro
System Objects
Technical Description

CP Color Allocation Policy
The color allocation policy used by the MicroSCADA monitor.

Data type: Text keyword
Value: "NONE", "PRIVATE" or "SHARED". MicroSCADA monitors of type "VS" cannot be "PRIVATE".
Default value: The value of the APL:BCP attribute of the application that uses the MicroSCADA monitor, or "SHARED" if APL:BCP = "NONE".
Suggested value: The MicroSCADA monitor object must be defined as "SHARED" if the graphics board of the screen supports more than 256 colors.
Access: No restrictions

See the CP attribute of APL object in Chapter 6. The value given with this attribute overrides the value given with the CP attribute of the APL object. The policy defined by this attribute can be temporarily overridden by definitions given with the SCIL command .COLOR.

DI Diagnostic Interval
Monitor diagnostic interval.

Data type: Integer
Value: 0 ... 65 535
0 = No diagnostics
Unit: Seconds
Default value: 10
Access: No restrictions

For X monitors the time for cyclical diagnostic commands to the X server and the TCP/IP connection can be set. When the connection to a X monitor is lost the error message "X Monitor n IO Error Event:" is printed. The communications with VS remote monitors is diagnosed with the same attribute. The attribute is the time between the diagnostic commands.

ED Enter Key Disabled
In MicroSCADA 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
**IL**  **Input Locked**
Prevents the input to a MicroSCADA monitor from keyboard or mouse.

- **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

Warning: When a monitor session is closed, the value of this attribute is restored to the value it had at the beginning of the session. This attribute has no effect on MicroWorkstations.

**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.

**MS**  **Monitor Stop**
Setting this attribute to 1 stops a running loop in the MicroSCADA monitor.

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

**PC**  **Picture Containers**
The maximum number of picture containers in a Visual SCIL MicroSCADA monitor.

- **Data type:** Integer
- **Value:** 0 ... 10
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.

**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, Chapter 10.
- **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:**

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

### 7.1.5. Miscellaneous MON attributes

**CX Comment Text**

A freely chosen comment text.

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

**SV System Variables**

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

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

7.2. IND objects

7.2.1. General

Each semi-graphic workstation, MicroWORKSTATION, to be used by the base system and its applications must be defined as an IND object (‘input device’). The IND objects are connected to a MON object (the MON object corresponding to the workstation) with the INDn:BMO attribute.

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

\[ \text{INDn:Bat} \]

where

'n' The object number of the input device. 'n' may not be omitted from the object notation.

'at' The attribute name

The IND attributes of input devices defined in another base system are accessed with the following object notation:

\[ \text{INDn:mBat} \]

where

'm' The logical application number of an external application (see the TT attribute, Chapter 6)

'n' The object number of the input device in the other base system

An alternative way of using freely chosen object names is described in Chapter 4.

7.2.2. IND attributes

Common naming attributes BN, BT and BM are described in Chapter 4.

The IND specific attributes are described below.

CX | Comment Text
---|---
A freely chosen comment text.

Data type: Text
Value: Any text, up to 255 characters
Access: No restrictions
MicroSCADA Pro
System Objects
Technical Description

DC  Device Connection
The type of the device connection. The attribute is mainly informative and needs not
to be set.
Data type:  Text keyword
Value:  "LINE"  LAN connection
"NET"  Connected to a NET unit
Default value:  "NET"
Access:  Read, conditional write

DT  Device Type
The type of the IND object.
Data type:  Text keyword
Value:  "NONE"  Undefined
"MW"  Semi-graphic workstation
Default value:  "MW"
Access:  Read, conditional write

MO  Monitor Number
The object number of the connected MicroSCADA monitor, that is the MON object
corresponding to the workstation.
Data type:  Integer
Value:  0 .. 100, MONn:B object number. Value 0 means that the input
device is not connected to any MicroSCADA monitor.
Default value:  0
Access:  No restrictions

TT  Translation Type
The operating state of the input device.
Data type:  Text keyword
Value:  "NONE"  Off, out of use
"LOCAL"  Normal operating state
Default value:  "NONE"
Access:  No restrictions
8. LIN objects for base system

About this chapter
This chapter describes the LIN objects and their attributes:
8.1 General: The link types and the LIN object notation.
8.2 LIN Attributes:
- Common naming attributes (BN, BT, BM)
- Basic LIN Definition Attributes (LT, TR, SC, SD)
- RAM Link Attributes (EN, NA, RE, TI)
- Diagnostic Counters (DC)
- Miscellaneous LIN attributes (CM)

8.1. General
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 frontends
- One or two RAM links to internal NET units
- One or more integrated links to PC-NET units

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, Chapter 14.)

'n' The object number of the link in the other base system

An alternative way of using freely chosen object names is described in Chapter 4.
8.2. LIN attributes

Common naming attributes BN, BT and BM are described in Chapter 4. The LIN specific attributes are described in the following sections.

8.2.1. Basic LIN definition attributes

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

| Data type: | Text keyword |
| Value:     | "NONE"       |
|           | "RAM"        |
|           | "LAN"        |
|           | "INTEGRATED" |

Undefined  Common RAM  LAN (TCP/IP)  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).

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:     | "TCP/IP" or "NONE" |

Default values: "TCP/IP"
Access: Read, conditional write

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: "INTNET"
Access: Read, conditional write
MicroSCADA Pro
System Objects
Technical Description

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_conf.txt"
; default configuration file \sc\sys\active\sys\pc_net_conf.txt used
#SET LIN3:BSC="\sc\prog\exec\pc_nets.exe"

SD System Device Name

The name of the RAM device ("RM00" or "RM01") when LT = "RAM". See the RAM driver installation instructions in the Installation manual.

Data type: Text keyword
Value: Device name, "RM00" or "RM01"
Default value: "RM00"
Access: Read, conditional write

8.2.2. RAM link attributes

The following attributes applies to RAM connections (LT = "RAM").

EN ENQ Limit

The maximum number of ENQs (response requests) that the base system will send. If the base system does not receive response when this limit is reached, an error status is produced.

Data type: Integer
Value: 0 ... 65535
Default value: 3
Suggested value: 3
Access: No restrictions

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 ... 65535
Default value: 3
Suggested value: 3
Access: No restrictions

RE Redundancy

The type of checksum added to each message.

Data type: Text keyword
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

8.2.3. Diagnostic counters

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 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
   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. Does not concern RAM links.

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 received from the line. Does not concern RAM links.

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. PARITY ERRORS
A counter that is incremented when a received message is rejected because of a parity error.

13. OVERRUN ERRORS
A counter that is incremented when a received message is rejected because of an overrun error.

14. CHECK SUM ERRORS
A counter that is incremented when a received message is rejected because of a discrepancy in the checksum (BCC or CRC).

15. FRAMING ERRORS
A counter that is incremented when a received message is rejected because of a framing error.

16. BUFFER OVERFLOW ERRORS
A counter that is incremented when a received message is longer than 259 bytes and therefore does not fit into the message buffer.

8.2.4. Miscellaneous LIN attributes

CX Comment Text
A freely chosen comment text.

Data type: Text
Value: Text, up to 255 characters
Access: No restrictions
9. NOD objects for base system

About this chapter
This chapter details the NOD objects and their attributes. It contains two sections:

9.1 General: The meaning of the NOD objects and the NOD object notation.

9.2 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), and Miscellaneous NOD attributes (CM, OP).

9.1. General

NOD objects
The NOD objects represent nodes in the MicroSCADA 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 units and DCP-NET units) which will communicate with the base system in question, directly or indirectly.
• Frontends (communication frontends) if their attributes (see Chapter 14) should be accessed from the base system.
• Gateways (protocol converters) of various types.
• 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 Chapter 14).

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

\[
\text{NODn:Bat}
\]

where

'\text{n}'  NOD object number 'n' can be 1 ... 250. 'n' may not be omitted from the object notation.

'\text{at}'  Attribute name

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

\[
\text{NODn:mBat}
\]

where

'\text{m}'  The logical application number of an external application
NOD attributes

The common naming attributes BN, BT and BM are described in Chapter 4.

The NOD specific attributes are described in the following sections.

### 9.2.1. Basic node definition attributes

#### LI Link Number

The number of the LIN object that defines the link along which the node is reached. See Chapter 8.

- **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.

#### 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.

#### NT Node Type

The type of the node.

- **Data type:** Text keyword
- **Value:**
  - "SYS" Base system
  - "NET" NET unit (PC-NET unit or DCP-NET unit)
  - "MFL" Communication frontend
  - "MW" Semi-graphic workstation
  - "GATEWAY" Gateway
  - "OPC_AE" OPC Alarm and Event Server
  - "UNKNOWN" Unknown
MicroSCADA Pro
System Objects
Technical Description

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 informational.

For OPC server nodes, the attribute must be set to "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_AE".

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_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

Example:
In the case shown in the Fig. 9.2.1.-1, define the RN attribute of the base system
NOD as the node number of the frontend (MFL node) or NET unit. The messages to
the base system are addressed to the frontend or NET unit. The messages are then
forwarded to the base system.
**SA  Station Address**

The station address of the node used by the MicroSCADA 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 MicroSCADA 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.

**Node diagnostic attributes**

**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 startup.

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

**DI  Diagnostic Interval**

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

- **Data type:** Integer
- **Value:** 0 ... 65 535
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 ... 65535
Unit: Seconds
Default value: 60 seconds
Access: No restrictions

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

9.2.2. Node communication attributes

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 Chapter 14.

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

Example:

#SET NET1:SLT = NOD1:BLT

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.
9.2.3. Miscellaneous NOD attributes

**CX** | **Comment Text**
---|---
A freely chosen comment text.

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

**OP** | **OPC Server**
---|---
OPC server configuration.

Data type: List  
Value: The following attributes:

- **CI** Text, the class id (CLSID) of the OPC server in the standard Windows format, e.g. 
  "\{CE0322A9-65A9-4268-84D5-DD7A17E94C56\}"  
- **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.
  - **AA** Auto acknowledge, integer 0 or 1. If AA is set to 1, acknowledging the MicroSCADA 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

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_AE").
10. STA and STY objects for base system

About this chapter
This chapter describes the base system objects related to stations (process units):

10.1 The STA objects and their attributes (AE, BN, BT, BM, CX, HS, IS, LP, MR, ND, PP, ST, TN, TB, TR, TT).
10.2 The STY objects and their attributes (BN, BT, BM, CT, CX, LP, NA, DB).

10.1. STA objects

10.1.1. General

STA objects

Each station (RTU, PLC, SRI, central station, etc.) which 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® networks need not be defined as STA objects in the base systems.

STA object notation

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

\[ \text{STAn:Bat} \]

where

'n' \(0 \ldots 5000\). '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:

\[ \text{STAn:mBat} \]

where

'm' The logical application number of an external application (see the TT attribute in Chapter 6)

'n' The object number of the station in the other base system

An alternative way of using freely chosen object names is described in Chapter 4.

10.1.2. STA attributes

Common naming attributes BN, BT and BM are described in Chapter 4.

The STA specific attributes are described below.

1. LONWORKS is a trademark of Echelon Corporation registered in the United States and other countries.
For further information about the mirroring concept (attributes AE, HS, IS and MR), see the System Configuration manual.

**AE  Analog Events**

The addresses of the analog input objects that are considered as event objects in mirroring communication.

Data type: Integer 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.
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 of this attribute is in use only when the mirroring role (MR) of the station is "HOST" or "BOTH".
2. The value may be set only as a whole. Single vector elements may not be set.
3. 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").

**CX  Comment Text**

A freely chosen comment text.

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

**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_TIMEStamped_ANALOGS"  The analog objects that are not time-stamped by the station are subject to 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.

The value of this attribute has a meaning only when the mirroring role (MR) of the station is "HOST" or "BOTH".

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 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.

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

The node number of the NET unit is the same as the NET system object number.
If the station type of the station is "OAE", the ND attribute is the node number of the node object that specifies the OPC server of the station (see Chapter 9).
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.

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 tells when the status change is considered to fulfill the criterion NEW VALUE.

Data type: Text keyword
Value: "NEVER" Never
**MicroSCADA Pro**  
**System Objects**  
**Technical Description**

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;WHEN_SET_TO_2&quot;</td>
<td>When OS is set to 2 (but not vice versa)</td>
</tr>
<tr>
<td>&quot;WHEN_SET&quot;</td>
<td>When OS is set from 0 to 2 or vice versa</td>
</tr>
<tr>
<td>&quot;ALWAYS&quot;</td>
<td>When OS changes from 0 to 2 or vice versa, even if caused by lost connection to the station (suspension)</td>
</tr>
<tr>
<td>&quot;DEFAULT&quot;</td>
<td>No station specific policy has been defined, the policy of the application (APL:BPP) is applied.</td>
</tr>
</tbody>
</table>

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.

If the policy is "DEFAULT" and the policy of the application (APL:BPP, see Chapter 6) is "DEFAULT" as well, the "WHEN_SET_TO_2" policy is applied for compatibility.

**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, SRI0, 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. RTUs
- "PCL" Procontrol 214
- "SIN" Sindac stations using the ADLP80 protocol
- "SID" Sindac stations using the ADLP180 protocol
- "PAC" PAC-5, LED & Keyboard interface
- "SAT" SATTCON Central Station through the COMLI Slave connection
- "RCT" PROCOL Station Interface
- "LCU" Local Control Unit (Load Management)
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

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.

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
For stations connected to a NET unit, TN is the STA system object number in the NET.

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.

**TT Translation Type**

The operating state and location of the station.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Text keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>&quot;NONE&quot;</td>
</tr>
<tr>
<td></td>
<td>Off, out of operation</td>
</tr>
<tr>
<td>&quot;ALIAS&quot;</td>
<td>The STA object number refers to another STA object</td>
</tr>
<tr>
<td>&quot;EXTERNAL&quot;</td>
<td>Connected to a communication unit (the normal case)</td>
</tr>
</tbody>
</table>

Default value: "NONE"

Access: No restrictions

10.2. STY objects

10.2.1. General

**Station types**

The predefined station types listed in the ST attribute description in Section 10.1.2. 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 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 you use the names given in Table 10.2.1-1 because these are the names used on NET unit. The table lists some station types and the corresponding database interfaces.
STY object notation

The STY objects and their attributes are accessed with the following object notation:

\[ \text{STYn:Bat} \]

where

- \( n \) is the internal station type number, 1 ... 33. The following station type numbers are preset:

<table>
<thead>
<tr>
<th>Station type</th>
<th>Name (recomm.)</th>
<th>Database interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations connected via the RP570 slave protocol</td>
<td>SPI</td>
<td>STA</td>
</tr>
<tr>
<td>LSG device and other LonWorks devices, except REX</td>
<td>LMK</td>
<td>REX</td>
</tr>
<tr>
<td>PLCs using MODBUS</td>
<td>PLC</td>
<td>RTU</td>
</tr>
</tbody>
</table>

* These station type numbers have to be created separately.

An alternative way of using freely chosen object names is described in Chapter 4.
10.2.2. **STY attributes**

Common naming attributes BN, BT and BM are described in Chapter 4. The STY specific attributes are described below.

**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.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>A text vector of length 255. Each element in the vector has one of the following keyword values: &quot;UNKNOWN&quot;, &quot;SPONTANEOUS&quot;, &quot;INTERROGATED&quot;, &quot;HISTORY&quot;</td>
</tr>
<tr>
<td>Indexing:</td>
<td>1 ... 255. The value of the process object attribute CT</td>
</tr>
<tr>
<td>Default value:</td>
<td>The CT attributes of the STY objects are all initialized to &quot;UNKNOWN&quot;, except for the following REX device values:</td>
</tr>
<tr>
<td></td>
<td>STY17:BCT1 &quot;SPONTANEOUS&quot;</td>
</tr>
<tr>
<td></td>
<td>STY17:BCT2 &quot;INTERROGATED&quot;</td>
</tr>
<tr>
<td></td>
<td>STY17:BCT3 &quot;HISTORY&quot;</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

**Example:**

If a device of type 17 (REXDEVICE) updates a process object's CT attribute to value 2, the attribute STY17:BCT2 defines the reason of that transmission (“INTERROGATED by default”).

**CX**  
**Comment Text**

A freely chosen comment text.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>Any text, up to 255 characters</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

**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 10.2.1-1. The predefined interface is used when NET unit issues commands to the stations of the new type.

| Data type: | Text keyword |
**MicroSCADA Pro**

**System Objects**

**Technical Description**

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th>The three-letter station type name of a predefined station type, see Table 10.2.1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access</strong></td>
<td>Read-only, configurable</td>
</tr>
</tbody>
</table>

**Example:**

Creating a station type "SPI" (type 22):

```plaintext
#CREATE STY22:B = LIST( NA = "SPI", DB = "STA")
```

**LP Load Control Policy**

Load control policy applied to the analog input objects of the stations of the type.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Text keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value:</strong></td>
<td>See below.</td>
</tr>
</tbody>
</table>

- "DEFAULT"
- "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.
- "KEEP_ALL_ANALOGS" No analog object events are subject to load control.

**Default value:** "DEFAULT"

**Access:** No restrictions

For those STY objects, which 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").

<table>
<thead>
<tr>
<th><strong>NA Type Name</strong></th>
</tr>
</thead>
</table>

The name of the station type.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value:</strong></td>
<td>Text of max. 10 characters</td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td>Read-only, configurable</td>
</tr>
</tbody>
</table>
11. PRI objects for base system

About this chapter
This chapter describes the PRI base system objects and their attributes. It contains two sections:

11.1 General: the definition of PRI objects and the PRI object notation.

11.2 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 (CM), global variables (SV, UV).

11.1. General

PRI object definitions
All printers used by the MicroSCADA base system (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 Chapter 6). 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 11.2.5). It is also possible to define more than one PRI object for one physical printer. This may be useful if you wish several types of printout (for example, semi-graphic and full graphic) to the same printer.

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 Chapter 6)

'n'  The object number of the printer in the other base system

An alternative way of using freely chosen object names is described in Chapter 4.

11.2. PRI attributes

The common naming attributes BN, BT and BM are described in Chapter 4.

The PRI specific attributes are described below.

11.2.1. Common printer attributes

**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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;NONE&quot;</td>
<td>No printer</td>
</tr>
<tr>
<td>&quot;LINE&quot;</td>
<td>Printer connected directly to the base system computer or to a LAN via a printer server</td>
</tr>
<tr>
<td>&quot;NET&quot;</td>
<td>Printer connected to a NET unit</td>
</tr>
<tr>
<td>&quot;CONSOLE&quot;</td>
<td>Printer connected to LPT1</td>
</tr>
</tbody>
</table>

Default value:  "NET"

Access:  Read, conditional write

**DT  Device Type**

The type of the printout - color, black-and-white or “transparent”.

Data type:  Text keyword

Value:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;NORMAL&quot;</td>
<td>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.</td>
</tr>
<tr>
<td>&quot;COLOR&quot;</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 Chapter 17). Some examples of PRI object definitions for different printer types and connections are presented later in this chapter.

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:

\[
\begin{align*}
\text{PRIn:BTT} & = \text{"LOCAL"} \\
\text{PRIn:BDT} & = \text{"NORMAL"} \\
\text{PRIn:BDC} & = \text{"LINE"} \\
\text{PRIn:BSD} & = \text{"/SCADA/LP"} \\
\text{PRIn:BOJ} & = 1
\end{align*}
\]

2. Printer connected to a NET unit:

Black-and-white ASCII Interface (PRIn:SPT = = 1):

\[
\begin{align*}
\text{PRIn:BTT} & = \text{"LOCAL"} \\
\text{PRIn:BND} & = \text{NET node number} \\
\text{PRIn:BTN} & = \text{PRI system object number in NET unit} \\
\text{PRIn:BDT} & = \text{"NORMAL"} \\
\text{PRIn:BDC} & = \text{"NET"}
\end{align*}
\]

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):

\[
\begin{align*}
\text{PRIn:BTT} & = \text{"LOCAL"} \\
\text{PRIn:BND} & = \text{NET node number} \\
\text{PRIn:BTN} & = \text{PRI system object number in NET unit} \\
\text{PRIn:BDT} & = \text{"NORMAL"} \\
\text{PRIn:BDC} & = \text{"NET"}
\end{align*}
\]

**TT Translation Type**

The operating state of the printer. See examples considering DT attribute above.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;NONE&quot;</td>
<td>Off, out of use</td>
</tr>
<tr>
<td>&quot;LOCAL&quot;</td>
<td>Normal operating state (independently of the type of connection)</td>
</tr>
<tr>
<td>&quot;ALIAS&quot;</td>
<td>The PRI object redirects the printout to another printer determined by the TN attribute, see below.</td>
</tr>
</tbody>
</table>
11.2.2. Printer connection attributes

**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"

**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
- **Default value:** None
- **Access:** Read, conditional write

**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"
11.2.3. Printout attributes

**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, Chapter 8. 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, Chapter 8.

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).

**HF Header Format**

The header text produced on each printed page when page numbering is in use (PN \( \geq 0 \)). 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:

\[
\begin{align*}
\text{HF}(1) &= \text{"EVENT LIST  ########  PAGE  ###"} \\
\text{HF}(2) &= \text{"DATE"} \\
\text{HF}(3) &= \text{"%PN"}
\end{align*}
\]

This HF value might generate the following page header:

EVENT LIST 97-03-20 PAGE 12.

**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

**LP Lines per Page**

Number of lines per page.

Data type: Integer
Value: Non-negative integer. 0 = no automatic form-feed.
MicroSCADA Pro
System Objects
Technical Description

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

11.2.4. Printer queue attributes

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, Chapter 8) 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

QU Queue Length Used
Current length of the printer queue. See attribute QM.

Data type: Integer
Access: Read-only

11.2.5. 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):

<table>
<thead>
<tr>
<th>Period</th>
<th>File Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>daily log</td>
<td>prnnymmddd.log</td>
<td>pr02920201.log</td>
</tr>
<tr>
<td>weekly log</td>
<td>prnnyywww.log</td>
<td>pr0292w05.log</td>
</tr>
<tr>
<td>monthly log</td>
<td>prnnyymm.log</td>
<td>pr029202.log</td>
</tr>
<tr>
<td>yearly log</td>
<td>prnnyy.log</td>
<td>pr0292.log</td>
</tr>
</tbody>
</table>

Storing printout in a printer log is possible even if the printer does not exist physically. Hence, you can define virtual printers, that is, PRI objects without corresponding physical printers, 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.

**LD Log Directory**

The directory where the printer log is stored on disk. The directory is given including complete path in the operating system format.

The attribute has no meaning if OD = "PRINTER". The PRI object attribute OD defines the output destination of the log.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>0 … 255 characters, see Chapter 3 of the Programming Language SCIL manual for the allowed characters.</td>
</tr>
<tr>
<td>Default value:</td>
<td>&quot;&quot; (an empty text string) which means the directory /sc/sys/active/sys_</td>
</tr>
<tr>
<td>Access:</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>

Define the path starting from the root if the attribute will be used with the #PATH command.
MicroSCADA Pro  
System Objects  
Technical Description

**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.

**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"

**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
11.2.6. **Printer control attributes**

**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"

**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
MicroSCADA Pro
System Objects
Technical Description

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

11.2.7.  Miscellaneous PRI attributes

CX  Comment Text
A freely chosen comment text.
Data type:  Text
Value:  Any text, up to 255 characters
Access:  No restrictions

SV  System Variables
Global variables used in tools, etc. The attribute should not be used in application programs.
Data type:  Vector
Value:  Defined by ABB
Default value:  A vector of 10 integer 0’s
Access:  No restrictions

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
12. Base System Object Navigator

About this chapter
This chapter describes the Base System Object Navigator tool located in the SYS 600 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).

12.1. 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 SYS 600 system.
- Viewing of the base system related attributes and their values.
- Editing of the base system object related attribute values (see certain limitations later on).
- 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 or sys_bascon.hsb).

Recognizing of the base system objects
During start-up of the tool, it 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. Following base system objects are presented:
- Base object (SYS)
- Applications (APL)
- Monitors (MON)
- Links (LIN)
- Nodes (NOD)
- Stations (STA)
- Station types (STY)
- Printers (PRI)
12.2. Viewing and editing of the base system objects

12.2.1. Base system objects

In the main view of this tool, all of the found above mentioned base system objects are presented in hierarchical way in an object tree. Navigation in the object tree, e.g. expanding of Applications category, shows a set of leaves, each of which represent one application that was located from the SYS 600 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
- 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.

12.2.2. Base system object attributes

In the main view of this tool, also all the base system object related attributes and their values are presented in hierarchical way in attribute tree. E.g., selecting of one leaf under certain category, lists all the base system attributes and values regarding to selection. The amount of shown attributes depends on the base system object type. Each unique attribute is belonging to the group of attributes. The used group names are identical to those used in the documentation of SYS 600 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
MicroSCADA Pro
System Objects
Technical Description

• Application diagnostics attributes
• Mirroring configuration attributes
• Mirroring diagnostics attributes
• Miscellaneous application attributes

Attributes may also be shown in alphabetical order by selecting All Attributes in Alphabetical Order from 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 of 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 included into the end of attribute description inside the brackets (e.g. Shadowing Flush Time [ms].

When an attribute is selected in the tool, the help text of an attribute is displayed in the information bar of the tool.

Limitations: In the attribute tree it is not possible to fully show complex data type (deeply nested lists or vectors containing elements of such lists, e.g. System Variable (SV) and User Variable (UV) attributes).

12.2.3. Base system object attribute values

In the main view of this tool, also the base system attribute values are presented and they may be modified in the edit attribute area. When a base system attribute is selected under certain group name, its current value is presented and using the appropriate dialog items a user may modify them. E.g. when an attribute value contains certain keywords, the possible keywords are presented in the 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). The existing element cannot also be removed. The capabilities of the edit area are to display the data structures as static and edit values of existing elements.

12.3. Adding of a new base system object

When a new base system object is added, 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 or enter keystroke Ctrl+N or click the object adding 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 asked from the user. Depending from 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, e.g. for a new application object, at least the following information needs to be specified:

- 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.

### 12.4. Starting of the tool

To start the Base System Object Navigator tool, click the Base System tool icon in the SYS 600 Tool Manager System Configuration page and select Open from File menu or double-click the tool icon (see Fig. 12.4.-1).

![Base System Object Navigator tool on the System Configuration page of Tool Manager.](image-url)
The Base System Object Navigator tool page includes a menu bar and a toolbar, which can be selected from the Option > Toolbar Visible menu. 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.

Fig. 12.4.-2 The menu bar, tool bar, object tree and attribute tree in the Base System Object Navigator tool.

Fig. 12.4.-3 The attribute area and attribute edit area in the attribute tree of the Base System Object Navigator tool.
12.5. 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 Fig. 12.4.-2). 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 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 clicking the right mouse button on the tree and selecting Expand All or Collapse All from the pop-up menu (see Fig. 12.6.-3).

![Fig. 12.5.-1 The Expand and collapse buttons and pop-up menu for the attribute tree.](image)

12.6. Tools in the tool

The following tool is introduced to ease the management of shadowing applications.

- Hot Stand-By Management tool.

Selecting Base Object from the object attribute tree and HSB Management from Tools menu starts the management tool. 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 tells versions and statuses of the installed package and disk package. Clicking the Install button installs the disk package (see Fig. 12.6.-1).

Fig. 12.6.-1 The Packages page of the HSB Management tool.

The Shadowing Applications page shows information from the local watchdog application and shadowing applications (see Fig. 12.6.-2).

Fig. 12.6.-2 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 Fig. 12.6.-3).

![Shadowing Application Edit Dialog]

*Fig. 12.6.-3 The edit dialog of a shadowing application.*
13. Communication system objects, overview

About this chapter
This chapter introduces the communication system objects:

13.1 General. This section gives a general description of communication system objects, their features and functions.

13.2 Communication system object types. The types of the communication system objects. The types are illustrated by a picture and each type is described briefly.

13.3 Defining communication system objects. The principles for defining communication system objects, the configuration software and the required definitions.

13.4 Attributes. The function of the communication system attributes and the attribute access levels.

13.1. General

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 RAM memory of the NET unit (DCP NET units) or host PC (PC NET units).

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 Fig. 13.2.-1). 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.

13.2. Communication system object types

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, base systems and communication frontends, 'n' = 1 ... 250
- **APLn**: Application objects, 'n' = 1 ... 32
- **STAn**: Station objects, 'n' = 1 (0) ... 255 (each station type has its own number series)
- **PRI n**: 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 "communication system object number" or, where no misunderstanding can occur, only "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 MicroSCADA system.

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 Chapter 4 for naming base system objects.

Fig. 13.2.-1 illustrates the system object types and their roles in the system. After that follows a brief presentation of each system object type.
**NOD (NET) objects**

The NOD or NET objects (both names are equivalent) represent the NET unit itself as well as every other known node - communication units, base systems and communication frontends - in the MicroSCADA system. The object numbers of the NET objects are global and may be 1 ... 250.

The MicroSCADA 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.

*Fig. 13.2.-1 The communication system objects defined in the shaded NET unit that may be an internal DCPNET or a PCNET running in the base system. The unbolded and unmarked devices require no object definitions in the NET. (If the shaded NET is a PCNET, no printer can be connected to it.)*
• Accessing application object attributes and frontend 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 chapters.

**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 recognise up to 32 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.

**STA objects**

This system object type is used to supervise and control properties of connected stations - RTUs, protective equipment, central stations, etc. Each station connected to the NET unit must be defined as a STA object. The STA objects specify and handle, e.g.:

• The data transfer to and from stations
• RTU diagnostics (type of diagnostic commands used, diagnostic counters)
• Response waiting timeout
• Operational status (in use or not in use)

The STA objects are defined in different ways and have different attributes depending on the station type they represent. There are also separate series of object numbers for the different station types.

One NET unit can be connected to up to about 200 stations (depending on the total configuration of the unit). There can be 20 stations defined in the preconfiguration, the rest must be defined with SCIL into a command procedure executed by APL-INIT1 (DCP-NET unit) or into the PCNET unit.COM file (PC-NET unit) read on execution command from APL-INIT1.

**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)
### Defining communication system objects

#### 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 Chapter 15).
- Giving the line its communication properties using the line attributes (Chapter 15).
- 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 communication system objects of DCP-NETs can be defined in a preconfiguration (see below). PC-NET units have no preconfiguration but an initialization file that contains the most elementary object definitions. The configuration of the NETs can be extended and modified on-line 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, ST, RT, PC, PR, etc. which are described in Chapter 14. The communication system objects cannot be created, deleted nor modified with the SCIL commands #CREATE, #DELETE and #MODIFY.

The preconfiguration of a DCP-NET unit is stored in the communication program and is therefore not lost though the NET unit is stopped. The on-line and start-up system object definitions are stored only as long as the NET unit is running. If it is stopped or restarted, the on-line configurations are lost.

#### Preconfiguration of DCP-NET units

To some extent, the communication system objects of DCP-NET units are defined as a "preconfiguration". The preconfiguration parameters (attributes) serve as default values of the attributes. When a DCP-NET unit is started or restarted, the preconfiguration becomes valid and remains valid until it is changed with the start-up configuration or on-line (with SCIL). The preconfiguration parameters are written with a MicroSCADA tool or with a program called NETCONF that runs under DOS (see the System Configuration Manual). A changed preconfiguration becomes valid when the NET unit is restarted.

All attributes cannot be preconfigured. The possibility to preconfigure the attribute is mentioned in the attribute descriptions.

#### 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.
Start up configuration for DCP-NET unit

SYS_NETCON.COM is a text file containing the start-up commands of internal DCP-NET units. For more information, see the LOAD_DCP function in Chapter 9 of the Programming Language SCIL manual. This file is automatically executed by the main program when the base system is started, after the SYS_BASCON.COM file. The start-up command in the SYS_NETCON.COM contains the name of the configuration program file that is to be loaded to the DCP-NET unit.

The SYS_NETCON.COM command file can be changed with a text editor on-line or off-line. The changes come into force when the base system is restarted.

A PC-NET unit is started by the SYS_BASCON.COM file when the LIN base system object connecting it to the base system is created.

On-line configuration

In addition, the communication system objects can be written on-line 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 Application Objects Chapter 9. It is also possible to build command procedures that are started at each start-up of a NET unit.

For PC-NET units there is a configuration file, where the SCIL definitions are written. It is named PCNET.COM. This file is a text file, which is read from a command procedure executed of an APL_INIT event channel at start-up.

Required Object Definitions

Table 13.3.-1 gives an overview of the communication system object definitions required by various configuration projects. It also refers to the chapters and sections where the attributes are detailed. The communication system objects required for various system set-ups are detailed in the System Configuration manual.
### 13.4. Attributes

#### General

The system object attributes are of two types:

- **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.

#### Attribute access

The communication system object attributes are of the following main access levels:

- **Read-only**: The attribute cannot be written either on-line or in the preconfiguration. There are still a few exceptions in which the values can be reset.
- **Not preconfigurable, otherwise no restrictions**: The attribute can be given values on-line, but not in the preconfiguration. Otherwise the attribute can be read and written without restrictions.
- **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 #SET command. It is not stored in NET unit, but transmitted directly to a station.
• **Write-only, not preconfigurable**: The attribute cannot be read, only written with #SET command. It is not stored in NET unit, but transmitted directly to a station. The attribute can be given values on-line, but not in the preconfiguration.

• **No restrictions**: The attribute can be read and written without restrictions.

The terms above are used in the subsequent attribute descriptions. With preconfiguration it is meant that a DCP-NET Unit is configured with Netconf or NET Preconfiguration tool. PC-NET is configured with the System Configuration tool or by using SCIL statements, so the limited access rights for preconfiguration are not valid for PC-NET Unit.
14. NOD (NET) objects for communication system

About this chapter
This chapter describes the NOD (NET) objects and their attributes.

14.1 General: The definition of NOD (NET) objects and the object notation.
14.2 Basic NET Attributes: Basic Definitions (NN, SA, SX), Functional Specifications (TL), NET Information (KP, NT, VE), System Message Attributes (MI, MS, SE), Connected Nodes (AD, LI).
14.3 Object Definition Attributes: External Nodes (NE), Applications (SY), Station Definition Attributes (LC, LM, PA, PC, RT, RX, SI, SM, SN, SO, SP, ST), Adding Devices of Exchangeable Device Types (DV, ED), Printer Definition Attributes (PR).
14.4 Application Attributes (DS, SU, SW).
14.5 Frontend Attributes (CS, DC, DI, RS, TM).
14.6 Redundancy Attributes (PN, RM, SH, XC, XS, XT).
14.7 Miscellaneous NET Attributes (FM, LT, MA, MU, TM).

The NET attributes related to the communication lines are described in the next chapter.

14.1. General

NET object definition
The following devices must be defined as NOD (NET) objects in a NET unit (NET):
- The NET unit itself.
- All other communication units which will be recognized by the unit.
- All base systems, which will communicate with the unit.
- Gateways (protocol converters) which will communicate with the NET.

The NET unit itself must always be defined in the preconfiguration if it is a DCP-NET, or in the initialization file if it is a PC-NET. The other nodes ("external nodes") can be defined in the preconfiguration (DCP-NETS) or in a SCIL program by setting the NE attribute (see Section 14.3) with the #SET command.

The system object numbers of the nodes (node numbers), 1 ... 250, must be unique within the entire MicroSCADA system. 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 Chapter 9). The node numbers can be taken into use in any order.

Object Notation
From SCIL the NET attributes of the NET unit itself is accessed with the notation (the NOD and NET names are equivalent):

\texttt{NODn:Sat (or NETn:Sat)}
where
'n' The object number (= node number) of the unit, 1 ... 250
'at' An attribute name

The same notation is used when accessing the attributes of communication frontends (provided that they have been defined as nodes in the base system). These have a few NET attributes, although they are not defined as NET objects to the communication units.

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:

\[ \text{NODn:Satm} \text{ (or NETn:Satm)} \]

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.

14.2. Basic NET attributes

14.2.1. Basic definitions

**NN**  
Node Number

The NET object number. The number is defined when the NET object is created in the preconfiguration or with the NET attribute NE (Section 14.3). 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):

\[ \text{#SET NET1:SNN2 = 3} \]

**SA**  
Station Address

The ACP station address of the NET object (communication unit, base system or communication frontend). 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 on-line station address configuration the communication program checks that this uniqueness is maintained.

Data type: Integer  
Value: 1 ... 254
MicroSCADA Pro
System Objects
Technical Description

Example:
 Defines that the NET2 connection in NET1 will have the station address 27:

#SET NET1:SSA2 = 27

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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1 ... 254</td>
</tr>
<tr>
<td>Default value</td>
<td>The value of the SA attribute set in the preconfiguration</td>
</tr>
<tr>
<td>Access</td>
<td>No preconfigurable, otherwise no restrictions</td>
</tr>
</tbody>
</table>

14.2.2. Functional specifications

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).

<table>
<thead>
<tr>
<th>Data type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1 ... 255</td>
</tr>
<tr>
<td>Default value</td>
<td>10</td>
</tr>
<tr>
<td>Suggested value</td>
<td>5 (normally there is no need to change the default value)</td>
</tr>
<tr>
<td>Access</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>
14.2.3. NET information

**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 Chapter 15.
Access: Read-only

**Example:**
The NET version NET_NC1_82 would return a vector with the following elements: (1,2,3,4,7,15)

**NT**  Node Type
The type of the node: NET, basesystem, communication frontend, workstation or gateway.
Data type: Integer
Value: 0 ... 6:
0 Unknown
1 Basesystem
2 NET (communication unit)
3 Communication frontend (MFL)
4 Semi-graphic workstation (MicroWORKSTATION)
6 Gateway
Default value: 2
Access: Read-only

**VE**  Program Version
The version number and generation date of the NET program, for example 8.2A, or the version identification of the communication frontend program (MFL).
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
Access: Read-only
Index 1 (only possibility for MFL nodes)
14.2.4. 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 14.4): suspension = application number, recovery = 1000 + application number (the application number as defined to NET unit).
- The MU attribute, Section 14.7, has been updated: code 16633.

In the process database of an application the attributes MS and MI 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 Fig. 14.2.4.-1.

For more information, see the System Configuration manual.
System Messages

**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 three types of system messages, which are directed to process objects as follows:

- General NET messages: MI
- Messages related to application diagnostics: MI + 50
- Messages related to redundant frontends: MI - 100

Data type: Integer  
Value: 1 ... 32760
MicroSCADA Pro
System Objects
Technical Description

Example:
The system message receiving process object in application 3 should have the following attribute features:

Example:

```
*SET STA2:SMS = 3
```

Logical Name: Any
Unit number: UN = 0
Object Address: OA = the MI attribute of the system object
Type: AI
and the desired event, alarm and printout attributes (see Chapter 4).

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 startup system message code (10001) is sent to all known applications, independently of the MS attribute.

Default value: The following default values, which are also the recommended values, are valid if the attribute is given no value or the value 00000 in the preconfiguration:

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

Data type: Integer
Value: Integer in the range 1 ... 32. 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

**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.

Data type: Integer
Values:
0 System message generation disabled
1 System message generation enabled (normal value)
2 NET startup system messages enabled, all other system messages disabled.

Default value: 1
Access: No restrictions

The value SE = 0 should only be used in special cases, for example if the base system application program often executes commands, which cause undesirable system messages.

### 14.2.5. Connected nodes

**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:

\[
4096 + (\text{NET object number})
\]
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: Not preconfigurable, otherwise 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.
MicroSCADA Pro
System Objects
Technical Description

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: 1 ... 8
- Indexing: The NET object number of the connected NET unit
- Access: Read-only

Example:
!SHOW LINE NET3:SLI2

14.3. Object definition attributes

14.3.1. 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. In the preconfiguration of DCP-NET units, the nodes are defined as "External nodes". The external nodes are defined by giving them a device number (= object number, node number) and assigning them a line number of an ACP, RAM or Integrated Link line. Up to 15 external nodes can be defined in the preconfiguration.

With SCIL, 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 14.2.

NE  NET
Adding and removing the NET objects of connected nodes - 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, RAM or Integrated Link line (see Chapter 15). The line number for a node that is connected indirectly via other communication units is the line number of the nearest communication unit. Internal communication units are connected to the base system on line number 13. Likewise, the base systems connected to a communication frontend via a LAN or a COM port is defined 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.
A new NET object gets certain default attribute values listed in appendix A. If a NET object is created in the preconfiguration, only those attributes that cannot be preconfigured get default values. The other ones get the values given in the preconfiguration.

**Data type:** Integer  
**Value:**  
- Adding an object: Integer, 1 ... 8. Line number. The line must be a common RAM, 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 MicroSCADA system.  
**Access:** No restrictions

**Example:**

A NET3:S object is added to line 4 of NET2:

```plaintext
#SET NET2:SNE3 = 4
```

The connection to NET3 is removed from NET2:

```plaintext
#SET NET2:SNE3 = "D"
```

### 14.3.2 Applications

All applications that will communicate with a NET unit, that is, 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). Up to 16 applications can be defined in the preconfiguration of DCP-NET units. Using SCIL, applications are defined by means of the SY attribute described below.

**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

150
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:SYS5 = (9,5)
```

14.3.3. Station definition attributes

Each 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.

Up to 20 stations can be defined in the preconfiguration of a DCP-NET unit. If the NET unit will be connected to more stations, they must be added on-line with SCIL or tools.

In the preconfiguration tools, and in this manual, 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
- SIN: SINDAC connected via the ADLP80 protocol
- PCL: P214 RTUs
- SID: SINDAC connected via the ADLP180 protocol
- PAC: PAC-5
- SAT: SATTCON/COMLI
- 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 14.3.4..

All types of stations, except REX and LMK, can be connected to the DCP-NET units, while only the RTU, REX, SPA and LMK stations can be connected to the PC-NET units.

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 Fig. 14.3.3.-1. 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. Reference objects can not be added in the preconfiguration.

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 Chapter 16. If a STA object is created in the preconfiguration, only those attributes, which cannot be preconfigured get the default values, the other attributes get the values given in the preconfiguration.

---

**Fig. 14.3.3.-1** 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.
MicroSCADA Pro
System Objects
Technical Description

**LC Load Control Unit**
Defining and removing STA objects corresponding to LCU500.

- **Data type:** Integer
- **Value:**
  - Adding a device: Integer, 1 ... 8. 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

**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 RE 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 ... 8. Line number. The line must be defined for the LonTalk protocol.
  - Removing an object: Text, "D"
- **Indexing:** Integer, 1 ... 255. 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 Chapter 16.):

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
   ```

**PA PAC - 5**
Defining and removing STA objects corresponding to PAC - 5.

- **Data type:** Integer or text
- **Value:**
  - Adding a device: Integer, 1 ... 8. Line number. The line must have been defined for the PAC-5 protocol.
Removing a device: Text, "D"

Indexing: Integer 1 ... 255. The STA object number. The number must be unique for STA objects of type PAC.

Access: No restrictions

**PC**  Procontrol P214

Defining and removing STA objects corresponding to P214 RTUs.

Data type: Integer or text

Value: Adding a device: Integer, 1 ... 8. 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

**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


Removing a device: Text, "D"

Indexing: Integer, 0 ... 255. 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

**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 ... 8. Line number. The line must have been defined for the LonTalk protocol.

Removing a device: Text, "D"

Indexing: Integer 1 ... 255. The STA object number. The number must be unique for STA objects of type REX.

Access: No restrictions
**SI**  **SINDAC ADLP80 Stations**

Creating STA objects of station type SIN (SINDAC using the ADLP80 protocol).

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>Adding a device: Integer, 1 ... 8. Line number. The line must have been defined for the ADLP80 protocol. Removing a device: Text, &quot;D&quot;</td>
</tr>
<tr>
<td>Indexing:</td>
<td>Integer 1 ... 255. The STA object number. The number must be unique for STA objects of type SIN.</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

**SM**  **S.P.I.D.E.R. SCADA Master Stations**

Creating STA objects of station type SPI.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer or text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>Adding a device: Integer, 1 ... 8. Line number. The line must have been defined for the RP570 slave protocol. Removing a device: Text, &quot;D&quot;</td>
</tr>
<tr>
<td>Indexing:</td>
<td>Integer 1 ... 255. The STA object number. The number must be unique for STA objects of type SPI.</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

**SN**  **SINDAC ADLP180 Stations**

Creating STA objects of station type SID.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer or text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>Adding a device: Integer, 1 ... 8. Line number. The line must have been defined for the ADLP180 protocol. Removing a device: Text, &quot;D&quot;</td>
</tr>
<tr>
<td>Indexing:</td>
<td>Integer 0 ... 255. The STA object number. The number must be unique for STA objects of type SID.</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

**SO**  **SATTCNON COMLI Stations**

Creating STA objects of station type SAT.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer or text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>Adding a device: Integer, 1 ... 8. Line number. The line must have been defined for the COMLI protocol. Removing a device: Text, &quot;D&quot;</td>
</tr>
<tr>
<td>Indexing:</td>
<td>Integer 1 ... 255. The STA object number. The number must be unique for STA objects of type SAT.</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>
Creating STA objects corresponding to SPA units (SPACOM modules or other stations) using the SPA or LonTalk protocol.

It is possible to configure about 200 SPA devices to one PC-NET. In one PC-NET, there can be a maximum of 12 SPA lines created. The amount of SPA devices in one SPA line could be about 30. 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 you have per line, and the more there are configured analog measurements, the slower the response time in the system. In other words, if you want the response times to be rather 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 ... 8. 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 ... 255. 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

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 ... 8. Line number. The line must be defined for the ANSI protocol.
Removing a device image: Text, "D"
Adding a reference: 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:

```plaintext
#SET NET1:SST5 = 4
```

Addition of the system object STA20 in NET1. STA20 refers to STA3 in NET1:

```plaintext
#SET NET1:SST20 = 10003
```

Addition of the system object STA4 in NET1. STA4 refers to STA2 in NET2:

```plaintext
#SET NET1:SST4 = 20002
```

Removal of the logical name STA20:

```plaintext
#SET NET1:SST20 = 0
```

**14.3.4. Adding devices of exchangeable device types**

Internally in NET unit, each object type is represented by an integer code, 1 ... 31. The code numbers 1 ... 23 are predefined, including all the station types enumerated in Section 14.3.3. The codes 24 ... 31 can be used for different device types defined by the ED attribute. By default, NET unit knows the exchangeable device types by the following codes:

- 24 = ADEMCO
- 25 = Procontic/RCOM
- 26 = Westinghouse
- 27 = Alpha meter
- 28 = PLC interface

STA objects of these types are created with the DV attribute.

There is no need to use the ED attribute for changing the device code as long as the codes 24 ... 31 are enough.

**DV Device**

Adding and removing devices of exchangeable device types. The device type must have been defined with the ED attribute.

<table>
<thead>
<tr>
<th>Data type: Vector</th>
<th>Value: Write: Vector of two elements:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Element 1: Communication system object number:</td>
</tr>
<tr>
<td></td>
<td>Element 2: Line number, when adding a device &quot;D&quot;, when removing a device</td>
</tr>
<tr>
<td>Read:</td>
<td>The defined device numbers of the given device type. 0 = undefined</td>
</tr>
</tbody>
</table>

| Indexing: Write: One index, the device type code, 24 ... 31 |
| Read: 1000 * type code + object no |

**Example:**

A station of type number 24 (ADEMCO) is added on line 3 as STA2:

```plaintext
#SET NET1:SDV24 = (2,3)
```
The station added above is removed:

```plaintext
#SET NET1:SDV24 = (2, "D")
```

The variable A will get the value 0 if no such object is defined, otherwise the value 5:

```plaintext
A = NET1:SDV24005
```

**ED**  
**Exchangeable Device table**

Defines the device types of the device type code numbers 24 ... 31. The attribute can be used for reading and writing the device type codes.

- **Data type:** Vector
- **Value:** Vector of integer numbers, 32 ... corresponding to a station type code from the following list:
  - 32 ADEMCO
  - 33 PROCONTIC
- **Indexing:** Device type code, 24 ... 31
- **Default values:**
  - Index 24 = 32
  - Index 25 = 33
- **Access:** Not preconfigurable, otherwise no restrictions

### 14.3.5. 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 in the preconfiguration, with SCIL or with tools. With SCIL, printers are defined, that is, new PRI objects are created, with 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 Fig. 14.3.3.-1. 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. If the object is created in the preconfiguration, only those attributes which can not be preconfigured get the default values in appendix A, the other attributes get the values given in the preconfiguration.

**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 ... 8. Line number. The line must be defined as an ASCII line.
  - Removing a device image: Text, "D"
Adding a reference: A vector of two integers of the form
(NETnr, device nr) where 'NET nr' is the object number of the NET
where 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"

14.4. 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.

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

**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 ... 32)  
*Access:* Read-only

Example:
A is a vector with the status of all applications known to NET1:

@A = NET1:SDS(1..32)

**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 nonzero status code to a diagnostic command, a splitted 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 canceled 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 14.2.4) 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 14.2.4.

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
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
```

**SW Application Reply Wait Time**
The reply wait time in seconds for diagnostic commands and splitted substation messages (see Chapter 16) 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

**14.5. Frontend node attributes**
Provided that they have been defined as NOD:B objects in the base system, the communication frontends can be accessed from SCIL as NOD (NET) objects. The frontend nodes have the attributes described in this section and the SA, NT and VE attributes in Section 14.2. (read-only). The attributes in this section apply exclusively for NET objects corresponding to communication frontends.

The attributes are accessed by the notation:

```
NODn:Sat (or NETn:Sat)
```
where
‘n’  Is the node number of the communication frontend (defined in the base system by the NODn:B object), and ‘at’ is the attribute name.

CS  Clock Status
The status of an internal or external PC31/PC32 clock in the communication frontend.

Data type:  Vector
Value:  Vector with two elements:
Index 1:  0 = Clock time invalid (not synchronized), or no C31/32 clock installed
         1 = Clock time valid (synchronized)
Index 2:  Last status byte value read from the PC31/PC32 clock (internal)
         255 = No internal clock installed, or no status read
Index 2 applies to internal clocks only
Access:  Read-only

Example:
CLOCK_STAT = NOD6:SCS

DC  Diagnostic Counter
The values of the 16 diagnostic counters of the communication interfaces of the communication frontend. The counters have the following meanings:

3. TRANSMITTED MESSAGES/TELEGRAM
4. FAILED TRANSMISSIONS
5. TRANSMITTED TIMEOUTS
6. TRANSMITTED ACKS
7. TRANSMITTED NAKS
8. TRANSMITTED ENQS
9. RECEIVED ACKS
10. RECEIVED NAKS
11. RECEIVED ENQS
12. RECEIVED EOTS
13. RECEIVED MESSAGES/TELEGRAMS
14. PARITY ERRORS
15. OVERRUN ERRORS
16. CHECK SUM ERRORS
17. FRAMING ERRORS
18. BUFFER OVERFLOW ERRORS

Data type:  Integer
Value:  0 ... 30000
Indexing:  100 * interface nr + counter number
MicroSCADA Pro
System Objects
Technical Description

Example:
The value of counter number 1 on LAN interface 1:
@COUNTER = NET5:SDC701

**DI**
**Diagnostic Interval**
The time between base system diagnostic commands sent by the communication frontend. The attribute is the value of the DI parameter defined in MFLCONF.DAT.

- **Data type:** Integer
- **Value:** 0 = Diagnostic turned off
- **Unit:** Seconds
- **Indexing:** 1 ... 4, base system interface number
- **Default value:** 15
- **Access:** Read-only, values can be reset

**RS**
**Running Status**
Reloads and starts the communication units in the communication frontend.

- **Data type:** Integer
- **Value:** 3 = Reload and start all NET units of the communication frontend
- **Access:** Write-only

**Example:**
#SET NET5:SRS = 3

**TM**
**Time**
Reading and writing the frontend time. When the frontend time is set with this attribute, the time of all NET units is set as well.

- **Data type:** Integer
- **Value:** Vector with (year, month, day, hour, minute, second, milliseconds)
- **Access:** No restrictions

**14.6. Redundancy attributes**
A redundant frontend pair consists of two communication frontends, each containing one NET unit, and line switches connected to RTUs. Each redundant NET unit can be either the primary unit, which handles the communication with the
RTUs, or the stand-by unit. The stand-by unit is ready to take over when a malfunction occurs in the primary one. The attributes in this section are valid only for NET units running within communication frontends.

**PN  Peer Node Number**

The node number of the peer NET in the redundant relation. The attribute has a meaning only if shadowing is in use (RM = 1 or 2). If RM = 1, the PN attribute is the number of the stand-by NET unit. If RM = 2, it is the number of the Hot NET.

- **Data type:** Integer
- **Value:** 0 ... 32. The node number of the peer NET
  - 0 = No redundancy
- **Default value:** NET nr + 1 for odd NET nrs, and NET nr - 1 for even NET nrs.
  - E.g. PN = 4 for NET3, and PN = 3 for NET4
- **Access:** No restrictions

**RM  Running Mode**

The redundancy mode of the NET unit. Using this attribute, the redundancy mode can be changed during operation.

- **Data type:** Integer
- **Value:**
  - 0  Single, no redundancy
  - 1  Redundant Hot
  - 2  Redundant stand-by
- **Default value:** The initial mode is read from the common RAM of the communication frontend, where it can be written by MFL (the CMOD parameter in MFLCONF.DAT). If no initial mode is found there, the default mode is SINGLE (RM = 0).
- **Access:** Not preconfigurable, otherwise no restrictions

**SH  Shadowing State**

The shadowing state of the NET unit - sending or receiving state. By setting the SH attribute to 0 (= no shadowing), the shadowing is stopped and the data transfer from the hot FE to the stand-by FE is disabled.

- **Data type:** Integer
- **Value:**
  - 0  None. No shadowing
  - 1  Sending
  - 2  Receiving
- **Default:** 0
- **Access:** No restrictions
MicroSCADA Pro
System Objects
Technical Description

**XC** Redundant Frontend Diagnostic Cycle
The time delay in the redundant relation that the peer NET inquires after the state of the Hot NET.
- Data type: Integer
- Value: Integer
- Unit: Seconds
- Default: 3 seconds
- Access: No restrictions

**XS** Redundant Frontend Diagnostic Counters
Defines the type and number of the messages sent between peer NET and the Hot NET.
- Data type: Vector
- Value: Vector of 10 integers. Each element is the value of a diagnostic counter.
- Indexing: 1 ... 10, counter number
- Access: Read-only, the values can be reset

**XT** Redundant Frontend timeout
The maximum time in milliseconds that the peer NET waits for the reply from the Hot NET.
- Data type: Integer
- Value: Integer
- Unit: Milliseconds
- Default value: 1000 ms
- Access: Read-only, the values can be reset

14.7. Miscellaneous NET device attributes

**FM** Free Memory
The amount of free memory in the NET unit.
- Data type: Integer
- Value: Integer
- Unit: Bytes
- Access: Read-only
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 preconfigurable, otherwise 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

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 set to another value than 0 previously).

Data type: Vector
Value: A vector of 250 real or integer elements
Indexing: 1 ... 250 real elements, 1001 ... 1250 integer elements
Access: No restrictions

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

TM  Time
The time of the NET unit in MicroSCADA time format (32 bits integer). Using this attribute, the time can be written manually. The attribute is used for the synchronization of NET from the base system (if the radio clock resides in the base system), or for synchronization of the base system from NET unit. The base system...
is synchronized from NET unit when the radio clock is connected to NET unit. If there is a radio clock in the base system computer, the time of an internal NET unit is automatically set according to the radio clock time.

Data type: Integer, vector
Value: Integer or vector of three integer elements
Indexing: No index or MicroSCADA 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 MicroSCADA time
Index 3: Increments the current NET time with the given number of milliseconds

Example:
The NET time is set equal with the MicroSCADA time on second level:

#SET NET1:STM = CLOCK

The NET time is set in accordance with the MicroSCADA time on millisecond level and incremented with 50 milliseconds, which is an estimation of the time consumed for transmission and execution of the command. An exact time setting, taking into account the execution time, requires a more extensive procedure:

@A = HR_CLOCK
#SET NET1:STM(1..2) = (A:VCL,(A:VUS DIV 1000))
#SET NET1:STM3 = 50

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, ADLP 180, 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
15. **NET lines for communication system**

This chapter describes the configuration of NET lines and the NET line attributes. The chapter is divided in the following sections:

15.1 **General**

NET communication lines

Each DCP-NET has 8 asynchronous serial lines numbered 1 ... 8. In addition, the communication program recognises a line number 13 for common RAM interface. This line is used for the communication between an internal NET unit and the base system. It is also used between a NET unit in a communication frontend and a base system connected to the communication frontend, and between the separate communication units in a communication frontend.

The PC-NETs communicate through the serial ports (COM ports) of the host computer and possible PCLTA cards (one or two) or a RocketPort. The COM ports, if used, represent by default NET line numbers 1 to 4. The line number is the same as the number of the COM port. 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 if no COM ports are used). The connection between the NET line numbers and the LonWorks channels are defined by a NET line attribute, the SD attribute. A PC-NET communicates with the base system (kernel) through line number 13, which is a software link.

15.2 **Basic Line Attributes** (PO, SD, IU, LK, PM, PS, MI, MS).

15.3 **Data Transmission Attributes** (BR, ER, PY, RD, RE, SB, TD).

15.4 **Communication Control Attributes** (DE, EN, HT, NA, OS, PD, RI, RK, RL, SG, TI, TI, TD).

15.5 **Polling Attributes** (AW, CP, PD, PL, PP, PT, RP).

15.6 **Autodialling Attributes** (AC, AS, CL, CN, CS, CT, DD, MC, PU, RC, RW, SR).

15.7 **RP570 Communication Loop Attributes** (BO, BU, CF, DR, LU, LS, MD, MT).

15.8 **Lon Configuration Attributes** (NC, XA).

15.9 **Miscellaneous Attributes**: Diagnostic counter (DC) and clock synchronization attributes (LK, SF, SS).

15.10 **DNP V3.00 Protocol**: Link Layer Attributes (ML, XR, TI, LA).

The attributes in this description cover mainly the protocols listed in Table 15.1.-1. The attributes of other protocols are described in separate documents, which are available on request.

15.1. **General**

**NET communication lines**

Each DCP-NET has 8 asynchronous serial lines numbered 1 ... 8. In addition, the communication program recognises a line number 13 for common RAM interface. This line is used for the communication between an internal NET unit and the base system. It is also used between a NET unit in a communication frontend and a base system connected to the communication frontend, and between the separate communication units in a communication frontend.

The PC-NETs communicate through the serial ports (COM ports) of the host computer and possible PCLTA cards (one or two) or a RocketPort. The COM ports, if used, represent by default NET line numbers 1 to 4. The line number is the same as the number of the COM port. 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 if no COM ports are used). The connection between the NET line numbers and the LonWorks channels are defined by a NET line attribute, the SD attribute. A PC-NET communicates with the base system (kernel) through line number 13, which is a software link.
PC-NETs support the following protocols on the COM lines (lines 1 - 4): ACP, SPA, RP570/571 master. On the LONWORKS lines, only the LonTalk protocol is supported. The PC-NET have the same attributes as the board based NETs, except for the line protocols and the station types that are not supported.

The DCP board based NETs does not support the LonTalk protocol.

**Auto-dialing**

Auto-dialing can be used on all NET serial lines defined for the ANSI X3.28 Half Duplex or Full Duplex protocols, ACP (MicroPROTOCOL), Modbus, IEC 1107, RP570 protocol, Alpha or the protocols of IEC 60870 family.

**Line definition**

Using a NET line - serial line or common RAM - requires that it is defined in the NET in question. A NET unit line is basically defined by assigning it a protocol. This can be done in the preconfiguration (DCP-NETs) or on-line with the PO attribute described in Section 15.2.

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 15.1.-1 lists the line protocols and the applicable line attributes.

When a line is defined on-line, its attributes get the protocol specific default values as given in the attribute descriptions. For preconfigured lines, only those attributes that cannot be preconfigured get the default values.

Generally, the line attributes can be changed only while the line is out of use (the In Use attribute, IU, is 0).

**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 15.1.-1 An overview of the NET line attributes.

<table>
<thead>
<tr>
<th>Protocol Attributes</th>
<th>RAM</th>
<th>ACP and ANSI full duplex</th>
<th>Integrated link</th>
<th>SPA</th>
<th>LonWORKS</th>
<th>RP570/571 Master</th>
<th>RP570 Slave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic attributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Basic definition</td>
<td>PO</td>
<td>PO IU MS, MI PS</td>
<td>PO IU MS, MI PK</td>
<td>PO</td>
<td>PO IU MS, MI PS</td>
<td>PO IU MS, MI PK</td>
<td></td>
</tr>
<tr>
<td>- In Use</td>
<td>IU</td>
<td>IU MS, MI PK</td>
<td>IU MS, PK</td>
<td>IU</td>
<td>IU MS, PK</td>
<td>IU MS, PK</td>
<td></td>
</tr>
<tr>
<td>- System Message</td>
<td>MS</td>
<td>MS, MI PK</td>
<td>MS, PK</td>
<td>MS</td>
<td>MS, PK</td>
<td>MS, PK</td>
<td></td>
</tr>
<tr>
<td>- Other</td>
<td>MI</td>
<td>MI PK</td>
<td>MI PK</td>
<td>MI</td>
<td>MI PK</td>
<td>MI PK</td>
<td></td>
</tr>
<tr>
<td>Data transmission</td>
<td></td>
<td>RE BR, ER, RD SB, TD</td>
<td>BR, PY, RD SB, TD</td>
<td>BR, PY, RD SB, TD</td>
<td>BR, PY, RD SB, TD</td>
<td>BR, PY SB, TD</td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polling</td>
<td></td>
<td>PD, PP, PR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td></td>
<td>BO, BU, CF DR, LU, LS MD, MT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loops</td>
<td></td>
<td>NC, XA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LonWORKS Conf.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous</td>
<td></td>
<td>DC DC DC DC DC DC DC DC DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Diagnostics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Clock Synchronizer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocol Attributes</th>
<th>ANSI half duplex</th>
<th>ASCII printer</th>
<th>General ASCII</th>
<th>Modbus</th>
<th>IEC 60870-5 (-101 and -103)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic attributes</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td></td>
</tr>
<tr>
<td>- Basic definition</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td></td>
</tr>
<tr>
<td>- In Use</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td></td>
</tr>
<tr>
<td>- System Message</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td></td>
</tr>
<tr>
<td>- Other</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td>PO IU MS, MI PK</td>
<td></td>
</tr>
<tr>
<td>Data transmission</td>
<td>BR, ER, RD SB, TD</td>
<td>BR, PY, RD SB, TD</td>
<td>BR, PY, RD SB, TD</td>
<td>BR, PY, RD SB, TD</td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td>DE, HT, RL HT PD, TI</td>
<td>DE, HT, RL HT PD, TI</td>
<td>DE, HT, RL HT PD, TI</td>
<td>DE, HT, RL HT PD, TI</td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>DE, HT, RL HT PD, TI</td>
<td>DE, HT, RL HT PD, TI</td>
<td>DE, HT, RL HT PD, TI</td>
<td>DE, HT, RL HT PD, TI</td>
<td></td>
</tr>
<tr>
<td>Polling</td>
<td>AW, CP, PD PP, RP</td>
<td>PD</td>
<td>PD</td>
<td>PD, PP, PL RP</td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LonWORKS Conf.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Diagnostics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Clock Synchronizer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
15.2. Basic line attributes

15.2.1. 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.

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. If the line is defined in the preconfiguration (DCP-NETs), only those attributes that cannot be preconfigured get the default values.

The DCP-NET program supports the following protocols (plus some other protocols):

- ACP, Application Communication Protocol. This protocol is used for the communication between MicroSCADA 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.

- PAC-5. Defined as one link type of ANSI X3.28.

- COMMON RAM protocol. This protocol is used for communication between an internal NET unit and the base system. It is also used between each NET unit in a communication frontend and the base systems connected to the communication frontend, and between the communication units in a communication frontend. It can be used only on line number 13.

- 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.
MicroSCADA Pro
System Objects

Technical Description

- The P214 protocol (Indactic 35). The protocol is used for communication with P214 RTUs (Indactic 35).
- ADLP180 Slave for communication with Sindac and S.P.I.D.E.R. SCADA. The protocol can be used only on one line per NET.
- ADLP180 Master for communication with Collector 100 and 200.
- COML1 Slave protocol for communication with SATTCON systems. One line per NET can be used with this protocol.
- 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 1107 for metering device communication.
- IEC 60870-5 for communication with external device which uses the IEC 60870-5-103 or IEC 60870-5-101 protocol.
- DNP V3.00 slave and master protocols.

Of the above mentioned protocols, PC-NET supports the ACP, SPA, and RP570 master protocols on the NET line numbers 1 ... 8. When any of the NET lines 1 ... 8 are assigned any of these protocols, the PC-NET reserves the COM port with the same number for the protocol. The line number cannot be used for the LONWORKS output channels.

In addition to three mentioned protocols, PC-NET supports the following protocols, which are not supported by DCP-NET:

- 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)
  3 Common RAM protocol, allowed only on line 13(RM)
  4 ASCII protocol for printer (or printer simulating device) (AS)
  6 COM-1
  7 RP570 protocol (SR)
MicroSCADA Pro
System Objects
Technical Description

<table>
<thead>
<tr>
<th>Line</th>
<th>Protocol/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>ADLP80 slave protocol (AD)</td>
</tr>
<tr>
<td>9</td>
<td>P214 protocol (PR)</td>
</tr>
<tr>
<td>10</td>
<td>ADLP180 slave protocol (AL)</td>
</tr>
<tr>
<td>11</td>
<td>COMLI slave (CL)</td>
</tr>
<tr>
<td>12</td>
<td>LCU500 (LC)</td>
</tr>
<tr>
<td>13</td>
<td>ADLP180 Master (AM)</td>
</tr>
<tr>
<td>14</td>
<td>SPA protocol (SP)</td>
</tr>
<tr>
<td>15</td>
<td>General ASCII (AG)</td>
</tr>
<tr>
<td>16</td>
<td>RP570 Slave (RP)</td>
</tr>
<tr>
<td>17</td>
<td>RCOM (Procontic) (PC)</td>
</tr>
<tr>
<td>18</td>
<td>Westinghouse F4F</td>
</tr>
<tr>
<td>19</td>
<td>Mastercom</td>
</tr>
<tr>
<td>20</td>
<td>IDLC</td>
</tr>
<tr>
<td>21</td>
<td>CCU</td>
</tr>
<tr>
<td>22</td>
<td>TRL-Mastercom</td>
</tr>
<tr>
<td>23</td>
<td>ABB Alpha meter protocol</td>
</tr>
<tr>
<td>24</td>
<td>General PLC protocol</td>
</tr>
<tr>
<td>25</td>
<td>Modbus RTU master protocol</td>
</tr>
<tr>
<td>26</td>
<td>IEC 1107 protocol</td>
</tr>
<tr>
<td>27</td>
<td>the LonTalk protocol (only PC-NETs)</td>
</tr>
<tr>
<td>28</td>
<td>Integrated link protocol (used on line 13 of PC-NETs)</td>
</tr>
<tr>
<td>29</td>
<td>IEC 60870-5-101 Unbalanced slave</td>
</tr>
<tr>
<td>30</td>
<td>IEC 60870-5-101 Balanced slave</td>
</tr>
<tr>
<td>31</td>
<td>IEC 60870-5-101 Unbalanced master</td>
</tr>
<tr>
<td>32</td>
<td>IEC 60870-5-101 Balanced master</td>
</tr>
<tr>
<td>33</td>
<td>IEC 60870-5-103 Unbalanced master</td>
</tr>
<tr>
<td>35</td>
<td>DNP V3.00 slave protocol</td>
</tr>
<tr>
<td>43</td>
<td>DNP V3.00 master protocol</td>
</tr>
<tr>
<td>44</td>
<td>IEC 60870-5-104 master</td>
</tr>
<tr>
<td>45</td>
<td>IEC 60870-5-104 slave</td>
</tr>
</tbody>
</table>

Indexing: Line number
Access: Read, conditional write

Example:

```plaintext
#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.

**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 of the PCLTA card (each LONWORKS channel) is associated with a specific device name (see the Installation Manual, Chapter 6). 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).

Using the SD attribute, it is possible to override these default values.

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)

**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"
```
15.2.2. Other basic attributes

**IU**  In Use

The state of the line – is it 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.

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

**LK**  Link Type

The type of data link connection used on the line. The attribute has no meaning for the RAM interface and printer lines, nor for LONWORKS lines. (LonTalk protocol lines do have a LK attribute but this has another meaning.)

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.
## System Objects

### Technical Description

**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) and the Modbus master protocol (PO = 25).

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Modem signals disabled. CTS is not used, and the transmission starts after a finite delay given by the DE attribute.</td>
</tr>
<tr>
<td>10</td>
<td>RTS is set and transmission starts immediately. No other modem signals are issued. NET does not wait for CTS signal. Used on optical lines.</td>
</tr>
<tr>
<td>14</td>
<td>Full duplex. No collision avoidance, Data Carrier Detect signal is handled as in other protocols.</td>
</tr>
<tr>
<td>15</td>
<td>Half duplex. Collision avoidance on, transmission when the Data Carrier Detect signal of the line is not set.</td>
</tr>
</tbody>
</table>

**Indexing:** Line number  
**Default value:** 0  
**Access:** Read, conditional write

**Additional Information**

For DNP V3.00 protocol, PM = 15.

- **RP570/571 master:**
  - 0: RP570  
  - 1: RP571

- **RCOM:**
  - 0: Event polling disabled  
  - 1: Event polling enabled

- **Modbus master:**
  - 0: Modbus RTU mode  
  - 1: Modbus ASCII mode

**Value:** General ASCII:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Receive only. This mode is used for ADEMCO alarm receiver.</td>
</tr>
<tr>
<td>5</td>
<td>Time Synchronization receive only. This mode is used for all clock synchronization receivers, except TAIP.</td>
</tr>
<tr>
<td>6</td>
<td>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</td>
</tr>
</tbody>
</table>

**Indexing:** NET line number
**MicroSCADA Pro**  
**System Objects**  
**Technical Description**

**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 is 228 bytes.

**Default value:**
- General ASCII: 5
- RP570 and RCOM: 0

**Access:** Read, conditional write

**Value:**
- RP570 slave lines: 1..250
- LONWORKS lines: 1..250

**Indexing:** NET line number

**Suggested value:**
The optimum value depends on the line protocol and the amount of RAM in the communication unit. Some rules of thumb:

- Integrated link: 200 (default 200)
- Common RAM interfaces: 50
- ANSI X3.28 Full and Half Duplex: 50 (default 50)
- ACP lines: 50 ... 100
- RP570 lines: 10 ... 30
- P214 lines: 30 ... 50
- LONWORKS lines: 1 ... 250

A low value like 50 may cause PC-NET crash, if several LON lines are used simultaneously.

**Printer lines:**
- Alarm printers 20
- Color printers 30

An undersized buffer pool will cause unnecessary retransmissions because the line runs out of buffers. On the other hand, an oversized value consumes more RAM in communication frontend without improving the line performance.

**Access:** Read, conditional write

### 15.2.3. 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 autodialling lines).
- When no time synchronization message is received from a General ASCII line.

Refer to Chapter 14 and the System Configuration manual to learn how to handle generated system messages.
MI  
Message Identification
Object address of system messages. See the MI attribute in Chapter 14.

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

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 ... 32. The APL object number as known to the communication unit.
Indexing: NET line number
Default value: 1
Access: Read, conditional write

15.3.  
Data transmission attributes

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
Suggested value: 300, 600, 1200, 2400, 4800, 9600 or 19200. NET allows any value in the specified range to be used (e.g. 8965), but usually the setting alternatives of the peer device are limited to the ones in the list of recommended values.
For printer connections 2400 bits/s is recommended.
Default baud rate is 19200 for IEC 60870-5-103 and 2400 for IEC 60870-5-101.
General recommendation: the sum of the speeds on all serial lines should be max. 30 kbits/s.
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] |
```

**ER - Embedded Response**

Indicates if NET transmits embedded responses (ACK, NAK) and ENQ:s within the messages, see the illustration in Fig. 15.3.-1. 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

**Fig. 15.3.-1 An illustration of the ER attribute**
**MicroSCADA Pro**

**System Objects**

**Technical Description**

<table>
<thead>
<tr>
<th>PY</th>
<th>Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>The parity check (if any) used for the characters transferred on the line. The attribute is essential for all types of protocols, except common RAM.</td>
<td></td>
</tr>
<tr>
<td>Data type:</td>
<td>Integer</td>
</tr>
<tr>
<td>Value:</td>
<td>0 No parity check</td>
</tr>
<tr>
<td></td>
<td>1 Odd parity</td>
</tr>
<tr>
<td></td>
<td>2 Even parity</td>
</tr>
<tr>
<td>Indexing:</td>
<td>Line number</td>
</tr>
<tr>
<td>Suggested value:</td>
<td>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. RP570, ADLP80, and SPA lines use even parity. On printer lines there is usually no parity check. Default value for IEC 60870-5 is even.</td>
</tr>
<tr>
<td>Access:</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RD</th>
<th>Receiver Data Bit Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>The number of data bits in each received character. The attribute is valid for all protocols, except common RAM.</td>
<td></td>
</tr>
<tr>
<td>Data type:</td>
<td>Integer</td>
</tr>
<tr>
<td>Value:</td>
<td>5, 6, 7 or 8</td>
</tr>
<tr>
<td>Unit:</td>
<td>Data bits</td>
</tr>
<tr>
<td>Indexing:</td>
<td>Line number</td>
</tr>
<tr>
<td>Suggested value:</td>
<td>ANSI X3.28, IEC 60870-5 and RP570 lines: 8 data bits</td>
</tr>
<tr>
<td></td>
<td>SPA lines: 7 data bits</td>
</tr>
<tr>
<td>Access:</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RE</th>
<th>Redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 and common RAM protocols. On ADLP80, P214 and RP570 lines, the checksum is always BCC.</td>
<td></td>
</tr>
<tr>
<td>Data type:</td>
<td>Integer</td>
</tr>
<tr>
<td>Value:</td>
<td>0 No checksum</td>
</tr>
<tr>
<td></td>
<td>1 CRC-16 (Cyclic Redundancy Check). Possible only on ANSI X3.28 lines.</td>
</tr>
<tr>
<td></td>
<td>2 BCC (Block Check Character)</td>
</tr>
</tbody>
</table>
SB Stop Bits

The number of stop bits attached to each transmitted byte. The attribute applies to all protocols, except common RAM.

Data type: Integer
Value: 1 … 3
Unit: Bits

Suggested value: On ANSI X3.28, IEC 60870-5 and RP570 lines: one stop bit is used.

Indexing: Line number
Access: Read, conditional write

TD Transmitter Data Bit Count

The number of data bits in each transmitted character. The attribute is essential for all protocols, except common RAM.

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

Suggested value: ANSI X3.28, IEC 60870-5 and RP570 lines: 8 data bits
SPA lines: 7 data bits
Printer lines: Normally 8 data bits

Indexing: Line number
Access: Read, conditional write

15.4. Communication control attributes

For more specific information about attributes in newer protocols, see the manuals for each protocol (for instance DNP v3.00, IEC 60870-5-101 or IEC 60870-5-103).

DE CTS Delay

Time delay between the activation of the RTS signal (Request to Send) and the start of a new transmission, see Fig. 15.4.-1. 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 communication frontend starts transmitting. If $DE = 0$, the communication frontend will transmit immediately, when CTS is activated. If $DE$ is larger than 0, the communication frontend waits the time indicated by $DE$. When this time has run out, the communication frontend transmits under the condition that 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

**DE - CTS Delay**

Fig. 15.4.-1 An illustration of the DE attribute on ANSI Half Duplex and RP570 lines
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.

Value: 0... 20
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 write operation is seen as complete when there is still one byte to be sent. Thus, value 0 will cause the RTS to be in nonsignaled state before the message is completely sent.

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.

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. After this the transmission has failed and the communication frontend moves on to the following message to be
transmitted onto the line. 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

**HT** **Header Timeout**

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. Fig. 15.4.-2 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 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: 1200 ms, 2000 ms for DNP V3.00 protocol
Access: Read, conditional write
NA NAK Limit

The number of NAK responses that the communication frontend accepts at the transmission of a message, without considering the transmission failed. The attribute is essential only for ANSI X3.28 Full Duplex, ACP and RAM lines (PO = 1 or 3).

Data type: Integer
Value: 1 ... 255
Indexing: Line number
Suggested value: 3
Access: No restrictions

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
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
**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.

- **Data type:** Integer
- **Values:** 0 ... 15
- **Indexing:** Line number
- **Access:** Read, conditional write

**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 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

**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 Fig. 15.4.-3.

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 in the communication frontend 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.

- **Data type:** Integer
- **Value:** 0 ... 255. Number of padding characters
MicroSCADA Pro
System Objects
Technical Description

Unit: Padding characters
Indexing: Line number
Default value: 0
Access: Read, conditional write

**RK - RTS Keep Up Padding Characters**

| RK = 3: | Data | NUL | NUL | NUL |

![Illustration_RK_attribute.eps](attachment:Illustration_RK_attribute.eps)

*Fig. 15.4.-3 An illustration of the RK attribute*

**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: Not preconfigurable, otherwise no restrictions

**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.

Data type: Integer
Value: 0 Passive
1 Active
Read value: DCD and CTS signals
Write: DTR signals
Indexing: 100 * line nr + signal no. Signal no. 5 = CTS, 8 = DCD, 20 = DTR
Access: Read-only

**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, RAM, 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 Fig. 15.4.-2.

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 
  \((11 \times (\text{byte count in longest message})) / (\text{transmission rate}))\)  
- 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

### 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.

**Data type:** Integer  
**Values:** 0 ... 15  
**Indexing:** Line Number  
**Access:** Read, conditional write
**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.

**TW**

**Acknowledge timeout with 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
MicroSCADA Pro
System Objects
Technical Description

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 preceding delay in the transmission.
- each SPACE character at the beginning of the message results in a 1-byte preceding 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 length 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 preceding 30-byte delay in transmission
#SET NETx:STB'line' = " @1" ; send "$1" with preceding 3-byte delay in transmission
#SET NETx:STB'line' = (0,32,32,32,32,64,49) ; send "$1" with preceding 15-byte delay in transmission
```

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 15.5.-1 Polling of an ANSI station:**

<table>
<thead>
<tr>
<th>NET</th>
<th>STA1</th>
<th>STA2</th>
<th>STA3</th>
<th>STA4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLE ENQ 1 BCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE EOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 2 BCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>DLE EOT</td>
</tr>
<tr>
<td>REPLY TO 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 3 BCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE EOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 4 BCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE EOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the message from the base system to STA4 is a command, the polling continues as follows:

**Table 15.5.-2 Polling of an ANSI station after a command to STA4 from the base system:**

<table>
<thead>
<tr>
<th>NET</th>
<th>STA1</th>
<th>STA2</th>
<th>STA3</th>
<th>STA4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLE ENQ 1 BCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE EOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 2 BCC</td>
<td></td>
<td></td>
<td>DLE EOT</td>
<td></td>
</tr>
<tr>
<td>COMMAND TO 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 4 BCC</td>
<td></td>
<td>DLE EOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 4 BCC</td>
<td></td>
<td>DLE EOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 4 BCC</td>
<td></td>
<td>DLE EOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ACK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 3 BCC</td>
<td></td>
<td></td>
<td>DLE EOT</td>
<td></td>
</tr>
<tr>
<td>DLE ENQ 4 BCC</td>
<td></td>
<td></td>
<td>DLE EOT</td>
<td></td>
</tr>
<tr>
<td>etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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: Not preconfigurable, otherwise no restrictions

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

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
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

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 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.
MicroSCADA Pro

System Objects

Technical Description

IEC 60870-5:

This attribute is used only with unbalanced master protocol mode IEC 60870-5-101

Data type:

Integer

Value:

0 ... 255

Indexing:

Line number

Suggested value:

2 ... 10

Default value:

10 for IEC 60870-5

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 communication frontend will poll it 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.

PR

This attribute is not used.

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 Chapter 16.

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:** Not preconfigurable, otherwise no restrictions

**Example:**

Each poll cycle on line 3 will contain 2 data polls:  

```plaintext  
#SET NET1:SPT303 = 2  
```

**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
15.6. Autodialling attributes

The following attributes are significant only when an autocaller (a modem with functions for automatic dial-up) is used. Auto-dialing is possible only on ACP, ANSI X3.28, RP570, Modbus, IEC1107, Alpha and the protocols of IEC 60870 family. The autocaller attributes are not preconfigurable, they are always set with SCIL. The link type (the LK attribute) must be 1 or 2. Dial-up may be initiated by NET or by a station.

### AC

**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:     | No autocaller |
|           | Yes, autocaller is connected |

| Default value: | 0 |
| Access:        | Not preconfigurable, otherwise no restrictions |

### AS

**AS ACE State**
The state of the autocaller.

| Data type: | Integer |
| Value:     | IDLE, the ACE is ready to make a call |
|           | CONNECTED, transmission is activated |
|           | BUSY, the ACE is busy, e.g., dialling |
|           | INITIAL, the ACE is uninitialized |
|           | CONFIGURE, the IU attribute of the line is set to 0 |

<p>| Access: | Read-only |</p>
<table>
<thead>
<tr>
<th>CL</th>
<th>Connection Time Limited</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Data type:</td>
<td>Integer</td>
</tr>
<tr>
<td>Value:</td>
<td>0 No time limit</td>
</tr>
<tr>
<td></td>
<td>1 Time limit</td>
</tr>
<tr>
<td>Default value:</td>
<td>1</td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CN</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialling devices from NET and for breaking telephone connections. Using CN presupposes that there is an autocaller connected to the line (AC = 1).</td>
<td></td>
</tr>
<tr>
<td>A call to a station or workstation is initiated by writing the phone number to the CN attribute. NET then commands the autodialling 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.</td>
<td></td>
</tr>
<tr>
<td>When dialling a station, the station number should be given at the end of the phone number string, preceded by the letter &quot;S&quot;. This option is normally used to increase the communication performance on multidrop lines.</td>
<td></td>
</tr>
<tr>
<td>Data type:</td>
<td>Text</td>
</tr>
<tr>
<td>Values:</td>
<td>Max. 25 characters:</td>
</tr>
<tr>
<td></td>
<td>Dialling: The telephone number, and possible the station number preceded by an S. A leading zero and fill characters such as &quot;/&quot;, &quot;,&quot; and &quot; &quot; may be used. By inserting comma characters (&quot; &quot;), 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.</td>
</tr>
<tr>
<td></td>
<td>Breaking the connection: An empty string (&quot; &quot;)</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

**Example:**

```
#SET NET1:SCN5 = "123456789S11"
```
MicroSCADA Pro
System Objects
Technical Description

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

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
Unit: Seconds
Access: No restrictions

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

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. The response from the modem is read using the same attribute.

Data type: Text
Value: Write: An AT/Hayes command
       Read: The response from the latest command
Access: Not preconfigurable, otherwise no restrictions

Example:
The current S0 setting of the modem, e.g. "002" is shown:

_SET NET1:SMC3 = ("AS0?")'
!SHOW RESP NET1:SMC3
PU  Pulse Dialing

The dialing principle used.

Data type:      Integer
Value:
   0  Tone dialling
   1  Pulse dialling
Default value:  0
Access:         No restrictions

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 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

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

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

15.7. **RP570 communication loop attributes**

The purpose of the communication loop is to provide redundancy in the connection between the NET and the RTUs using one single cable, see Fig. 15.7.-1. A communication loop starts out from one NET line, connects all the RTUs and ends up to another NET line. Each RTU can be accessed from either direction. The ends of a loop are connected to separate NETs. The loop has a breakpoint at one of the RTUs. The RTUs on one side of the breakpoint are polled from that direction, the RTUs on the other side are polled from the other direction.

The RTUs are connected to the loop via a Loop Reversal Unit. Each RTU can control its own Loop Reversal Unit via two modem signals. One signal opens the loop, i.e., a breakpoint is located at the RTU. This means that no messages are forwarded to the next RTU. The other signal closes the loop, i.e. the messages from the NET are forwarded to the next RTU and vice versa. The loop reversal units of the last RTUs of each polling direction are open, all others are closed.

If an RTU does not receive a poll message from the NET unit within a specified time, it automatically turns its load reversal unit to listen in the other direction. This happens provided that the RTU is configured for loop communication). The loop reversal units can also be changed from SCIL.

The communication loop attributes cannot be preconfigured.

Communication loops are configured and controlled using an application package built for this purpose. See the System Configuration manual.
**BO  Break Down**

Starting, stopping and reading the "break-down state" of the communication loop. During the break-down state, NET polls through all RTUs in the communication loop. If RTUs do not respond, system messages are sent to the application. The system messages can be used in SCIL programs for deciding where to place the breakpoint and for specifying a configuration table (the CF attribute). The breakdown state is exited when the BO attribute is set to 0.

The loop configuration tables are loaded to NET using the CF attribute and the loop configuration is taken into use by the BU attribute.

**Data type:** Integer  
**Value:**  
1 The break-down state is started (write) or it is going on (read)  
0 The break-down state is stopped (write) or it is not going on (read)  
**Access:** No restrictions

**BU  Build Up**

Starting, stopping or reading the loop configuration build-up mode of the line. When the build-up mode is started, the CF table (see below), which must have been set before the BU attribute, is taken into use.

The loop configuration is realized so that the last RTU of each polling direction in the loop is commanded to open the loop, while all other RTUs are commanded to close the loop. Both NETs try to establish contact with their respective RTUs (all RTUs that are in use) by sending SCI (Status Request) commands. If it succeeds, a system message (Configuration_ready) is sent to the application. No response is received from one or more RTUs within the time specified by the MT attribute as an
indication that the contact between NETs and their respective RTUs did not succeed. If it did not succeed, the build-up mode is exited and a system message (Configuration_Failure) is sent to the application.

Data type: Integer
Value: 1 Building up mode
0 Normal operation mode
Default value: 0
Access: No restrictions

**CF Loop Configuration Table**

The physical order and the polling direction of the RTUs on the loop.

The attribute specifies all RTUs in the loop and their polling direction. All RTUs up to the breaking point will be polled from one direction and, the rest are polled from the other.

When this attribute is set, the RTU polling order and direction is loaded to NET. The loading of the RTU order stops the polling on the line. After the loading, a Build Up command (BU attribute) must be given to activate the loop configuration.

Data type: Integer
Value: Integer vector where the odd elements are station identification numbers and the following even elements specify the polling direction of the stations

Indexing: Odd indices: LN = Logical number of RTU
Even indices: The polling direction of the RTU:
1 Station polled from this direction
0 Station not polled from this direction

The loop switch of the last RTU in the vector that will be opened.

Access: Write-only

**Example:**

The loop contains five RTUs. The three first ones are polled from NET1, line 2 and the fourth and fifth stations are polled from the other end of the loop:

```
*SET NET1:SCF2 = (1,1,2,1,3,1,4,0,5,0)
```

**DR Direction**

The loop direction used on the line. The application engineer specifies two directions, one called "A" and the other "B". One frontend line of the loop has direction "A" and the other has direction "B".

Data type: Text
Value: "A" or "B"
Access: No restrictions
LU Loop Mode In Use

Specifies if the loop mode is in use or not on the line. The other communication loop attributes are meaningful only if LU = 1.

Data type: Integer

Value:
- 0: Not in use
- 1: In use

Default value: 0

Access: No restrictions

LS Loop Status

The polling state of a line in communication loop mode.

Data type: Integer

Value:
- 0 ... 3:
  - 0: OK. Normal polling
  - 1: BREAK DOWN. The line is in break-down mode. Each RTU with the same polling direction as the NET line is polled in its turn.
  - 2: BUILD UP. The line is in build up mode. It tries to open a connection to each RTU with the same polling direction as the NET line.
  - 3: POLLING DISABLED. The RTU order on the loop is being configured. Polling is disabled, and will be started by writing to attribute BU.

Access: Read-only

MD Modem Delay

The number of modem pairs between NET and the RTU.

On communication loop lines, time synchronization messages can not be sent as broadcast, because the modems are connected in series. This fact causes a delay, which is proportional to the number of modem pairs between the NET and the RTU. The estimated mean delay caused by one pair of modems can be written to the MD attribute, whereby it is automatically taken into account when time synchronization commands are sent to the RTUs. The RP570 protocol code in NET calculates with a modem delay, which is MD * the number of modem pairs between NET and the RTU.

Data type: Integer

Value: 0 - 1000 ms

Unit: Milliseconds

Default value: 30 ms

Access: No restrictions
**MT**  
Maximum Scan Time

The maximum time for RTU to response. If NET does not get response from an RTU within the number of seconds given by the MT attribute after a polling command, NET exits the Building Up mode. It also sends a CONFIGURATION_FAILED message to the application. The MT value should be at least twice the Loop Switch Turning Time of the RTU, which is 30 s by default. The Loop Switch Turning Time of the RTU can be set with an FTAB command.

Data type: Integer  
Value: 0 - 600  
Unit: Seconds  
Access: No restrictions

---

**15.8. LON configuration attributes**

**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™ 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 Chapter 16) 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:

<table>
<thead>
<tr>
<th>element</th>
<th>1: p</th>
<th>d</th>
<th>nv_selector_high</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:</td>
<td>nv_selector_low</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:</td>
<td>t</td>
<td>st</td>
<td>addr_index</td>
</tr>
<tr>
<td>4:</td>
<td>ext_addr_index</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

---

1. LONMARK is a trademark of Echelon Corporation.
More information can be found in the manual Connecting LONWORKS Devices to MicroSCADA.

**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
```

**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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: type</td>
<td>Type of address entry or 80H + group size</td>
</tr>
<tr>
<td>2: d node_or_member</td>
<td></td>
</tr>
<tr>
<td>3: rpt_timer retry</td>
<td></td>
</tr>
<tr>
<td>4: recv_timer tx_timer</td>
<td></td>
</tr>
<tr>
<td>5: subnet_or_group</td>
<td></td>
</tr>
</tbody>
</table>

where

- ‘type’ Type of address:
  - 0 = Unbound address table entry (if element 2 = 0)
MicroSCADA Pro

System Objects

Technical Description

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

15.9. Miscellaneous NET line attributes

15.9.1. Diagnostic counter

This attribute applies to ANSI X3.28 lines, RAM connections, RP570 lines, P214 lines, SPA lines, LonTalk protocol lines and printer lines. It is also available for IEC 60870-5-101 protocol.

For correct attribute information in different protocols, see the corresponding protocol manual.

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 autodialling). 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.

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.

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.

13. OVERRUN ERRORS
   Is incremented when a received message is rejected because of an overrun error.

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).

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. The counters 17 to 26 are used on autocaller lines as follows:

18. ACE_CONNECTIONS_LOCAL_ORIGIN
   Calls originating from the communication frontend.

19. ACE_CONNECTIONS_REMOTE_ORIGIN
   Calls originating from outside.

20. ACE_RECEIVED_ERROR_RESPONSES

21. ACE_RECEIVED_NO_CARRIER_RESPONSES

22. ACE_TIMEOUTS

23. ACE_FAILED_DIALINGS

24. ACE_FAILED_CONNECTINGS_OF_REMOTE_CALLS

25. ACE_FAILED_DISCONNECTIONS

26. ACE_IGNORED_RINGS

27. ACE_RECEIVED_RINGS

None of the line protocols updates all of the counters. The following counters are
updated for the ANSI, RAM, SPA, LON, IEC 60870 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
   COMMON RAM protocol: nr 1, 2, 3, 4, 5, 7 and 8
   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
   IEC 60870: 1, 2, 4, 5 and 11 - 16

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.
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

15.9.2. Clock synchronization attributes

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 behaviour 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
MicroSCADA Pro
System Objects
Technical Description

**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

**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):

```plaintext
#IF BIT(NET1:SSS2,7)==1 #THEN #BLOCK
........................;Clock not synchronized
#BLOCK_END
#ELSE  #BLOCK
........................;Clock synchronized
#BLOCK_END
```
15.10. DNP V3.00 protocol

15.10.1. Link layer attributes

**ML**  
Maximum Message Length  
Maximum length of a data link fragment (LPDU) in octets.

- **Data type:** Integer
- **Value:** 50 ... 249
- **Index range:** 1 ... 8 (NET line numbering)
- **Default value:** 230 for DNP V3.00 protocol
  249 for others
- **Access:** No restrictions

**XR**  
Maximum Random Delay for Retransmission

Simple collision avoidance method for full duplex lines. When unsolicited responses are enabled for slave stations on a multi-drop line, there is possibility that several slave devices send messages at the same time. This message collision is seen as timeout in a slave station since the master is not responding. XR attribute limits the possibility that two devices retransmit messages at the same time.

- **Data type:** Integer
- **Value:** 0 ... 65535
- **Unit:** Milliseconds
- **Index range:** 1 ... 8 (NET line numbering)
- **Default value:** 0
- **Access:** No restrictions

**TI**  
Response Timeout

The time in milliseconds that the DNP link waits for the end of the received message.

- **Data type:** Integer
- **Value:** 0 ... 65535
- **Unit:** Milliseconds
- **Index range:** 1 ... 8 (NET line numbering)
- **Default value:** 2 for DNP V3.00 protocol
  700 for others
- **Access:** No restrictions
### LA Link Layer Confirmations Enabled

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

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>0, 1</td>
</tr>
<tr>
<td>Index range:</td>
<td>1 ... 8 (NET line numbering)</td>
</tr>
<tr>
<td>Default value:</td>
<td>1</td>
</tr>
<tr>
<td>Access:</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>
16. STA objects for communication system

About this chapter

This chapter describes the STA objects and their attributes. The chapter is divided into eleven sections as follows:

16.1 General: The station types, the definition of STA objects, the object notation.

16.2 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).

16.3 STA attributes specific to ANSI stations (STA): 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).

16.4 STA attributes specific to S.P.I.D.E.R. RTUs and Collector (RTU): 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).

16.5 STA attributes specific to SPACOM stations (SPA): 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).

16.6 STA Attributes specific to REX stations: Basic Attributes (NN, SN, UN, UT), Session Handling (SC, SH, SI, SK, SR, SS), Process Communication (DA, RQ, SM, TQ, GI, GO, IL), Event Handling (EF, HI, HS, RM), Suspension Attributes (RT), SPA Point Definition (SP), File Transfer Handling (FO, FP).

16.7 STA Attributes specific to LMK type stations: Basic Definition Attributes (NN, SN, UT), Polling Attribute (CT), Process Communication (DA, GI, LM, RT), Diagnostic Attributes (DC, DI), LON Point Definition (LP).

16.8 STA Attributes specific to SPI type stations: Basic Attributes (SA), Configuration Attributes (FT), 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).

16.9 STA Attributes specific to IEC stations: Basic Attributes of the Application Layer (SA), Process Communication Attributes (CF, CO, EV, GI, RS, SD, SY, TD), Device Configuration Attributes (CA, CL,
16.1. General

Station types

All the station types that can be connected to NET unit were listed in Chapter 14. This chapter describes the attributes of the following station types:

• ANSI stations (STA): the process units that communicate with NET unit using the ANSI protocol, for example, Allen-Bradley PLCS and SRO. These types of stations can only be connected to DCP-NET units.

• RTUs: 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 MicroSCADA through a LONWORKS line and the PC-NET unit. The REX stations can only be connected to PC-NET units.

• LMK stations: LSG devices and other devices connected to the LONWORKS network through a standard LONWORKS interface (for example Weidmüller). The LMK stations can only be connected to PC-NET units.

• 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, which communicate with MicroSCADA through PC-NET unit by using the IEC Protocols. The IEC type stations can only be connected to PC-NET units.

• PLC stations: Relays and control devices, which communicate with MicroSCADA through DCP-NET unit by using Modbus Master protocol.

The attributes of other station types are described in separate documents.
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 in the preconfiguration of DCP-NET units (up to 20 stations) or on-line with the NET station definition attributes described in Chapter 14. 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.

When the STA objects are defined on-line, the STA attributes get the default values given in the attribute descriptions. When the objects are defined in the preconfiguration, only the attributes that cannot be preconfigured get the default values, the other attributes get the values given in the preconfiguration.

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 16.5). Provided that the broadcast stations have been mapped for the application, they can be accessed from SCIL.

Object notation

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

\[ \text{STAn:Sat} \]

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 Chapter 6. As mentioned above, NET unit knows the broadcast stations as STA0.

'at' The attribute name

16.2. Common STA attributes

The attributes that are described in this section apply to all station types.

16.2.1. Basic attributes

IU In Use

The operational status of the station - 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.

**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.

**Device reservation**

**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.
MicroSCADA Pro
System Objects

Technical Description

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 16.3.

Example:

```
#SET STA1:SAL=1
```

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. If AL is set to 1 in the preconfiguration, AS must also be set there.

The allocating application will receive all the spontaneous process data from the station (if message split, section 16.3, is used, also other applications will receive the messages).

Data type: Integer
Value: 0 ... 32. 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.

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 MicroSCADA main program automatically updates the validity stamp of the object values in the process database (the OS attribute). See the System Message Attributes in Chapter 14. 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 Chapter 14 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 the types ANSI, SPA, REX and LMK 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.

**MI Message Identification; ANSI, SPA, REX, LMK and IEC Stations**

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). They are stored only if there is a fictitious process object in the database with this object address.

Data type: Integer

Value: 1 ... 65535, process object addresses

Default value: 1000 + STA object number
29000 + station number for IEC 60870-5 protocol
This default value can be used as such (copied to the process object address), or it can be changed.
MicroSCADA Pro
System Objects
Technical Description

Example: See the MI attribute in Chapter 14
Access: No restrictions

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

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 ... 32, or vector of six integers in the range 1 ... 32. 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 station 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.
MicroSCADA Pro
System Objects
Technical Description

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

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.
1.6. STA attributes, ANSI stations

Besides the common attributes described in Section 16.2, the STA objects of type ANSI stations (STA) have the attributes described in this section.

1.6.1. Basic attributes

**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

**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 Chapter 14. 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

**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
Polling attributes

The attributes in this section apply exclusively to stations on ANSI half-duplex lines.

**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:** Not preconfigurable, otherwise no restrictions

A high value may slow down the communication to other stations, for example in hardware fault situations.

**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:** Not preconfigurable, otherwise no restrictions

The attribute should always be given a value when a new half-duplex station is created on-line.
### 16.3.3. 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 out-dated, 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 16.3.3-1 shows an overview of the reasons for suspension and the states that cause recovery from the suspension.

**Table 16.3.3-1 Reasons for station suspension and recovery**

<table>
<thead>
<tr>
<th>Suspension Reason</th>
<th>Recovery Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>No ACK to replay</td>
<td>ACK to command or reply</td>
</tr>
<tr>
<td>No reply in time</td>
<td>Command from substation or Reply in time</td>
</tr>
<tr>
<td>Severe error status in reply</td>
<td>Command from substation or Reply in time</td>
</tr>
<tr>
<td>No poll response in time</td>
<td>Response to poll (EOT or message)</td>
</tr>
</tbody>
</table>

### FS Fast Select during Suspension

Determining which kind of commands from the application that 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
16.3.4. Diagnostic and counter attributes

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  
**Indexing:** Read: Word addresses of the counters to be read.  
Write: No index  
**Access:** Not preconfigurable, otherwise no restrictions

---

**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, each communication frontend 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 communication frontend transmits a reply message with a non-zero status code value (STS). Usually, the error code is caused by missing definitions in the communication frontend 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

---

**DE Diagnostics Enable**

Indicates whether the communication frontend 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

**DI**

**Diagnostic Interval**

The time between diagnostic commands to a station, which is not in suspended state (cp. the SU attribute, Chapter 13).

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

**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
LS  Last Error Status
The MicroSCADA 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 MicroSCADA error codes are translated to ANSI codes. See the Status Codes manual.

### 16.3.5. 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.

**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, not preconfigurable

**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, not preconfigurable

**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
16.3.6. 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 16.3.7), 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 Chapter 13.

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)

**AD Start Address**

The word address of the first word of the memory area.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Value</th>
<th>Indexing</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>The NET unit allows values from 0 to 32767, the size of the station memory may set a lower maximum value.</td>
<td>Memory area number. Only one index is allowed.</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>
**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

---

**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.
MicroSCADA Pro
System Objects
Technical Description

Indexing: Memory area number. Only one index is allowed.
Access: Read, conditional write

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

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
```

**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

**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
```

**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 communication frontend 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:

<table>
<thead>
<tr>
<th>DT</th>
<th>BI (Binary Input)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT 1</td>
<td>Binary Input</td>
<td></td>
</tr>
<tr>
<td>CO 3</td>
<td>16 bit binary number</td>
<td></td>
</tr>
<tr>
<td>AD 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LE 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AT 0</td>
<td>Unprotected</td>
<td></td>
</tr>
<tr>
<td>BF 1</td>
<td>Basic Allen-Bradley format</td>
<td></td>
</tr>
<tr>
<td>TS 0</td>
<td>No time stamp</td>
<td></td>
</tr>
<tr>
<td>SL 0</td>
<td>All five elements</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DT</th>
<th>BO (Binary Output)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT 2</td>
<td>Binary Output</td>
<td></td>
</tr>
<tr>
<td>CO 3</td>
<td>16 bit binary number</td>
<td></td>
</tr>
<tr>
<td>AD 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LE 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AT 1</td>
<td>Protected</td>
<td></td>
</tr>
<tr>
<td>BF 1</td>
<td>Basic Allen-Bradley format</td>
<td></td>
</tr>
<tr>
<td>TS 0</td>
<td>No time stamp</td>
<td></td>
</tr>
<tr>
<td>SL 0</td>
<td>All five elements</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DT</th>
<th>AI (Analog Input)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT 3</td>
<td>Analog Input</td>
<td></td>
</tr>
<tr>
<td>CO 5</td>
<td>3 digit BCD number</td>
<td></td>
</tr>
<tr>
<td>AD 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LE 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AT 0</td>
<td>Unprotected</td>
<td></td>
</tr>
<tr>
<td>BF 1</td>
<td>Basic Allen-Bradley format</td>
<td></td>
</tr>
<tr>
<td>TS 0</td>
<td>No time stamp</td>
<td></td>
</tr>
<tr>
<td>SL 0</td>
<td>All five elements</td>
<td></td>
</tr>
</tbody>
</table>
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:

```c
#SET STA1:SMR5 = 417
```

**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.

### 16.3.7. Message split

The spontaneous messages from the station are sent to the application specified by the AS attribute (see Section 16.2). 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 Fig. 16.3.7.-1. The feature must be activated for each STA individually. The receiving applications are memory area specific.
Message Split

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 with five integer elements, which can have the following values:
- 0 Element not used
- 2049 APL1
- 2050 APL2
- 2064 APL16

Any of the values above can be given to any element in the list. In the preconfiguration, the attribute is found among the "Memory Rung" 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 in the preconfiguration.

Indexing: The indices used in connection with the SL attribute are obtained from the formula:
\[ \text{Index} = 100 \times (\text{memory area no}) + (\text{SL element number}) \]
where
- 'memory area no' = see the MR attribute in Chapter 13
- '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

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 repies
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

16.3.8. Memory access

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 16.3.6)
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: Not preconfigurable, otherwise, 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

16.3.9. Time synchronization

SY Clock Synchronization

Synchronising 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 synchronising a station requires that a memory area with DT = 6 and LE = 9 has been defined, see Section 16.3.6. 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

16.4. STA attributes, S.P.I.D.E.R. and collector RTUs

Besides the common attributes described in Section 16.2, stations of type RTU (S.P.I.D.E.R. RTUs and Collector RTUs) have the attributes described in this section.

16.4.1. Basic attributes

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 16.4.1.-1.

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
Fig. 16.4.1.-1  An illustration of the HR attribute

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

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
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, a too big value, compared to the actual transmission delay, will cause the RTU time to be ahead of the actual time.

16.4.2. Diagnostic counters

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 Chapter 13
Indexing: Counter number
Access: Read-only, the values can be reset

16.4.3. 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, the communication frontend responds with an error code.
**MicroSCADA Pro**  
**System Objects**  
**Technical Description**

**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:**

<table>
<thead>
<tr>
<th>No</th>
<th>Meaning</th>
<th>Additional info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cold start</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Activate RTU after FTAB's</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>End of period/intermediate reading of PC's</td>
<td>1 = Intermediate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = End of period</td>
</tr>
<tr>
<td>4</td>
<td>Generate event with PROM version</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>Activate/deactivate local printer</td>
<td>1 = Activate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = Deactivate</td>
</tr>
<tr>
<td>9</td>
<td>Send system message queue</td>
<td>-</td>
</tr>
<tr>
<td>18</td>
<td>Deactivate RTU, accept complete set of FTAB's</td>
<td>-</td>
</tr>
<tr>
<td>19</td>
<td>Database time tag (configuration time)</td>
<td>Time, generated by the SCIL function RTU_ATIME</td>
</tr>
</tbody>
</table>

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`

**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
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 in the communication frontend.

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.

Data type: Any
Value: Any or no value
Access: Write-only

Example:

STA2 is synchronized:

`#SET STA2:SSY`

⚠️ The communication frontend itself should be synchronized before synchronising the RTUs.

16.4.4. Process communication

DA Process Data

Object and regulation commands, digital and analog setpoints and general outputs.

The DA attribute is automatically set by the base system when the output values in the process database are set. It is therefore not recommended to set the DA attribute directly from SCIL.

When receiving a write command of the DA attribute the communication frontend will generate the corresponding RP570 telegram and send it down to the RTU. Unless the command is a regulation command, the communication frontend will wait for a response from the RTU. If there is no response within a certain time (configurable, see the RT attribute), the communication frontend will respond by an error code to the base system. Otherwise, the communication frontend will check the response telegram and respond with an error code to the base system if it differs from what is expected as a positive acknowledgement to the command.

When this attribute is set, the communication frontend 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, the communication frontend responds with an error code.

Data type: Integer
Value: Depends on the object type
MicroSCADA Pro

System Objects

Technical Description

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

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
Rec. 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

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

Index: One index that expresses the RTU object number and type in a packed format (see the SCIL function RTU_OA)

Access: Write-only

!! Reading the DA attribute is not supported. Writing the attribute with SCIL is not recommended.
Example:

```bash
#SET STA2:SSC = 1
```

### SM  Sync Mode

Selecting the time synchronization mode, monologue or a dialog.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Default value:</td>
<td>0</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

### 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 MicroSCADA 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.

**Recommended values:** Recommended value for TA(3) = 512  
**Default values:** 0 for indices 1 ... 3  
TA(4) = 2304  
TA(5) = 1792  
**Access:** No restrictions

**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:**

```plaintext
#SET STA1:STD = "ABCDEF"  
#ON RTU1:E1 #IF RTU1:P1 = 12683 #THEN -  
@RESPONSE = STA1:SRD
```
16.4.5. Terminal reports

**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 RTUC_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

**TM**  Terminal Message

Reading of the terminal message (system message in the RTU) stored in the communication frontend. There is a ring buffer storage of 3 messages for each RTU in the communication frontend. Each time the communication frontend receives a terminal message it will send a tag number (1 ... 999) as a system message (see the MI attribute). The corresponding message can be fetched from the communication frontend 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), the communication frontend responds with the error code RTUC_MESSAGE_NOT_FOUND.

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

**TS**  Terminal Status

Reading the current (= last reported) terminal status stored in the communication frontend 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.
16.4.6. Communication loop attributes

The attributes in this subsection are not valid for stations of type Collector, only S.P.I.D.E.R. RTUs. Communication loops are configured by a special application package, and the application engineer does not normally need to use these attributes. See the System Configuration manual.

DR Direction

The polling direction of the RTU. The direction is specified by the loop line configuration (the NET line attributes DR and CF).

Data type: Text
Value: "A" or "B"
Access: Read-only

LS RTU Loop Status

The loop status of the RTU.

Data type: Integer
Value: 0 ... 8:

0 OK. Polling works as normally
1 CONNECT PENDING. The polling direction has been changed, but the RTU has still not responded to polls from the new direction.
2 REMOVE PENDING. The polling direction has been changed, but the RTU has still not been removed from the old polling direction.
3 CLOSE_PENDING. The RTU is not any more the last one of its direction, and so its loop reversal unit must be closed
4 OPEN_PENDING. The RTU is the last in the polling chain of a direction, and its loop control unit must open the loop.
5 PASSIVATED. The RTU is passivated. It does not receive communication commands. This value means that LS has been read from the "Wrong" NET unit or that the loop mode has not been initialized in NET unit.
6 NOT_RESPONDING. The RTU has not responded to the last poll
7 CONNECT FAILED. The RTU did not respond during the whole scan time
LU  Loop Mode In Use
The state of the loop mode - if it is in use on the RTU line. The loop mode is activated and deactivated with the line attribute LU (Chapter 13).
Data type:  Integer
Value:     0  Not in use
           1  In use
Access:    Read-only

LW  Loop Switch
The state of the Loop Reversal Unit of the RTU. The state can be changed with the line attribute CF and automatically.
Data type:  Integer
Value:     0  Open
           1  Closed
Access:    Read-only

16.5. STA attributes, SPACOM
Besides the common attributes described in Section 16.2, the STA object of type SPA (SPACOM), have the attributes described in this section.

16.5.1. Basic attributes

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 STA0.
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
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.

**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 ... 255 if set in the preconfiguration, 1 ... 899 on-line.
- **Access:** No restrictions

**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

**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

**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

### 16.5.2. Event handling attributes

#### 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

#### 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

#### EP  
**Event Poll Priority Class**

The event poll priority class of the station. Using the SPA line attribute PT (see Chapter 13) is it 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
16.5.3. Diagnostic attributes

**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 DISCREPANCY
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

16.5.4. Station suspension attribute

**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
SPG point definitions

These attributes specify the handling of individual SPA points in NET unit. Station specific sequence numbers identifies the SPA points.

**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 16.5.5-1.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Object type</td>
</tr>
<tr>
<td>2</td>
<td>Channel 1</td>
</tr>
<tr>
<td>3</td>
<td>Channel 2</td>
</tr>
<tr>
<td>4</td>
<td>Bits per channel</td>
</tr>
<tr>
<td>5</td>
<td>Significant bits</td>
</tr>
<tr>
<td>6</td>
<td>Event number</td>
</tr>
<tr>
<td>7</td>
<td>Event value</td>
</tr>
<tr>
<td>8</td>
<td>Event number for odd elements and event value for even elements (optional)</td>
</tr>
</tbody>
</table>

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:

```plaintext
#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:

```plaintext
#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 (1/16I) as event updated:

```plaintext
#SET STA1:SSP1 = (0,1,16,"I",1,1,4,0,0,2)
#SET STA1:SED1 = (0,1,16,1,1,1,1,2,0)
```
### Table 16.5.5-1 Explanations of the SPA point and event updating definition parameters (the SP and ED attributes)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit transpose mask</td>
<td>Integer, 0 ... 65535. The bits in the bit mask of the integer specify in pairs a possible change of bit order in double indications. &quot;00&quot; = no change of order. &quot;11&quot; = change of order. In MicroSCADA the first bit in a double indication is supposed to be &quot;closed&quot; and the second bit &quot;open&quot;. If the SPACOM unit uses another order, the bits must change order.</td>
</tr>
<tr>
<td>bit type mask</td>
<td>Integer, 0 ... 65535. The bits in the bit mask of the integer specify in pairs the type of indication: &quot;11&quot; = double indication, &quot;0&quot; = single indication.</td>
</tr>
<tr>
<td>channel 1</td>
<td>Integer, 0 ... 999. The lowest channel which updates the point.</td>
</tr>
<tr>
<td>channel 2</td>
<td>Integer, 0 ... 999. The highest channel which updates the point.</td>
</tr>
</tbody>
</table>
| 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, 1 ... 999                                                                                                                                 |
| data nr 2               | Integer, 1 ... 999                                                                                                                                 |
| 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        | An integer, whose bit mask specifies the bits that are affected by the event to data conversion. 1 - single indication, 3 = 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.     |
SPYSPA 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.

When writing to this attribute, all parameters must be present. See the parameter explanations in Table 16.5.5-1. 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 16.5.5-1.

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: MicroSCADA 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: MicroSCADA process object address
9: Filter (deadband)
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_UNSIGNED_8
   4 = LBT_SIGNED_8
   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: MicroSCADA 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 MicroSCADA process object address is 200:

#SET STA1:SSP1 = (2,0,0,"I",3,3,3,200,0.1,1)
**16.5.6. Miscellaneous SPACOM STA attributes**

**DA Data**

This attribute is used for process database communication. It may not be used in SCIL programs.

**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

**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.
MicroSCADA Pro
System Objects
Technical Description

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"

**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

**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
16.6. STA Objects, P214 RTUs

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 in the preconfiguration (up to 20 stations) or on-line with the PC attribute, see section 11.2.4.

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. When the objects are defined in the preconfiguration, only the attributes which can not be preconfigured get the default values, the other attributes get the values given in the preconfiguration.

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.

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:

\[\text{STAn:Sat}\]

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.

Attributes
The STA attributes in this section are valid only for stations of type P214. The attributes are described in the following subsections:

14.6.1 Basic Attributes: IU, LI, SA
14.6.2 Device Reservation: AL, AS
14.6.3. Suspension and Diagnostics: DC, RT
14.6.4. System Message Handling: MI, MS, OS
14.6.5. Data Communication: DA, EC, FC, FE, GP, NR, TV
14.6.6. Priority Control: PC, PM

16.6.1. Basic Attributes

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

**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:**
Integer 1 .. 12

**Default value:**
The NETn:SPC attribute

**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

16.6.2. **Device Reservation**

**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).

**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 ... 16. The application number as known to the communication unit. 0 = no application.

**Access:**
No restrictions
### 16.6.3. 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 11.3.5.).
- When a reply message from a station does not arrive in time (REPLY TIMEOUT), see the RT attribute.

#### 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:** Not preconfigurable, otherwise no restrictions.

#### RLT  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.

### 16.6.4. 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 MicroSCADA main program automatically updates the validity stamp of the object values in the process database (the OS attribute), see figure 11-4 and Section 11.4.6.

Refer to Section 11.2.6. to learn more about the system message handling.
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 11.2.6.
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 11.2.6

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 ... 16. The APL object number as known to the
communication unit.
Default value: 1
Access: No restrictions

OS    Objects 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

16.6.5.     Data Communication Attributes

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.

**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.
```
MicroSCADA Pro
System Objects
Technical Description

**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
```

**FE**  
*Flush Events*

Setting this attributes clears all the event buffers of the station.

Value: 1  
Access: Write-only

**Example:**

```
#SET STA1:SFE = 1
```

**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
```

**NR**  
*Normalize*

After start-up the RTU is normalized by writing to this attribute.

Value: 1  
Access: Write-only

**Example:**

```
#SET STA{n}:SNR = 1
```
ST  Set Time
Synchronizes the clocks of all P214 RTUs to the NET clock using one broadcast command to each P214 line.

Value: 1
Access: Write-only, only for broadcast station

Example:

```
#SET STA\n:SST = 1
```

The front-end clock should be synchronized first (see the NET\n:TM attribute).

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 STA\n:STV=5
```

Requests current measurand values.

16.6.6. Priority Control Attributes

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

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

16.7. STA attributes, REX stations

Besides the common attributes described in Section 16.2, the STA object of type REX (REF, RED, REC, etc. relays) have the attributes described in this section.

16.7.1. Basic attributes

**NN**  
**Node Number**
The LONWORKS node number of the station.
Data type: Integer
Value: 1 ... 127
Access: No restrictions

**SN**  
**Subnet Number**
The subnet number of the station.
Data type: Integer
Value: 1 ... 127
Access: No restrictions

**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

**UT**  
**Unit Type**
Not used in the present PC-NET version (8.4.3).
Data type: Integer
Value: 0 = REx device
       (1 = REC 561)
Index: None
Default value: 0
Access: No restrictions
16.7.2. Session handling

**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

**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, not preconfigurable

**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
```

**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)
MicroSCADA Pro  
System Objects  
Technical Description

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

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

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
16.7.3. Process communication

**DA**  Data

This attribute is used for process database communication. It is not used from SCIL programs.

**RQ**  Receive Quota

Receive quota for the station. Generally, the default value is suitable.

Data type: Integer
Values: 1 .. 10
Default value: 10
Indexing: None
Access: No restrictions

**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, MicroSCADA 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"
```

**TQ**  Transmit Quota

Transmit quota for this device. Generally, the default value is suitable.

Data type: Integer
Values: 1 .. 10
Indexing: None
Default value: 10
Access: No restrictions
MicroSCADA Pro  
System Objects  
Technical Description

**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
```

**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 |
| 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 startup 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 Debloc
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

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)
16.7.4. Event handling

**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 MicroSCADA.

- **Data type:** Integer
- **Value:** 0 ... 4
- **Default:** 0
- **Suggested value:** 2
- **Access:** No restrictions

**HI Historical Events**

Specifies whether history events are requested at event session startup 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

**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:

```plaintext
#SET STA1:SHS=(NEWEST_EV:PRT1, NEWEST_EV:PRM1)
```

**RM Running Mode**

Determination 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

**16.7.5. Suspension attributes**

**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

**16.7.6. SPA point definition**

The binary output objects of the REX stations must be defined in NET unit as SPA points using the SP attribute.

**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
System Objects

Technical Description

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)
```

16.7.7. File transfer handling attributes

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
```

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
16.8. STA attributes, LMK stations

Besides the common attributes described in Section 16.2, the STA object of type LMK (LSG devices and other LONWORKS devices, but not REX relays) has the attributes described in this section.

16.8.1. Basic definition attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Name</th>
<th>Description</th>
<th>Data type</th>
<th>Values</th>
<th>Indexing</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>Node Number</td>
<td>The LONWORKS node number of the station.</td>
<td>Integer</td>
<td>1 ... 127</td>
<td>None</td>
<td>No restrictions</td>
</tr>
<tr>
<td>SN</td>
<td>Subnet Number</td>
<td>The subnet number of the station.</td>
<td>Integer</td>
<td>1 ... 127</td>
<td>None</td>
<td>No restrictions</td>
</tr>
<tr>
<td>UT</td>
<td>Unit Type</td>
<td>The type of the LMK station:</td>
<td>Integer</td>
<td></td>
<td></td>
<td>No restrictions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LSG device.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Multiple LONMARK devices (devices which take input from many physical devices).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Other devices using the standard LONWORKS interface (for example Weidmüller).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value:</td>
<td>1</td>
<td>LONMARK device (not multiple and not LSG device)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Multiple LONMARK device (this device takes input from many physical devices)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>LSG device</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default value:</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
16.8.2. Polling attribute

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).

- **Data type:** Integer
- **Value:** 0 .. 2550 = No consistency checking
- **Unit:** Minutes
- **Indexing:** None
- **Access:** No restrictions

16.8.3. Process communication

DA  Data
The attribute for process communication database. It is not used from SCIL programs.

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
```

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
### RT  Reply Timeout

Maximum time in milliseconds to wait for reply from a LONWORKS node when sending commands and transparent SPA messages.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0 ... 65535</td>
</tr>
<tr>
<td>Unit</td>
<td>Milliseconds</td>
</tr>
<tr>
<td>Indexing</td>
<td>None</td>
</tr>
<tr>
<td>Default value</td>
<td>5000 milliseconds</td>
</tr>
<tr>
<td>Access</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

### Diagnostic attributes

#### 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

<table>
<thead>
<tr>
<th>Data type</th>
<th>Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Vector of 10 integers in the range 0 ... 30000</td>
</tr>
<tr>
<td>Index</td>
<td>1 ... 8, counter number</td>
</tr>
<tr>
<td>Access</td>
<td>Read-only, the values can be reset</td>
</tr>
</tbody>
</table>

#### 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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>0 ... 65535</td>
</tr>
<tr>
<td>Unit</td>
<td>Seconds</td>
</tr>
<tr>
<td>Indexing</td>
<td>None</td>
</tr>
<tr>
<td>Access</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>
16.8.5. LON point definition

**LP LON Point**

Ties together the LONWORKS Network Variable indices with process objects in the MicroSCADA 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 16.8.5-1.
- **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.
- **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 ... 65535

_**Digital output definition:**_

- **Element 1:** 4
- **Element 2:** Network variable index. Integer, 0 ... 4095.
- **Element 3:** LON base type, see Table 16.8.5-1.
- **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
Analog output definition:
Element 1: 5
Element 2: Network variable index. Integer, 0 ... 4095.
Element 3: LON base type, see Table 16.8.5-1.
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 16.8.5-1.
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.

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 1 current on bay 3”,1, 6666, 1.123)*
MicroSCADA Pro

System Objects

Technical Description

Table 16.8.5-1 The LON base types

<table>
<thead>
<tr>
<th>Type Code</th>
<th>Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LBT_UNSIGNED_16</td>
</tr>
<tr>
<td>2</td>
<td>LBT_SIGNED_16</td>
</tr>
<tr>
<td>3</td>
<td>LBT_UNSIGNED_8</td>
</tr>
<tr>
<td>4</td>
<td>LBT_SIGNED_8</td>
</tr>
<tr>
<td>5</td>
<td>LBT_SIGNED_32</td>
</tr>
<tr>
<td>7</td>
<td>LBT_FLOAT_IEEE754</td>
</tr>
<tr>
<td>8</td>
<td>LBT_STRUCTURE</td>
</tr>
<tr>
<td>9</td>
<td>LBT_LSG_CONTROL</td>
</tr>
<tr>
<td>10</td>
<td>LBT_LSG_BIT_WRITE</td>
</tr>
</tbody>
</table>

16.9. 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 MicroSCADA base system.

The attribute interface of Modbus Slave station resembles the process communication attributes of SPI stations.

Besides the common attributes described in Section 16.2, the STA object of type SPI has the attributes described in this section.

16.9.1. Basic attributes

<table>
<thead>
<tr>
<th>SA</th>
<th>Station Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data type</td>
<td>Integer</td>
</tr>
<tr>
<td>Value</td>
<td>1 ... 255</td>
</tr>
<tr>
<td>Access</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>

16.9.2. Configuration attributes

<table>
<thead>
<tr>
<th>FT</th>
<th>Function Table (FTAB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading FTABs to NET unit. By using this attribute, NET unit can be parameterized with FTABs loaded from the MicroSCADA base system, and FTABs need not be downloaded from the master station.</td>
<td></td>
</tr>
</tbody>
</table>
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 document "Functional specification for RP-570 Slave protocol in NET" lists supported FTAB fields. 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("0A640000000000010000000A00"); WRITE FTAB(s)
#SET STA1:SST1=(8,1). ; ACTIVATE WHEN ALL FTABS ARE WRITTEN
```

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

16.9.3. Process data communication attributes
The following attributes are used for sending data from MicroSCADA 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 MicroSCADA are updated in the MicroSCADA process database as input data.

These attributes apply also to Modbus Slave station. For more information on Modbus Slave stations, see the manual Configuring MicroSCADA for Modbus Slave.
**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 MicroSCADA 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 behaviour can be modified with the configuration attribute OL. If the value is outside -2047 ...+2047, it is also 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:

<table>
<thead>
<tr>
<th>Bit nr in status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B3 ... B0</td>
<td></td>
</tr>
<tr>
<td>0 0 0</td>
<td>Normal</td>
</tr>
<tr>
<td>0 0 1</td>
<td>Low alarm zone</td>
</tr>
<tr>
<td>0 1 0</td>
<td>High alarm zone</td>
</tr>
<tr>
<td>0 1 1</td>
<td>Low warning zone</td>
</tr>
<tr>
<td>1 0 0</td>
<td>High warning zone</td>
</tr>
<tr>
<td>1 0 1</td>
<td>Spare</td>
</tr>
<tr>
<td>1 1 0</td>
<td>Spare</td>
</tr>
<tr>
<td>1 1 1</td>
<td>Faulty value</td>
</tr>
</tbody>
</table>
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):

```scil
#SET STA99:SAV(100) = (RTU_ATIME(%RT,%RM),%OV,0)
```

### 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:

```scil
(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
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.

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:

```scil
#SET STA1:SEI55 = (%TIME, 8, 0, 5)
```

To send an ERMI message when bit 8 changes state to 1, execute the following SCIL statement:

```scil
#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:

```scil
#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:

```scil
#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:

```scil
#SET STA1:SEI55 = ( TIME, 2, 3, 5)
```

**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 MicroSCADA 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:

<table>
<thead>
<tr>
<th>Vector element</th>
<th>ERMI</th>
<th>ERMI'R</th>
<th>ERMA</th>
<th>ERMD</th>
<th>ERMFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32</td>
<td>41</td>
<td>33</td>
<td>34</td>
<td>43</td>
</tr>
<tr>
<td>2...7</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>8</td>
<td>Time quality</td>
<td>Time quality</td>
<td>Time quality</td>
<td>Time quality</td>
<td>Time quality</td>
</tr>
<tr>
<td>9</td>
<td>Bit number</td>
<td>Bit number</td>
<td>Limit info</td>
<td>Limit info</td>
<td>Format</td>
</tr>
<tr>
<td>10</td>
<td>Bit value</td>
<td>Bit value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>11</td>
<td>Rel.time msb</td>
<td>lsb</td>
<td>Value</td>
<td>Value</td>
<td>lsb</td>
</tr>
<tr>
<td>12</td>
<td>lsb</td>
<td>Number msb</td>
<td>Value</td>
<td>lsb</td>
<td>Ret.time msb</td>
</tr>
<tr>
<td>13</td>
<td>lsb</td>
<td>Cause of transmission</td>
<td>Value</td>
<td>lsb</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>lsb</td>
<td>Cause of transmission</td>
<td>Value</td>
<td>lsb</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>lsb</td>
<td>Cause of transmission</td>
<td>Value</td>
<td>lsb</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Number msb</td>
<td>lsb</td>
<td>Cause of transmission</td>
<td>Value</td>
<td>lsb</td>
</tr>
<tr>
<td>17</td>
<td>lsb</td>
<td>Cause of transmission</td>
<td>Value</td>
<td>lsb</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>lsb</td>
<td>Cause of transmission</td>
<td>Value</td>
<td>lsb</td>
<td></td>
</tr>
</tbody>
</table>

- 'time' - Vector of 6 text elements
- 'time quality' - Bit0 As T_QUALITY part in ERMx messages
  - ... Bit2
  - 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 As STATUS part in ERMx messages
  - ... Bit2
  - Bit3 LLD (level limit direction)
  - Bit4 LLP (level limit passed flag)

Indexing: Block number
Access: Write-only
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 128 to ‘bit_nr’, when ‘bit_nr’ is regarded as the number of the lower bit (0,2,4,...14). This feature is included for compatibility with older (ADLP-180) slave protocols, but should not be used it in new applications.
‘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.
MicroSCADA Pro
System Objects
Technical Description

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)
```

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)
MicroSCADA Pro
System Objects
Technical Description

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:

<table>
<thead>
<tr>
<th>Bin number</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>Spare (not used)</td>
</tr>
<tr>
<td>B1</td>
<td>EPR, 1 = end of period reading</td>
</tr>
<tr>
<td>B2</td>
<td>IR, 1 = intermediate reading</td>
</tr>
<tr>
<td>B3</td>
<td>LS, 1 = Local storage implemented</td>
</tr>
<tr>
<td>B4</td>
<td>IT, 1 = Invalid Time</td>
</tr>
<tr>
<td>B5</td>
<td>CT, 1 = Changed Time</td>
</tr>
<tr>
<td>B6</td>
<td>IV, 1 = Invalid Value</td>
</tr>
<tr>
<td>B7</td>
<td>RC, 1 = Restarted Counter</td>
</tr>
</tbody>
</table>

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:

```plaintext
@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
```

**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
MicroSCADA Pro  
System Objects  
Technical Description

**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

**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 most 230 byte strings  
Indexing: None  
Access: Write-only

**16.9.4. Function Control Attributes**

**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
**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

**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

**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
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

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

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 MicroSCADA 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: Integer
Unit: Seconds
Indexing: None
Access: No restrictions
MicroSCADA Pro
System Objects
Technical Description

TI Time Initialized
When all relevant parts of the MicroSCADA 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

16.9.5. Terminal messages

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
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):

<table>
<thead>
<tr>
<th>Bit number</th>
<th>Meaning if the bit is set (=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>* spare *</td>
</tr>
<tr>
<td>1</td>
<td>Plain texts are loaded</td>
</tr>
<tr>
<td>2</td>
<td>Backup medium for plain texts faulty</td>
</tr>
<tr>
<td>2-15</td>
<td>* spare *</td>
</tr>
</tbody>
</table>

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 16.9.4) is equivalent to setting ST(1) bit 9.

**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
6: INFO6

Indexing: Terminal event number. Only TEV 13 is allowed at the moment.

Access: Write-only
16.9.6. Loop control attributes

**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

**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 MicroSCADA 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 MicroSCADA base system.

- **Data type:** Integer
- **Value:** Integer
- **Unit:** Seconds
- **Indexing:** None
- **Default value:** 20 seconds
- **Access:** No restrictions

16.9.7. Redundant line attributes

The following attributes can be used for defining redundant lines for RP-570 Slave and IEC 60870-5-101 Slave protocols.

**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 16.2.2..

**Example:**

```
#SET STA1:SLI(2) = 5
```
**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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Value</th>
<th>Index range</th>
<th>Default value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>0...255</td>
<td>1 ... 8 (NET line numbering)</td>
<td>0</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>

16.10. **STA attributes, IEC station**

The attributes that are common to all protocols are found in Section 16.2. (IU, LI, AL, AS, MI, MS, OS, SE).

16.10.1. **The application layer**

The purpose of the application layer of the NET is a protocol conversion between IEC 60870 and ACP (MicroSCADA internal protocol) messages. The MicroSCADA sees the application layer of IEC 60870 protocol as STA object whose station type is IEC.

**The slave mode**

In the MicroSCADA implementation of the IEC 60870 slave device, the link address and common address of asdu can be different. This means that it is possible to allocate several STA objects on one line. In practice, this can be difficult, because the IEC 60870-5-101 protocol does not have any standard mechanism to inform the status of the particular station of a link (ex. is the station in use or not).

The IEC 60870-5-101 protocol converter does not have the internal database to store user data. All user data sent by MicroSCADA is transmitted to the controlling station as spontaneous data through the event queues of the NET unit. The slave device receives "data in controlling direction" from the master and sends "data in monitoring direction" to the master.

**The master mode**

When MicroSCADA acts as an unbalanced master mode, there can be several external devices (controlled stations or protection equipment's) on the same line. Each device on the line must have a unique STA object in MicroSCADA (and in NET unit). Devices on the line are addressed using SA and PA attributes (both are station attributes). The SA attribute corresponds to the "common address of asdu" and PA to the "link address" of a message. Two STA objects on the line cannot have the same SA, but they can have the same link address (PA). With balanced mode of protocol, only one link address (PA) on the same line is allowed.

The master device sends "data in controlling direction" to the slave and receives the "data in monitoring direction". The unbalanced master reads the data from slave by polling.
16.10.2. Data in monitoring direction with IEC 60870-5-101

The slave mode
The IEC station has two event queues for data in monitoring direction. These queues are called as class 1 and class 2. According to IEC 60870-5 the class 1 is intended to use for spontaneous data and class 2 is for cyclic data. However the MicroSCADA implementation does not take care how these data classes are used in application program. In the unbalanced mode of protocols, the controlling station reads data of queues by polling. When the balanced mode of protocol is used, the data is sent directly to controlling station and event queues are used only as temporary stores.

The master mode
The master receives data in monitoring direction from slave. This data is send directly to the process objects of base system.

16.10.3. Data in monitoring direction with IEC 60870-5-103

The slave mode
Not implemented.

The master mode
The NET unit operates with the data in monitoring direction mainly the same way in both protocols. With the -103 protocol the balanced mode is not specified. The NET unit sends the normal process data directly to the corresponding process objects of base system but the disturbance data and the most of generic data as bitstream.

16.10.4. Data in controlling direction

IEC 60870-5-101 protocol
When master sends a command to a slave, it begins to wait confirmation from slave. With -101 protocol the confirmation may have two steps, "activation confirmation" and "activation termination". When master receives the "activation confirmation" message from slave, it quits the SCIL command of base system and also begins to wait also "activation termination" message if "C_SE ACTTERM" is used. When termination message is received, the info of it is send to the termination process object. This PO is an analog input process object whose OA is the same as OA of command object incremented by sub-address 1. PO can have following values:

0 Positive confirmation -> "OK"
1 Negative confirmation
2 Activation timeout
3 Termination timeout
4 Negative termination

With -101 protocol, the confirmation message has actually same format than original command, only the COT code is changed.
IEC 60870-5-103 protocol

Although the -103 does not have two step confirmations as -101 protocol, the NET unit handles -103 commands with same method as -101. With -103 the link layer ACK of slave station to a command is handled like "activation confirmation" and the actual confirmation message as "activation termination". With -103 protocol the confirmation message is an ASDU 1 message which COT code informs the positive or negative acknowledge of command. The confirmation message for general interrogation is ASDU 8.

16.10.5. Data in controlling direction with IEC 60870-5-101

The master mode

When the MicroSCADA acts as protocol master, there is two alternating ways to send commands to the slave. The first way is to use direct process data interface: When output type process object is updated, the MicroSCADA generates automatically DA type ACP message to the NET unit.

Another possibility is use the pseudo transparent CO attribute, by writing a user configurable vector of data on it.

The most important application commands has own attributes: SY for time synchronization and GI for general interrogation.

The slave mode

The IEC 60870-5-101 slave of NET unit divides the IEC 60870-5-101 data in controlling direction to three classes as following:

Command class: ASDU number
Application commands: 100, 101, 102, 103, 104, 105, 106
Data commands: 45 ... 51
System commands: All others (110 ... 130)

Data commands in slave

The slave device translates received data commands to the ACP messages and sends them to the base system to update the corresponding process object. The object address in the ACP message is directly the information object address of ASDU. The MicroSCADA accepts the incoming data if address and type of data fits the corresponding process object. The 101 slave device handles ASDU's 45 ... 51 as data commands. With one-step confirmation, the NET does the confirmation automatically, but with two step confirmation, manual confirmation has to be used. Automatic confirmation means that NET unit creates internally confirmation message with COT=7 ("activation confirmation") when base system accepts data command. Manual confirmation means that user must create confirmation message using CF attribute.
Application commands in slave

The slave device sends the received application commands to the process object of base system called application command object. This process object is REX analog value object and its OA is 0. The OV of process object shows the ASDU number of command. The SCIL programming is needed to convert command to actions.

Those commands, which have general meaning, for example General Interrogation, belong to application commands.

System commands in slave

The slave device sends received system commands to the bitstream type object whose object address is the same as the value of the CA attribute. The NET does not translate to the system commands, so the user of SCIL must solve these. For example commands for parameter handling are system commands.

16.10.6. Data in controlling direction with IEC 60870-5-103

The master mode

The master device has special attributes for most important application commands. The SY attribute generates time synchronization and GI general interrogation commands. Other commands have to create using CO attribute or using command type process object.

16.10.7. Basic attributes of the application layer

This is an addition to the attributes presented in Section 16.2.

SA Station Address

The station address of IEC device. The Station address corresponds to the common address of ASDU. Station address must be unique among all IEC devices on the NET unit.

Data type: Integer
Value: 0 ... 255, when SL=1
0 ... 65535, when SL=2
Access: Read, conditional write

16.10.8. Process communication attributes

CF Command Confirmation

Slave:

The manual confirming of the received messages. When user writes the only the COT value to the CF attribute, NET unit sends confirmation message with written COT to the master. In this case the confirmation message is formed from the latest received command.
The user may also define which command is confirmed. NET unit automatically stores incoming commands to an internal table (20 latest commands is stored). In case OBJ_ADDR (or OBJ_ADDR and ASDU_ID) is given in CF-writing, the confirmed command is searched (newest first) from the table. If no command with matching address (or OBJ_ADDR and ASDU_ID) is found, error ICCC_NO_ACTIVE_COMMAND is returned and confirmation message is not sent. If a matching command is found, the given COT is set and the command is sent as a confirmation.

- **Data type:** Vector
- **Value:** Vector
- **Indexing:** No
- **Access:** Write-only

Vector (COT, [OBJ_ADDR, [ASDU_ID], [TDT]])

- **COT:** Cause of transmission
- **OBJ_ADDR:** Information object address of confirmed message
- **ASDU_ID:** TYPE IDENT (ASDU type) of confirmed message
- **TDT:** SPA message which is about to be sent as response to SPA command (as text)

TDT can be used only when confirming Transparent-SPA command i.e. TYPE IDENT 133 'C_SB_NA_1'. Also Information object address and TYPE IDENT must be given in this case.

To create the correct confirmation for application or system commands the COT, OBJ_ADDR and ASDU_ID parameters are needed. But to confirm data commands the COT and OBJ_ADDR parameters are enough.

**Master:**

If device is running in -101 master mode this attribute is used as configuration attribute to show is the two-step confirmation used or not.

- **Data type:** Integer
- **Value:**
  - 0: One step confirmation, no activation termination
  - 1: Activation termination used
- **Default value:** 1
- **Access:** No restrictions

**CO Command Out**

Sending commands to the external device. When user writes the vector to the CO attribute, the NET unit creates the command message with use of vector parameters and sends it to the slave device. The command vector contains three parameters and vector of transparent data. This transparent data part contains the part of ADS which is called in IEC standard "set of information elements". The user is responsible that this data part is correct.
This attribute can be used only when device runs in MASTER mode.

Data type: Vector
Value: Vector = (ID, IOA, COS, TDT)
- The ID number or name of the ASDU
  Name ex. ID = "C_SC_NA_1"
  Number ex. ID = 45
- Access: Write-only

Both notations can be used with IEC 60870-5-101 protocol but only number notation is accepted with IEC 60870-5-103 protocol.

IOA: Information object address
- With IEC 60870-5-101 the value range depends of IL attribute
- With IEC 60870-5-103 the IOA is a word (IL=2), which has following structure:
  - Msb = INF information number
  - Lsb = FNC function type

COS: Cause of transmission
- The value range 0 ... 0Xff
  If CL = 1
  The value range 0 ... 0xFFFF
  If CL = 2

TDT: Transparent data array of byte

**EV**

**Event Data with Time Stamp**

By writing data to the EV attribute user can send event data, i.e. time-tagged messages from MicroSCADA to controlling station. The EV attribute updates also the static data object in the Net database. By reading the EV attribute the user can enquire the number of information objects (events) in the queue not yet sent to the controlling station. This attribute can be used only when device runs in SLAVE mode.

Data type: Vector or integer
Value: When writing: Vector (TYPE, ADDR, VAL, COT, QU, PRI, RT, RM)
- When reading: Integer 0 ... 100
Index range: 1 ... 2
- When writing: The number of the class the message is assigned to
- When reading: The number event ASDUs not yet sent to controlling station
Access: No restrictions
Description of the vector parameters:

**TYPE:** Type identification of the ASDU, either as a text string (e.g. “M_IT_NA_1”) or as an integer.

**ADDR:** Information object address

**Value range:**
- 0 ... 255, when IL attribute = 1
- 0 ... 65535, when IL attribute = 2
- 0 ... 16777215, when IL attribute = 3

**VAL:** The value of the object. The value range depends on the data object type.

**COT:** Cause of transmission of the ASDU. See the description of the SD attribute.

**QU:** Bit mask of the qualifier descriptor bits of the ASDU. See the description of the SD attribute.

**PRI:** Priority of the information object. See the description of the SD attribute.

**Value:**
- Integer 0 ... 3
- 0 Always at the end of queue
- 1, 2 The event is allocated on the queue according the priority value. Event with higher priority passes the lower priority events. If the queue contains more than one event with the same priority new event is allocated to last of them.
- 3 Always at the beginning of queue

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

**RM:** Registration milliseconds. The millisecond part of the time stamp of the unsolicited message.

**Value:** Integer 0 ... 999

**GI General Interrogation**

Setting this attribute sends a general/group interrogation command (ASDU 100) to the IEC station. This attribute can be used only when device runs in MASTER mode.

**Data type:** Vector

**Value:** Vector = (ENA, [QOI])

**Access:** Write-only

**ENA:** Deactivate/ active general interrogation value: 0, 1

**QOI:** Qualifier of interrogation, optional value: 20 ... 36

If QOI is not used the IEC 60870-5-101 protocol creates global general interrogation (QOI=20).
RS  
**Reset Event Queues**

Clearing the content of event queue(s). This attribute can be used only when device runs in SLAVE mode.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Access:</td>
<td>Write-only</td>
</tr>
</tbody>
</table>

SD  
**Spontaneous Data**

Enables MicroSCADA to send spontaneous or cyclic data to the controlling station. This data is sent without time stamp. By reading the SD attribute the user can enquire the number of information objects in the queue not yet sent to the controlling station. The maximum value of the SD attribute when reading depends on the packing of the information objects to ASDUs and is approximately 3000. This attribute can be used only when the device runs in SLAVE mode.

<table>
<thead>
<tr>
<th>Data type:</th>
<th>Vector or integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>When writing: Vector (TYPE, ADDR, VAL, COT, QU, PRI, OW)</td>
</tr>
<tr>
<td></td>
<td>When reading: Integer</td>
</tr>
<tr>
<td>Index range:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>When writing:</td>
<td>The number of the class the message is assigned to</td>
</tr>
<tr>
<td>When reading:</td>
<td>The number of the class queue the length of which is enquired</td>
</tr>
<tr>
<td>Access:</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

Description of the vector parameters:

**TYPE:** Type identification of the ASDU, either as a text string (e.g. “M_IT_NA_1”) or as an integer.

**ADDR:** Information object address

| Value range: | 0 ... 255, when IL attribute = 1 |
|              | 0 ... 65535, when IL attribute = 2 |
|              | 0 ... 16777215, when IL attribute = 3 |

**VAL:** The value of the object. The value range depends on the data object type.

**COT:** Cause of transmission of the ASDU. This parameter describes the reason why a message is sent. If originator address is used, it can be set by adding 256*originator address to the COT value.
SY Synchronize

Making an accurate time synchronization of an IEC station. It is not necessary to write the time for this attribute, because the time is sent from the internal clock of the NET unit. The user is responsible for the first update of the internal clock of the NET unit. Setting of this attribute generates the ASDU 103 with the IEC 60870-5-101 master protocol and ASDU 6 with the IEC 60870-5-103 master protocol.

This attribute can be used only when the device runs in the MASTER mode.

Data type: Vector
Value: Vector = (COT, [BRO,[ADDR]])
Access: Write-only

COT: Cause of transmission of the synchronization messages. Value range: 0 ... 255 (same code as in IEC standard).
BRO: Broadcast, determines whether the synchronization message is a broadcast message (value 1) or not (value 0). If the value is 0 or omitted, the time synchronization is done only to the destination station. If the value is 1, the time synchronization is made as broadcast and the time of all stations on this line is updated. Value range: 0 or 1.
**TD**

**Transparent Data**

This attribute can be used only with IEC 60870-5-101 -protocol.

**Master:**

Sending commands to the external device. When the user writes vector to TD attribute, the NET creates a command message using vector parameters and sends it to the slave device. Command vector contains two parameters and transparent data. Transparent data is given as text-string.

**Data type:** Vector  
**Value:** Vector  
**Access:** Write-only

**Example:**

Transparent-SPA command:

```
#SET STA99:STD = ("C_SB_NA_1", 12345, 6, "+539281/2/3/4:1:"
```

**Slave:**

User can send transparent data information to the external device as an indication when NET is used as slave. When vector is written to TD-attribute by the user, the NET creates an indication message using vector parameters and puts it to the Class queue (unbalanced mode) or sends it spontaneously (balanced mode). Vector contains two parameters and transparent data. Transparent data is given as text-string.

**Data type:** Vector  
**Value:** Vector
MicroSCADA Pro

System Objects

Technical Description

Indexing: 1 ... 2 (Class 1...2)
Access: Write-only

Example:
Transparent-SPA Reply-message:

```
#SET STA99:STD(1) = ("M_SB_NA_1", 12345, 5, "<53D2E1/4")
```

Vector (TYPE IDENT, IOA, COT, TDT)

**TYPE IDENT**: The ID number or name of the ASDU
Value: 1 ... 255 when given as number

**IOA**: Information object address
Value: 1 ... 255 when IL-attribute equals one
       1 ... 65535 when IL-attribute equals two
       1 ... 16777215 when IL-attribute equals three

**COT**: Cause of transmission
Value: 1 ... 255 when CL-attribute equals one
       1 ... 65535 when CL-attribute equals two
       (typical value: 5 = requested)

**TDT**: Transparent data given as text-string

16.10.9.

Device configuration attributes

**CA**

Command Address
The object address of bitstream type process object in the MicroSCADA process object database, where the protocol converter sends the system commands from controlling station. This object also receives all commands that are not detected inside protocol converter.

Data type: Integer
Value: 0 ... 65534
Default value: 32000
Access: Read, conditional write

⚠️ The unit number (UN attribute) of the process object must be the same as the address (SA attribute) of the station.

**CL**

Length of Cause of Transmission Info
The length of COT field on the IEC 60870 message.

Data type: Integer
Value: 1 ... 2
Unit: Bytes
**CT**  
**Activation Termination Waiting Time**

The maximum time that the NET unit waits the activation termination reply from external (controlled) device. This attribute has meaning only when device runs in MASTER mode.

- **Data type:** Integer
- **Value:** 0 ... 255
- **Unit:** Seconds
- **Default:** 60
- **Access:** No restrictions

**IL**  
**Information Address Length**

The length of station address field on the message.

- **Data type:** Integer
- **Value:** 1 ... 3
- **Unit:** Octets
- **Default:** 2
- **Access:** Read, conditional write

**PA**  
**Polling Address, Address of Link**

The test address of the message (corresponds to the link address field in the message).

- **Data type:** Integer
- **Value:**
  - 0 ... 255, when PL attribute = 1
  - 0 ... 65535, when PL attribute = 2
- **Indexing:** Line number
- **Default:** 1
- **Access:** Read, conditional write

**PC**  
**Process Data Confirmation**

Controls how the confirmation message is sent to the controlling station. The manual confirmation means that user have to accept (or not accept) the incoming message by writing the confirmation result to the attribute CF. The automatic confirmation is done once the base system accepts the data message sent by protocol converter. This attribute has meaning only when device runs in SLAVE mode.

- **Data type:** Integer
- **Value:**
  - 0  Manual confirmation of process data in control direction
1 Automatic confirmation allowed of process data in control direction
Default value: 1 (automatic confirmation)
Access: Read, conditional write

**PL**  **Polling Address Length, Link Address Length**
The number of bytes in the link address field. If a link address is omitted, all messages are networked to the first (and only) process unit that is allocated on the line.

- **Data type:** Integer
- **Value:** 1, 2
- **Indexing:** Line number
- **Default:** 1
- **Access:** Read, conditional write

**RT**  **Application Reply Timeout**
The maximum time that the NET unit waits the activation confirmation reply from external (controlled) device. This attribute has meaning only when device runs in MASTER mode.

- **Data type:** Integer
- **Value:** 0 ... 255
- **Unit:** Seconds
- **Default:** 10
- **Access:** No restrictions

**SL**  **Station Address Length**
The length of common address field (station address) on the IEC 60870 message. The protocol driver fills the station address of device (SA attribute) to common address field of message using number of octets specified by SL attribute.

- **Data type:** Integer
- **Value:** 1, 2
- **Unit:** Bytes
- **Default:** 1. With IEC 60870-5–103 protocol this is always 1
- **Access:** Read, conditional write

**ST**  **Sys Waiting Time**
The maximum time that the NET unit waits for reply from the base system.

- **Data type:** Integer
- **Value:** 0 ... 60000 ms
- **Unit:** Milliseconds
TC Time Synchronization

Determines the behaviour of the slave device when it receives a time synchronization message. This attribute has meaning only when device runs in SLAVE mode.

Data type: Integer
Value: 0, 1, 2, 3:

0 Time synchronization is handled inside the NET unit. Message sets the NET internal clock (or directly SYS when NT-NET is used.).
1 Message sets the NET internal clock but the NET sends time setting message also to MicroSCADA.
2 Time setting message is sent directly to the MicroSCADA without setting NET’s time.
3 Not handled at all

Default: 1 (automatic confirmation)
Access: Read, conditional write

16.10.10. Other device handling attributes

DC Diagnostic Counters

The status of the diagnostic counters.

1. Suspended information (0 if OK, 1 if SUSPENDED)
2. Suspensions, counter
3. Sent data messages
4. Sent command messages
5. Sent confirmation messages
6. Received data messages
7. Received command messages
8. Received confirmation messages
9. Unknown messages

Data type: Vector of 10 integers
Value: 0 ... 65535
Default value: 0
Access: Read-only, the values can be reset
MicroSCADA Pro
System Objects
Technical Description

DR  Direction
When protocol is running in balanced mode, this attribute shows that MicroSCADA acts as station A or B in the protocol. This attribute is used only with balanced mode of protocol.
Data type: Integer
Value: 0, 1
Default value: 0 Secondary station (slave)
1 Primary station (master)
Indexing: Line number
Access: No restrictions

ML  Maximum Message Length
The maximum length of a transmitted message in octets.
Data type: Integer
Value: 20 ... 255
Indexing: Line number
Default: 230
Access: Read, conditional write

MT  Maximum Response Delayed Time
The maximum time to delay response of SD or EV writing when the number of items in the queue is greater than RW attribute. This attribute has meaning only when device runs in SLAVE mode.
Data type: Integer
Value: 0 ... 600
0 = delayed response mechanism not used
Unit: Seconds
Default: 15
Access: No restrictions

A large value of this attribute may cause queuing of command procedure executions during a communication disturbance.

RM  Running Mode
This attribute sets internal flags that control the function of the protocol converter.
Data type: Integer
Value: 1 ... 65535
Default value: 1
Access: Read, conditional write
Following controlling bits are currently used:

Bit 0: The hour transmission method of the events from the slave station
Bit 1: Time synchronization method
Bit 2: Confirmation queue (significant only in the unbalanced mode)
Bit 3: Handling of unrecognized commands
Bit 4: Sending general interrogation command at zero (OK) status
Bit 5: Sending general interrogation command at ASDU 70

**Bit 0 = 0**

Slave operation:

When the hour part of the time tagged events is changed, the slave sends the clock setting message (ASDU ID=103) before actual event message, to show new hour and date part of time tag. (The standard IEC 60870-5-101 time stamp message does not have hour part) This is the (default) operation.

Master operation:

The master device takes date for the indication timestamp sent by slave. If the slave has not sent its date, wrong time stamp in indication may result. (default)

**Bit 0 = 1**

Slave operation:

Do not send the clock setting message, events are sent using full time stamps if possible. However most ASDU types cannot be sent with full time stamps.

Master operation:

The master device takes date and hour for the indication timestamp from internal clock. The time difference detection (max. 1 minute) is used.

**Bit 1 = 0**

Slave operation:

The slave device waits the clock synchronization from master. The time invalid bit IV (in time stamps) is true until the first time synchronization. (default)

Master operation:

Not used in master

**Bit 1 = 1**

Slave operation:

The slave device does not expect the clock synchronization from master (ex. when radio clock is used). The time invalid bit IV is always false.

Master operation:

Not used in master
MicroSCADA Pro
System Objects
Technical Description

**Bit 2 = 0**
Slave operation:
The slave device (in unbalanced mode) puts the confirmation messages to CLASS1 queue. (default)
Master operation:
Not used in master

**Bit 2 = 1**
Slave operation:
The slave device (in unbalanced mode) puts the confirmation messages to CLASS2 queue
Master operation:
Not used in master

**Bit 3 = 0**
Slave operation:
Unknown commands are neglected (default)
Master operation:
Unknown indications are neglected (default)

**Bit 3 = 1**
Slave operation:
Unknown commands are directed to a bitstream object of which address is defined with CA-attribute
Master operation:
Unknown indications are directed to a bitstream object of which address is defined with CA-attribute

**Bit 4 = 0**
Slave operation:
Not used in slave
Master operation:
When this bit is 0 a general interrogation command is sent always when the object status of the IEC master station gets the value zero, e.g. when set in use or after a suspension.

**Bit 4 = 1**
Slave operation:
Not used in slave
Master operation:
When this bit is 1, general interrogation is not sent automatically at zero status.
Bit 5 = 0
Slave operation:
Not used in slave
Master operation:
When this bit is 0 a general interrogation command is sent always when the end of initialization message (ASDU 70) is received from the IEC slave

Bit 5 = 1
Slave operation:
Not used in slave
Master operation:
When this bit is 1, general interrogation is not sent automatically when receiving ASDU 70.

RW  Reply Window Size
Defines how many items (single-point information, measured values etc.) can be written from base system to the NET without reply from endpoint (external station). Normally the NET unit sends data items (written by SD or EV attribute) immediately to external station. But in case this is not possible, the NET stores data into local buffers and creates a local reply to base system (so the command procedure can continue). The NET stores items until the number of items in local buffers increases to be equal to RW. Then the NET delays the replies to base system until the number of items drops again below RW (NET gets reply from external device and sends next data message).

Data type: Integer
Value: 0 ... 500
Default value: 10 for IEC 60870
100 for DNP V3.00 slave protocol
Access: Read, conditional write

SR  Single Char Response
Enabling or disabling the single char responses. If single char response is enabled the NET unit replies with 0x65 character in case of ACK message (to command) or "requested data not available" message (to poll).

Data type: Integer
Value: 0, 1
Default value: 0  Single char responses disabled (slave)
1  Single char responses enabled (master)
Indexing: Line number
Access: Read, conditional write
16.10.11. **Redundant Line Attributes**

The following attributes can be used for defining redundant lines for RP-570 Slave and IEC 60870-5-101 Slave protocols.

**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 16.2.2..

**Example:**

```
#SET STA1:SLI(2) = 5
```

**RU Redundant Line Station**

This attribute defines the number of the STA object connected to redundant IEC 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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0...255</td>
</tr>
<tr>
<td>Index range</td>
<td>1 ... 8 (NET line numbering)</td>
</tr>
<tr>
<td>Default value</td>
<td>0</td>
</tr>
<tr>
<td>Access</td>
<td>Read, conditional write</td>
</tr>
</tbody>
</table>

16.11. **STA attributes, PLC stations**

PLC station is the heart of Modbus Protocol Converter of MicroSCADA. It coverts communication messages from ACP protocol to Modbus Master protocol and to the opposite direction. The PLC station stores the necessary information of a protocol and address conversion to topic data. The PLC station also stores the data that is scanned from external device to the internal database of a NET unit. The purpose of storing is to minimise the amount of messages between base system and NET unit, because the NET unit sends only changed data to the base system. The changed data is sent to the process database of a base system as RTU process data.

The scanning of external device data is controlled by topics of PLC station. Since the PLC station can contain several (max. 100) different topics, it is possible to divide the memory (and IO) of an external device into separate areas. These areas can contain different type of data and they can have their own scanning intervals.
Besides the common attributes described in section, the STA object of type PLC has the attributes described in this section. Note also that the PO attribute for the line that is used for communication has to have value 25.

16.11.1. Basic attributes

**SA**  
**Station Address**

The station address of the PLC station. The value of this attribute must be the same as the corresponding station address value defined in the station.

Each PLC station connected to a NET unit must have a unique station address. This demand for uniqueness also comprises the NET unit itself, see the SX attribute in Chapter 14. 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.

Legal addresses are 1 ... 255, except for the broadcast STA object, which has the address 0. The PLC cannot be taken in use (see the IU attribute) unless it has a legal address.

- **Data type:** Integer  
- **Value:** 0 ... 255  
- **Access:** Read, conditional write

16.11.2. Configuration attributes

**DC**  
**Diagnostic Counters**

The values of the diagnostic counters the NET unit keeps for the station.

To make the supervision and testing of the station communication easier, each communication frontend holds five diagnostic counters (numbered 1 ... 5) for each station. Each counter monitors a certain events, according to the following list:

- **RT**  
  **Reply Time-out**

The maximum time that the NET unit waits for a reply from the station.

- **Data type:** Integer  
- **Value:** 0 ... 655
MicroSCADA Pro
System Objects
Technical Description

Unit: Seconds
Default value: 15
Access: No restrictions

**TP Topic Configuration**

When this attribute is written, a vector configures the topic.

Data type: Vector
Value: Vector of 8 elements
Element 1 Allocation
2 FirstOA
3 LastOA
4 Type
5 BaseAddress
6 Format
7 Interval
8 Delta

Indexing: Topic number
Access: No restrictions

**Example:**

```
#SET STA1:SIU=0
#SET NET1:SIU4=0

Indication data type with OA=28673 .. 28673, format is io_bit, data type single indication and base address 1. The topic is polled every second:

#SET STA1:STP(1)=(1,1,7,000,1,1000,0)

The digital value block. OA value range is 36865 ... 36866, word_type, digital value, base address 400:

#SET STA1:STP(2)=(1,1,2,9,399,4,1000,000)

The analog value block. OA value range is 24576 ... 24577, int_type, analog value, base address 200:

#SET STA1:STP(3)=(1,1,2,6,199,4,10000,10)

The analog value block. The OA value range is 24578 ... 24578, long type, analog value, base address 400:

#SET STA1:STP(4)=(1,3,3,6,399,6,1000,00)

The object command block. The OA value range is 4097 ... 4126, io_bit, object command, base address 1:

#SET STA1:STP(5)=(1,1,30,1,0000,1,0,0)

The digital setpoint block. The OA value range is 12288 ... 12288, word type, digital setpoint, base address 254:

#SET STA1:STP(6)=(1,1,3,3,253,5,0,0)
```
The analog setpoint block. The OA value range is 16384 ... 16387, int type, analog setpoint, base address 314:

#SET STA1:STP(7)=(1,1,3,4,313,4,0,0)
#SET NET1:SIU4=1
#SET STA1:SIU=1

**Allocation**

This item specifies is the topic in use or not. The memory needed for the topic is reserved when the topic is taken into use.

**FirstObjectAddress**

This parameter specifies the first object address used with this topic. Object address and object type parameter together specify the actual process object address (OA) where the first item in the topic is stored:

\[
OA = 4096 \times OBJECT\_TYPE + OBJECT\_ADDRESS
\]

**LastObjectAddress**

The object address of the last item of the topic. The number of items reserved by the topic is calculated as following:

\[
NUMBER\_OF\_ITEMS = LASTOBJECTADDRESS - FIRSTOBJECTADDRESS
\]

**Type**

This parameter specifies the data type of process objects. Possible data types of a PLC station are listed in “Data types of a PLC station” on page 318.

*Table 16.11.2-1: Data types of a PLC station*

<table>
<thead>
<tr>
<th>Code</th>
<th>Type of process object</th>
<th>Possible data formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Object command</td>
<td>IO_BIT,M_BIT</td>
</tr>
<tr>
<td>3</td>
<td>Digital set-point</td>
<td>INT,WORD</td>
</tr>
<tr>
<td>4</td>
<td>Analog set-point</td>
<td>CHAR,INT,WORD,LONG,FLOAT</td>
</tr>
<tr>
<td>6</td>
<td>Analog value</td>
<td>CHAR,INT,WORD,LONG,FLOAT</td>
</tr>
<tr>
<td>7</td>
<td>Indication (single or double)</td>
<td>IO_BIT,M_BIT</td>
</tr>
<tr>
<td>8</td>
<td>Pulse counter</td>
<td>LONG</td>
</tr>
<tr>
<td>9</td>
<td>Digital value</td>
<td>INT,WORD</td>
</tr>
</tbody>
</table>

With indication type, one object address (OA) contains 16 bits.

**BaseAddress**

The address of the first item of a topic in the memory of the PLC station.
MicroSCADA Pro
System Objects
Technical Description

Format
Specifies how the data is stored in an external device. Table 16.11.2-2 lists possible formats. The Table 16.11.2-3 describes the relationship between Modbus message functions and topic parameters.

Table 16.11.2-2 Possible formats and their description

<table>
<thead>
<tr>
<th>Code</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IO_BIT</td>
<td>Bit in PLC’s input or output</td>
</tr>
<tr>
<td>2</td>
<td>M_BIT</td>
<td>Memory bit in PLC’s working memory</td>
</tr>
<tr>
<td>3</td>
<td>CHAR</td>
<td>Unsigned 8 type object in PLC’s registers. A register can allocate two CHAR.</td>
</tr>
<tr>
<td>4</td>
<td>INT</td>
<td>One register in Memory of the PLC station. The MSB bit is used as sign bit.</td>
</tr>
<tr>
<td>5</td>
<td>WORD</td>
<td>One register in Memory of the PLC station. The object is used in unsigned form.</td>
</tr>
<tr>
<td>6</td>
<td>LONG</td>
<td>Signed 32-bit object, which needs two registers from Memory of the PLC station in msw-lsw order</td>
</tr>
<tr>
<td>7</td>
<td>MSB_LONG</td>
<td>Signed 32-bit object, which needs two registers from Memory of the PLC station in lsw-msw order.</td>
</tr>
<tr>
<td>8</td>
<td>F32_TYPE</td>
<td>Floating point type, allocates two registers in PLC memory</td>
</tr>
<tr>
<td>10</td>
<td>IN_WORD</td>
<td>Input register of PLC (3X references)</td>
</tr>
</tbody>
</table>

Table 16.11.2-3 Modbus message functions and topic parameters

<table>
<thead>
<tr>
<th>Msg. Function</th>
<th>Topic type</th>
<th>Topic format</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>7</td>
<td>M_BIT</td>
</tr>
<tr>
<td>02</td>
<td>7</td>
<td>IO_BIT</td>
</tr>
<tr>
<td>03</td>
<td>6,8,9</td>
<td>CHAR,WORD,INT,LONG,FLOAT</td>
</tr>
<tr>
<td>04</td>
<td>6,9</td>
<td>IN_WORD</td>
</tr>
<tr>
<td>05</td>
<td>1</td>
<td>IO_BIT or M_BIT</td>
</tr>
<tr>
<td>06</td>
<td>3,4</td>
<td>CHAR,WORD,INT,LONG,FLOAT</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>M_BIT (only when writing to the DI attribute )</td>
</tr>
<tr>
<td>16</td>
<td>3,4</td>
<td>CHAR,WORD,LONG,FLOAT (when vector written)</td>
</tr>
</tbody>
</table>

Interval
The frequency the data of a topic is read from an external device. The unit of the value of the interval is milliseconds. If the interval is 0, the topic is not polled.

Delta
If the topic type is an analog value (type=4), then the delta value is used to minimise the amount of updating messages from the NET unit to the base system. The new analog value is sent to the base system when the change or sum (integral) of changes is bigger than delta.
16.11.3. Communication attributes

The base system communicates with PLC using communication attributes. The object of a communication attribute is the topic of the NET unit. When the base system writes to (or reads) the communication attribute, the station process changes the data of writing or reading to the plc message. The plc message is created using topic type and attribute parameters. The written data is sent forward from NET unit without storing it in internal database of the NET unit. The value that is get from PLC by reading communication attribute is not stored on NET database. The index for communication attribute must fit to a topic with right topic type (between IndexFirst and IndexLast).

**AV Analogue Value**

A reference to the register data in the memory of a PLC station. The register is always 16-bit word in the memory, but the PLC program can use registers in 1, 2 or 4 byte format. The PLC program can also use successive registers in different formats but for technical reasons this is not allowed for those register areas that are transferred to the base system. The whole register area that is referred by one topic must have equal format (type parameter) as topic.

Value: \( 0 \ldots 0xFFFFFFFF \) (hex)
Indexing: \( 0 \ldots 65535 \), Index = Object_Address
Access: No restrictions

**DI Double Indication**

A reference to the bit memory or bit IO of the PLC station. Double indication has (as single indication) two logical states (0 and 1), but in the double indication both directions have their own bit (0 -> 01, 1->10). The DI attribute can be used only to bit type topics.

The use of double indication needs support from PLC’s application program, because the program languages of PLCs do not support double indication data type.

Data type: Integer
Value: 1 or 2
Indexing: \( 0 \ldots 65535 \). The index for DI attribute is calculated as follows:
When reading: Index = Object_Address * 16 + bit number
When writing: Index = Object_Address (to DI attribute)
Access: No restrictions
Example:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LI</td>
<td>Line Number</td>
<td>00004</td>
</tr>
<tr>
<td>AL</td>
<td>Allocation</td>
<td>00001</td>
</tr>
<tr>
<td>AS</td>
<td>Allocating Applic.</td>
<td>00001</td>
</tr>
<tr>
<td>IU</td>
<td>In Use</td>
<td>00001</td>
</tr>
<tr>
<td>MI</td>
<td>Message Ident.</td>
<td>01001</td>
</tr>
<tr>
<td>MS</td>
<td>Message Application</td>
<td>00001</td>
</tr>
<tr>
<td>SA</td>
<td>Station Addr.(Dec)</td>
<td>00001</td>
</tr>
<tr>
<td>DE</td>
<td>Diagnostics Enable</td>
<td>00002</td>
</tr>
<tr>
<td>DI</td>
<td>Diagnostic Interval</td>
<td>00060</td>
</tr>
<tr>
<td>FS</td>
<td>Fast Sel. to Susp.</td>
<td>00000</td>
</tr>
<tr>
<td>RT</td>
<td>Reply Timeout</td>
<td>00060</td>
</tr>
<tr>
<td>SP</td>
<td>Message Split</td>
<td>00000</td>
</tr>
<tr>
<td>SU</td>
<td>Suspension Time</td>
<td>00060</td>
</tr>
</tbody>
</table>

**DV**  
**Digital Value**
A reference to the register data in the memory of a PLC station. The register is used as 16-bit word.

- **Data type:** Integer
- **Value:** 0 ... 65535
- **Indexing:** 0 ... 65535. Index = Object_Address
- **Access:** No restrictions

**GD**  
**General request of Data**
A request to send all the data of NET internal PLC database to base system. It does not cause communication between NET and remote PLC. When the data has been sent, the PLC station sends the system status message to the base system. The value of GD remains to 1 (when reading) until the updating is ready.

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

**SI**  
**Single Indication**
A reference to the single bit in Memory of the PLC station or IO. Writings to this attribute change the status of one bit in PLC. The SI attribute can be used only to bit type topics.

- **Data type:** Integer
- **Value:** 0 or 1
- **Indexing:** 0 ... 65535. The index for SI attribute is calculated as follows:
16.12. **STA attributes specific to DNP V3.00**

**AR**  
**Application Message Data Retries**  
The maximum number of retransmissions of an application data fragment.  
Data type: Integer  
Value: 0 ... 5  
Default value: 0  
Access: No restrictions

**AT**  
**Application Response Timeout**  
The time in seconds within which the application layer response from the slave must be completed. This is the maximum time from the beginning of the first fragment of the response to the end of the last fragment of the response.  
Data type: Integer  
Value: 0...600  
Unit: Seconds  
Default value: 30  
Access: No restrictions

**CA**  
**Command Address**  
The object address of bitstream process object in the MicroSCADA process database, where the slave device sends unidentified messages. If the value of the CA attribute is 0, the unidentified messages are not sent and the bitstream object is not updated.  
Data type: Integer  
Value: 0 ... 65534  
Default value: 0 (unidentified messages not sent)  
Access: No restrictions

**Example:**  
In the example of the communication system configuration, the slave addresses are 1 and 2 (slave station numbers).  
! The unit number of the bitstream process object must be the same as the slave address (SA attribute) of the slave station.
CF  Command Confirmation

The CF attribute is used to accept commands received by the NET unit. The NET unit does not send a response message to the command before the command is confirmed using CF attribute. The NET automatically stores the received commands to a table. When the parameter ADDR is given when writing data to the CF attribute, the confirmed command is searched (latest first) from the table. If no command with matching address is found, the error ICCC_NO_ACTIVE_COMMAND is returned and a response message is not sent.

Data type:  Vector (CS, OBJ_ADDR)
Value:  See above
Access:  Write-only

CO  Command Out

CO attribute can be used for generating command messages, i.e. requests, to DNP slave stations. All kinds of commands can be generated. The data content of the command is given as transparent data octet by octet. It must be noted that the user is responsible for the validity of the data content. For more information, please refer to the DNP standards listed in Chapter 3.

Data type:  Vector
Value:  (FNC, [TYPE, VAR, QUAL, [TDT]])
Access:  Write-only

CT  Confirmation Timeout

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

Data type:  Integer
Value:  0 .. 600
Unit:  Seconds
Default value:  10
Access:  No restrictions

DC  Diagnostic Counters

The values of the diagnostic counters which NET keeps for the station. The counters have the following meaning:
1. Suspension information (0 = OK, 1 = suspended)
2. Suspension counter
3. Transmitted data messages
4. Transmitted command messages
5. Transmitted confirmation messages
6. Received data messages
7. Received command messages
8. Received confirmation messages
9. Received unknown messages
10. TSDU in queue length
11. APDU out queue length
12. TSDU in queue length
13. TSDU out queue length
14. WAIT CONFIRM queue length
15. SYS transition queue length
16. Confirmation transition queue length
17. Select transition queue length
18. Free APDUs queue length
19. Free events queue length
20. Free SYS transitions queue length

Data type: Integer
Value: 1 ... 65535
Index range: 1 ... 20
Access: Read-only, the values can be reset

**DI Database Initialized**

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

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

**DP Data Point**

By using the DP attribute the user can configure the data polling of the DNP master station.

Data type: Vector
Value: When writing:

- Index 0: Vector (PI, NUM)
- Other indices: Vector (PI, TYPE, VAR, FIRST, LAST)

When reading:

- Index 0: Vector (PI, NUM, STATUS, TIME)
- Other indices: Vector (PI, TYPE, VAR, FIRST, LAST, STATUS, TIME)

Index range: 0...50:
- 0: Event poll
- 1...50: Freely defined polls

Access: No restrictions
**EO  Event Offset**

The address offset between process objects for static data and events with the same DNP address (index). The address of the event process object is the address of the process object for static data added with the value of this attribute. If both static data and events is wanted to be received in the same process object this attribute must be set to 0.

- **Data type:** Integer
- **Value:** 0..65535
- **Default value:** 0
- **Access:** Read, conditional write

**EV  Event Data with Time Stamp**

By writing data to the EV attribute user can send unsolicited messages (time-tagged events) from MicroSCADA to the master. The EV attribute updates also the static data object in the Net database. By reading the EV attribute user can inquire the number of items in the queues of unsolicited messages.

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

**FZ  Freeze Counters**

The FZ attribute can be used for freezing the binary counters of the DNP slave.

- **Data type:** Integer
- **Value:** 7...10:
  - 7: Immediate freeze
  - 9: Freeze and clear
  - 10: Freeze and clear, no ack
- **Access:** Write-only

**GI  General Interrogation**

By using the GI attribute a request for all static (class 0) data can be made. This corresponds to a general interrogation command used in other protocols.

- **Data type:** Integer
- **Value:** 1
- **Access:** Write only
### IL  Information Address Length
The length of object address (in octets) used in DNP messages.

- **Data type:** Integer
- **Value:** 1, 2
- **Default value:** 2
- **Access:** Read, conditional write

### IN  Internal Indications
The current value of the internal indications of the DNP slave station. See DNP protocol documentation for details of the internal indications.

- **Data type:** Integer
- **Value:** 0 ... 65535
- **Access:** No restrictions

### MA  Master Address
The station address of the master station, the destination address of the unsolicited messages sent by the slave.

- **Data type:** Integer
- **Value:** 0 ... 6554
- **Access:** Read, conditional write

### ML  Maximum Message Length
The maximum length (in octets) of an application data fragment.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Value</th>
<th>Unit</th>
<th>Default value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>249 ... 2048</td>
<td>Octets</td>
<td>2048</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>

### MT  Maximum Delayed Response Time
The maximum time in seconds to delay response for writing data to the SD and EV attributes, if the number of items in the queue is greater than the value of the RW attribute.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Value</th>
<th>Unit</th>
<th>Default value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>0 ... 600</td>
<td>Seconds</td>
<td>15</td>
<td>No restrictions</td>
</tr>
</tbody>
</table>
MicroSCADA Pro
System Objects
Technical Description

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

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

PC Process Data Confirmation
By setting the value of this attribute to 0 application level confirmations can be disabled, and by setting the value to 1 the confirmations can be enabled. DNP slave station sets the confirmation request of a data fragment on, regardless of the value of the PC attribute, in the following cases:

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

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

PT Pulse Length for Control Relay Output Block
The length of the pulse used in control relay output commands.

Data type: Integer
Value: 1… 4294967295
Indexing: 0 Pulse Off time
1 Pulse On time
Unit: Milliseconds
Default value: 0
Access: No restrictions

RM Running Mode
This attribute consists of a set of flags that control the behaviour and functionality of the DNP slave station. Each flag is one bit of this attribute. The bits are as follows:

Bit 0: Sending messages while waiting for a confirmation

When this bit is 0 the sending of a new message other than confirmation may not be started, if the DNP slave station is waiting for a confirmation from remote station. The message other than confirmation may be e.g. a response to a request. When this bit is 1 the sending of a new APDU other than confirmation may be started, thought the STA object is waiting for a confirmation from the master.

Bit 1: Variations in response messages

327
When this bit is 0, NET uses dynamic variations in response messages. Variations depend on the status flags of data object, variations can change between with and without status types. When this bit is 1 the variations are fixed, NE’T replies always with the same variation that was in master’s request.

Data type: Integer
Value: 1 ... 65535
Access: No restrictions

**RS**  
**Reset Event Queues**

The RS attribute can be used for clearing the content of unsolicited message queue(s).

Data type: Integer
Value: 0  Reset all queues  
1  Reset the class 1 queue  
2  Reset the class 2 queue  
3  Reset the class 3 queue

Access: Write-only

**RT**  
**Reply Time-out**

The maximum time in seconds that the DNP application layer waits for a reply from the master or slave.

Data type: Integer
Value: 0 ... 600
Unit: Seconds
Default value: 10
Access: No restrictions

**SA**  
**Slave Address**

The station address of the DNP slave station.

Data type: Integer
Value: 0 ... 65534
Access: Read, conditional write

**SD**  
**Spontaneous Data**

Writing data to the SD attribute updates the corresponding static data object (class 0) of the NET database. It does not create an unsolicited message even if the value changes. The second purpose of the SD attribute is to construct the NET database.
When a data object is updated while the DI attribute is 0, the object is added to the database if it is not previously known. There must be a data object for each input DNP address.

**Data type:** Vector  
**Value:** Vector  
**Index range:** 0 ... 3, see above  
**Access:** No restrictions

### ST  SYS Waiting Time

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

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

### SY  Synchronize

The SY attribute is used to make an accurate time synchronization of DNP stations. No time arguments are needed since the time sent in the synchronization message is taken from the internal clock of MicroSCADA. Stations can be synchronized one by one or by using a broadcast synchronization message, which synchronises all stations configured on a DNP line.

By using the do the first delay measurement option (FUNC = 2) the synchronization can be corrected with the delay of the application layer of the slave. This is done by sending a command to the slave to measure its delay. This information is then used for correcting the actual synchronization.

**Data type:** Vector  
**Value:** (FUNC, [BRO])  
**Access:** No restrictions

### TT  Transport Layer Timeout

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

**Data type:** Integer  
**Value:** 0 ... 600  
**Unit:** Seconds  
**Default value:** 10  
**Access:** No restrictions
17. PRI objects for communication system

About this chapter
This chapter describes the communication system PRI objects and their attributes:

17.1. General: printer object definition and object notation.
17.2. 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).

17.1. General

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 (Chapter 14), either in the preconfiguration or with SCIL. Defining a PRI object requires that the line has been defined as a printer line (Chapter 15). 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. When the objects are defined in the preconfiguration, only the attributes that cannot be preconfigured get the default values. The other attributes get the values given in the preconfiguration.

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 Chapter 6). 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

17.2. PRI attributes

17.2.1. Basic PRI attributes

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.

**LI Line Number**
The number of the NET line to which the printer is connected. The line is determined when the printer is created - in the preconfiguration or with the NETn:SPR attribute.

- **Data type:** Integer
- **Values:**
  - 0 Not in use
  - 1 In use
- **Default value:** 0
- **Access:** No restrictions

**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 colour information, which is not printed. Graphical characters can be replaced by printer characters using the CT attribute (section 17.2.6.).
- Character based colour printout. This type is used on FACIT 4544 printers. The characters CR and LF are generated by the communication unit.
- Pixel based colour 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.

**Device reservation**

**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 17.2.5).

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.

**Access:** Read, conditional write

- **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

---

1. Character based black-and-white printer (ASCII)
2. "Transparent printer". Printer for the full-graphic printout.
3. Pixel based, black and white printer. All EPSON FX compatible printers can be used
4. Character based "black and white" printer (ASCII)
5. Character based colour printer of type FACIT 4544
6. Pixel based colour printer. All EPSON JX compatible printers can be used.

**Access:** Read, conditional write
AL must be set to 1 in the preconfiguration.

### 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 ... 32. 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.

#### 17.2.3. System message handling

A PRI object generates system messages, for example in the following situations:

- The printer has been off-line 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 Chapter 14 and the System Configuration manual.

### 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 Chapter 13
MicroSCADA Pro
System Objects

Technical Description

**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 ... 32. The APL object number as known to the NET unit (defined by the NETn:SSY attribute).
- **Access:** No restrictions

**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 off-line. 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
```

17.2.4.  

**Diagnostic counter**

**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

### 17.2.5. Printer control

#### 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

#### 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:** Not preconfigurable, otherwise, no restrictions
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)

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

17.2.6. Printout properties

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
CT Character Transformation Table

This attribute applies to printers of type 5 (PT = 5).

An array of 256 elements that translates the MicroSCADA specific character codes to corresponding printer codes. 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. MicroSCADA character code number
Default value: CT (0 ... 64) = 43
CT (65) = A
CT (66) = B, etc.
Access: No restrictions

Example:
Characters with code values 1 ... 20 will be printed as "A":

*SET PRI1:SCT(1..20) = 65

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):

```plaintext
#SET PRI1:SPX93 = (0,30,65,1,1,65,30,0)
```
## Index

### Symbols
- " ................................................................. 71
- "FLYTECH FPC-046 ........................................... 37

### Numerics
- 844_COMPATIBLE_MIRRORING ................................................... 73

#### A
- AA ........................................................................................ 37, 58
- ABB Alpha ................................................................................. 173
- AC .............................................................................................. 67, 197
- Access ....................................................................................... 139
- Access Type .............................................................................. 232
- ACE ................................................................................................. 197
- ACE AT S Register ......................................................................... 200
- ACE State ...................................................................................... 197
- ACP .................................................................................. 23, 142, 172
- Activation Termination Waiting Time ........................................ 308
- AD .......................................................................................... 37, 148, 231
- ADLP180 Master ........................................................................... 173
- ADLP180 Slave ............................................................................ 173
- ADVANTECH PCI-1760 ................................................................. 37
- AE ............................................................................................... 104
- AL ................................................................................................. 218, 333
- Alarm Count ................................................................................ 67
- Alarm Picture Queue Handling .................................................. 64
- Alarm Tag .................................................................................... 67
- ALIAS .............................................................................................. 45
- Allocating Application ............................................................ 161, 219, 334
- Allocation .................................................................................... 218, 333
- ALLOW_CONFLICTING_F_ATTRIBUTE_NAMES .......................... 74
- AN .............................................................................................. 44, 83
- Analog Events ............................................................................. 104
- Analog Value ............................................................................... 283
- Analogue Value .......................................................................... 320
- ANSI Stations ............................................................................... 223
- ANSI X3.28 .................................................................................. 231
- ANSI X3.28 Full Duplex ............................................................... 172
- ANSI X3.28 Half Duplex .............................................................. 172
- ANSI X3.28 protocol .................................................................... 151, 156
- ANSI X3.28 Station address ....................................................... 143
- AP ................................................................................................. 49, 83
- APL ........................................................................................... 15, 16, 19, 43, 134, 136, 150
- APL objects ............................................................................... 43
- APL_EVENT .................................................................................. 75
MicroSCADA Pro
System Objects
Technical Description

APL_INIT_1, ................................................................. 138
APL_INIT_2 ................................................................. 138
APL-APL diagnostics .................................................... 75
APL-APL Server Count ................................................. 58
APL-APL Server Queue Used ......................................... 59
Application .............................................................. 13, 43, 44, 49, 134, 150
Application Diagnostic Interval ....................................... 75
Application Diagnostic Status ......................................... 76
Application Diagnostic Timeout ...................................... 75
Application diagnostics ................................................ 75
Application Layer ....................................................... 300
Application layer ........................................................ 297
Application Mapping .................................................... 49
Application Message Data Retries ................................. 322
Application Monitor Number ......................................... 83
Application Number .................................................... 44, 83
Application Reply Timeout ........................................... 309
Application Reply Wait Time ......................................... 161
Application Response Timeout ........................................ 322
Application State ........................................................ 44
Application Suspension Time ......................................... 160
Application Wait Time .................................................. 193
AQ .................................................................................. 64
AR .................................................................................. 322
AS .................................................................................... 44, 197, 218, 219, 237, 334
ASCII printer protocol .................................................. 172
ASCII protocol ............................................................ 177
AT .................................................................................... 67, 232, 322
Attribute .......................................................................... 14
Attribute name ............................................................ 14
AU .................................................................................... 59
Audio Alarm Address ..................................................... 37
Audio Alarm Device ....................................................... 37
Audio Watchdog Cycle ................................................... 37
Audio-visual alarm unit .................................................. 37
Autocaller ....................................................................... 197, 200, 209
AV .................................................................................... 283, 320
AW .................................................................................... 37, 193

B
Base system object .......................................................... 11
Base System Object Name ............................................... 18
base system object names .............................................. 15
Base System Object Navigator tool ................................. 125
Base System Object Number .......................................... 18
Base System Object Type ............................................... 18
base system objects ...................................................... 16, 19
# MicroSCADA Pro System Objects

## Technical Description

<table>
<thead>
<tr>
<th>Basic Attributes ........................................................................</th>
<th>217, 240, 250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud Rate ...............................................................................</td>
<td>179</td>
</tr>
<tr>
<td>BF ..........................................................................................</td>
<td>232</td>
</tr>
<tr>
<td>BL ..........................................................................................</td>
<td>250</td>
</tr>
<tr>
<td>Blink Policy ..........................................................................</td>
<td>84</td>
</tr>
<tr>
<td>Block Format .........................................................................</td>
<td>232</td>
</tr>
<tr>
<td>BM .......................................................................................</td>
<td>18</td>
</tr>
<tr>
<td>BN .......................................................................................</td>
<td>18</td>
</tr>
<tr>
<td>BO ........................................................................................</td>
<td>202</td>
</tr>
<tr>
<td>BP ........................................................................................</td>
<td>84</td>
</tr>
<tr>
<td>BR ........................................................................................</td>
<td>179</td>
</tr>
<tr>
<td>Break Down ..........................................................................</td>
<td>202</td>
</tr>
<tr>
<td>Broadcast ............................................................................</td>
<td>241, 316</td>
</tr>
<tr>
<td>Broadcast Lines ...................................................................</td>
<td>250</td>
</tr>
<tr>
<td>Broadcast station ..................................................................</td>
<td>156</td>
</tr>
<tr>
<td>Broadcast Stations ................................................................</td>
<td>217</td>
</tr>
<tr>
<td>BT .......................................................................................</td>
<td>18</td>
</tr>
<tr>
<td>BU ........................................................................................</td>
<td>202</td>
</tr>
<tr>
<td>Buffer Pool Size ..................................................................</td>
<td>178</td>
</tr>
<tr>
<td>Build Up ...............................................................................</td>
<td>202</td>
</tr>
</tbody>
</table>

### C

| CA .......................................................................................... | 37, 307, 322 |
| Cause of Transmission Mapping ......................................... | 111          |
| CB ....................................................................................... | 291          |
| CBXC Timeout ....................................................................... | 292          |
| CC ....................................................................................... | 337          |
| CD ....................................................................................... | 38, 336      |
| CF ....................................................................................... | 39, 203, 300, 323 |
| Character Transformation Table ....................................... | 338          |
| Checksum ............................................................................ | 93           |
| CL ....................................................................................... | 39, 122, 198, 307 |
| Class 1 ............................................................................... | 298          |
| Class 2 ............................................................................... | 298          |
| Clear to Send ...................................................................... | 183          |
| Clock Status ......................................................................... | 40, 162      |
| Clock Synchronisation ...................................................... | 240          |
| CN ....................................................................................... | 198          |
| CO ....................................................................................... | 233, 301, 323 |
| Coding ............................................................................... | 233          |
| COLD .................................................................................... | 44, 45       |
| Collector RTUs ...................................................................... | 240          |
| COLOR .................................................................................. | 114          |
| Color Allocation Policy .................................................... | 66, 85       |
| Color Conversion .................................................................. | 337          |
| COM port ............................................................................. | 169          |
| COMLI Slave protocol ........................................................ | 173          |
### MicroSCADA Pro System Objects Technical Description

<table>
<thead>
<tr>
<th>Description</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Address</td>
<td>307, 322</td>
</tr>
<tr>
<td>Command Blocked from Master Station</td>
<td>291</td>
</tr>
<tr>
<td>Command Confirmation</td>
<td>300, 323</td>
</tr>
<tr>
<td>Command Out</td>
<td>301, 323</td>
</tr>
<tr>
<td>Command Poll Count</td>
<td>193, 224</td>
</tr>
<tr>
<td>Comment Text</td>
<td>42, 79, 87, 88, 95, 102, 104, 111, 123</td>
</tr>
<tr>
<td>Common RAM</td>
<td>169</td>
</tr>
<tr>
<td>COMMON RAM protocol</td>
<td>172</td>
</tr>
<tr>
<td>Common STA Attributes</td>
<td>217</td>
</tr>
<tr>
<td>Communication</td>
<td>11, 25</td>
</tr>
<tr>
<td>Communication loop</td>
<td>201, 202</td>
</tr>
<tr>
<td>Communication Loop Attributes</td>
<td>249</td>
</tr>
<tr>
<td>Communication system object</td>
<td>11, 135</td>
</tr>
<tr>
<td>Communication unit</td>
<td>134</td>
</tr>
<tr>
<td>Compatibility issues</td>
<td>70</td>
</tr>
<tr>
<td>Confirmation Timeout</td>
<td>323</td>
</tr>
<tr>
<td>Connected Station</td>
<td>199</td>
</tr>
<tr>
<td>Connection</td>
<td>198</td>
</tr>
<tr>
<td>Connection Time</td>
<td>199</td>
</tr>
<tr>
<td>Connection Time Limited</td>
<td>198</td>
</tr>
<tr>
<td>Consistency Check Time</td>
<td>277</td>
</tr>
<tr>
<td>Control Data</td>
<td>336</td>
</tr>
<tr>
<td>Control Data Store</td>
<td>336</td>
</tr>
<tr>
<td>Control Sequences</td>
<td>117</td>
</tr>
<tr>
<td>Counter Attributes</td>
<td>226</td>
</tr>
<tr>
<td>Counters and Timers</td>
<td>226</td>
</tr>
<tr>
<td>COUPLE_AUDIO_ALARMS_AND_PRINTOUTS</td>
<td>74</td>
</tr>
<tr>
<td>CP</td>
<td>66, 85, 193, 224</td>
</tr>
<tr>
<td>CREATE</td>
<td>19</td>
</tr>
<tr>
<td>CREATE_VERSION_1_FILES</td>
<td>72</td>
</tr>
<tr>
<td>CREATE_VERSION_2_SCIL_DATABASES</td>
<td>74</td>
</tr>
<tr>
<td>CS</td>
<td>40, 117, 162, 199, 336</td>
</tr>
<tr>
<td>CT</td>
<td>111, 199, 226, 277, 292, 308, 323, 338</td>
</tr>
<tr>
<td>CTS</td>
<td>183</td>
</tr>
<tr>
<td>CTS Delay</td>
<td>182</td>
</tr>
<tr>
<td>CX</td>
<td>42, 79, 87, 88, 95, 102, 104, 111, 123</td>
</tr>
<tr>
<td>D</td>
<td>159, 244, 258, 270, 277, 337</td>
</tr>
<tr>
<td>Data</td>
<td>258, 270, 277, 337</td>
</tr>
<tr>
<td>Data Interface</td>
<td>111</td>
</tr>
<tr>
<td>Data Point</td>
<td>324</td>
</tr>
<tr>
<td>Data transmission</td>
<td>179</td>
</tr>
<tr>
<td>Data Type</td>
<td>233</td>
</tr>
<tr>
<td>Data type</td>
<td>14</td>
</tr>
<tr>
<td>Database Initialised</td>
<td>292, 324</td>
</tr>
<tr>
<td>DB</td>
<td>111</td>
</tr>
</tbody>
</table>
MicroSCADA Pro

System Objects

Technical Description

DC . 83, 89, 94, 114, 162, 207, 227, 242, 253, 278, 292, 310, 316, 323, 335
DCP-NET ................................................................. 169
DD ........................................................................... 34
DDE server ............................................................... 34
DDE Server Diagnostics ............................................. 25
DDE Server Enabled ................................................... 34
DDE Server in Use ...................................................... 34
DE .. 34, 182, 183, 228
Default NET Node Number ......................................... 25
Default STA Type ...................................................... 26
DEFAULT_DAYLIGHT_POLICY_ISCALENDAR .................. 73
Delay measurement .................................................. 329
Device ................................................................. 133, 152, 157
Device Connection .................................................... 34
Device image ............................................................ 133
Device mapping ......................................................... 34
Device names ............................................................ 34
Device number .......................................................... 134
Device Reservation ................................................... 218
Device Type ............................................................. 82, 89, 114
DF ................................................................. 85, 100, 163, 229, 278, 292, 320, 324
Diagnostic Attributes ................................................ 226, 253
Diagnostic Counter .................................................... 162, 335
Diagnostic counter ..................................................... 242
Diagnostic Counters .................................................. 54, 94, 207, 227, 242, 253, 278, 292, 310, 316, 323
Diagnostic Counters for APL connections ................... 159
Diagnostic Event from First Found ................................. 100
Diagnostic Interval ................................................... 85, 100, 163, 229, 278, 292, 320, 324
Diagnostic Status ..................................................... 160, 229
Diagnostic Timeout .................................................. 101
Diagnostics Enable ................................................... 228
Digital Value ............................................................. 321
Direction ................................................................. 203, 249, 311
DN ................................................................. 75, 82, 89, 101, 114, 233
DNP ................................................................. 85, 100, 163, 229, 278, 292, 320, 324
DO NOT_SYNCHRONIZE_PICTURE_UPDATE .................. 74
Double Indication ..................................................... 284, 320
DP ................................................................. 203, 249, 311
DR ................................................................. 284, 320
DS ................................................................. 284, 320
DT ................................................................. 284, 320
DU ................................................................. 284, 320
DV ................................................................. 284, 320

E
EC ................................................................. 252, 293
<table>
<thead>
<tr>
<th>Technical Description</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>ED</td>
<td>85, 158, 254</td>
</tr>
<tr>
<td>EE</td>
<td>64</td>
</tr>
<tr>
<td>EI</td>
<td>285</td>
</tr>
<tr>
<td>EL</td>
<td>252</td>
</tr>
<tr>
<td>EM</td>
<td>59</td>
</tr>
<tr>
<td>Embedded Response</td>
<td>180</td>
</tr>
<tr>
<td>Embedded response</td>
<td>180</td>
</tr>
<tr>
<td>EN</td>
<td>93, 184, 230</td>
</tr>
<tr>
<td>Enable Routing</td>
<td>26</td>
</tr>
<tr>
<td>ENQ Limit</td>
<td>93</td>
</tr>
<tr>
<td>Enquiry Limit</td>
<td>184</td>
</tr>
<tr>
<td>Enter Key Disabled</td>
<td>85</td>
</tr>
<tr>
<td>EO</td>
<td>325</td>
</tr>
<tr>
<td>EP</td>
<td>76, 252</td>
</tr>
<tr>
<td>ER</td>
<td>26, 180</td>
</tr>
<tr>
<td>Ermi from Indications</td>
<td>285</td>
</tr>
<tr>
<td>EU</td>
<td>59</td>
</tr>
<tr>
<td>EV</td>
<td>302, 325</td>
</tr>
<tr>
<td>Event Buffer Length</td>
<td>252</td>
</tr>
<tr>
<td>Event Data with Time Stamp</td>
<td>302, 325</td>
</tr>
<tr>
<td>Event History Start Time</td>
<td>273</td>
</tr>
<tr>
<td>Event Offset</td>
<td>325</td>
</tr>
<tr>
<td>Event Poll Priority Class</td>
<td>252</td>
</tr>
<tr>
<td>Event Queue Length Maximum</td>
<td>59</td>
</tr>
<tr>
<td>Event Queue Overflow Policy</td>
<td>76</td>
</tr>
<tr>
<td>Event Queue Used</td>
<td>59</td>
</tr>
<tr>
<td>Event Recording Message</td>
<td>286</td>
</tr>
<tr>
<td>Event to Data</td>
<td>254</td>
</tr>
<tr>
<td>Event to Data Consistency Check period</td>
<td>252</td>
</tr>
<tr>
<td>EX</td>
<td>286</td>
</tr>
<tr>
<td>Exchangeable Device table</td>
<td>158</td>
</tr>
<tr>
<td>Exchangeable device types</td>
<td>157</td>
</tr>
<tr>
<td>Exclusive Commands from Master Station</td>
<td>293</td>
</tr>
<tr>
<td>Extended Address Table</td>
<td>206</td>
</tr>
<tr>
<td>EXTERNAL</td>
<td>45</td>
</tr>
<tr>
<td>External Clock</td>
<td>39</td>
</tr>
<tr>
<td>External clock</td>
<td>37</td>
</tr>
<tr>
<td>External Clock Address</td>
<td>37</td>
</tr>
<tr>
<td>External Clock Data</td>
<td>38</td>
</tr>
<tr>
<td>External Clock Read Frequency</td>
<td>39</td>
</tr>
<tr>
<td>External node</td>
<td>149</td>
</tr>
<tr>
<td><strong>F</strong></td>
<td></td>
</tr>
<tr>
<td>Fast Select during Suspension</td>
<td>225</td>
</tr>
<tr>
<td>FC</td>
<td>243</td>
</tr>
<tr>
<td>File Sync Criterion</td>
<td>29</td>
</tr>
<tr>
<td>File Transfer Progress</td>
<td>275</td>
</tr>
<tr>
<td>Technical Description</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td></td>
</tr>
<tr>
<td>File Transfer Timeout .......................................................... 275</td>
<td></td>
</tr>
<tr>
<td>FILE_FUNCTIONS_CREATE_DIRECTORIES ........................................ 70</td>
<td></td>
</tr>
<tr>
<td>FM .......................................................................................... 165</td>
<td></td>
</tr>
<tr>
<td>FO .......................................................................................... 275</td>
<td></td>
</tr>
<tr>
<td>FORM_FEED .............................................................................. 119</td>
<td></td>
</tr>
<tr>
<td>FP .......................................................................................... 275</td>
<td></td>
</tr>
<tr>
<td>Free Memory ........................................................................... 165</td>
<td></td>
</tr>
<tr>
<td>Freeze Counters .................................................................. 325</td>
<td></td>
</tr>
<tr>
<td>Front-end ............................................................................ 134, 161</td>
<td></td>
</tr>
<tr>
<td>FS ....................................................................................... 29, 225</td>
<td></td>
</tr>
<tr>
<td>FT ....................................................................................... 243, 281</td>
<td></td>
</tr>
<tr>
<td>Function Command ................................................................ 243</td>
<td></td>
</tr>
<tr>
<td>Function Table ..................................................................... 243</td>
<td></td>
</tr>
<tr>
<td>Function Table (FTAB) ......................................................... 281</td>
<td></td>
</tr>
<tr>
<td>FZ ....................................................................................... 325</td>
<td></td>
</tr>
<tr>
<td>GD .......................................................................................... 321</td>
<td></td>
</tr>
<tr>
<td>General ASCII ....................................................................... 173</td>
<td></td>
</tr>
<tr>
<td>General Interrogation ......................................................... 271, 277, 303, 325</td>
<td></td>
</tr>
<tr>
<td>General Object Handling ....................................................... 271</td>
<td></td>
</tr>
<tr>
<td>General request of Data ........................................................ 321</td>
<td></td>
</tr>
<tr>
<td>GI ....................................................................................... 271, 277, 303, 325</td>
<td></td>
</tr>
<tr>
<td>Global Paths .......................................................................... 33</td>
<td></td>
</tr>
<tr>
<td>Global variable ....................................................................... 42</td>
<td></td>
</tr>
<tr>
<td>GO ....................................................................................... 271</td>
<td></td>
</tr>
<tr>
<td>Hardware ............................................................................... 23</td>
<td></td>
</tr>
<tr>
<td>HB ........................................................................................ 60</td>
<td></td>
</tr>
<tr>
<td>HD ........................................................................................ 78</td>
<td></td>
</tr>
<tr>
<td>HE ........................................................................................ 77</td>
<td></td>
</tr>
<tr>
<td>Header Format ....................................................................... 118</td>
<td></td>
</tr>
<tr>
<td>Header Timeout ..................................................................... 185</td>
<td></td>
</tr>
<tr>
<td>HF ........................................................................................ 118</td>
<td></td>
</tr>
<tr>
<td>HI ....................................................................................... 273</td>
<td></td>
</tr>
<tr>
<td>Historical Events ................................................................... 273</td>
<td></td>
</tr>
<tr>
<td>History Buffer Length .......................................................... 60</td>
<td></td>
</tr>
<tr>
<td>History logging Policy ............................................................ 65</td>
<td></td>
</tr>
<tr>
<td>History Tag .......................................................................... 67</td>
<td></td>
</tr>
<tr>
<td>Host Diagnostics ..................................................................... 78</td>
<td></td>
</tr>
<tr>
<td>Host Enabled .......................................................................... 77</td>
<td></td>
</tr>
<tr>
<td>Host RTU ............................................................................... 240</td>
<td></td>
</tr>
<tr>
<td>Host Station .......................................................................... 104</td>
<td></td>
</tr>
<tr>
<td>HOT .................................................................................... 44</td>
<td></td>
</tr>
<tr>
<td>Hot Stand-by base system ...................................................... 34</td>
<td></td>
</tr>
<tr>
<td>HP ....................................................................................... 65</td>
<td></td>
</tr>
</tbody>
</table>
## SYS 600 *9.1.5 System Objects Technical Description

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td>240, 241</td>
</tr>
<tr>
<td>HS</td>
<td>104, 273</td>
</tr>
<tr>
<td>HT</td>
<td>67, 185, 186</td>
</tr>
<tr>
<td>HW</td>
<td>23</td>
</tr>
<tr>
<td>ID</td>
<td>78, 288, 306</td>
</tr>
<tr>
<td>IE</td>
<td>77</td>
</tr>
<tr>
<td>IEC</td>
<td>108</td>
</tr>
<tr>
<td>IEC 60870 slave device</td>
<td>297</td>
</tr>
<tr>
<td>IEC 60870-5-101 Protocol</td>
<td>298, 299</td>
</tr>
<tr>
<td>IEC 60870-5-103 Protocol</td>
<td>298, 299, 300</td>
</tr>
<tr>
<td>IEC Station</td>
<td>297</td>
</tr>
<tr>
<td>IL</td>
<td>86, 272, 308, 326</td>
</tr>
<tr>
<td>Image Diagnostics</td>
<td>78</td>
</tr>
<tr>
<td>Image Enabled</td>
<td>77</td>
</tr>
<tr>
<td>Image Stations</td>
<td>105</td>
</tr>
<tr>
<td>Image Stations for System Messages</td>
<td>77</td>
</tr>
<tr>
<td>IN</td>
<td>326</td>
</tr>
<tr>
<td>In Use</td>
<td>176, 217, 331</td>
</tr>
<tr>
<td>IND</td>
<td>15, 17, 88</td>
</tr>
<tr>
<td>Index</td>
<td>14</td>
</tr>
<tr>
<td>Index range</td>
<td>14</td>
</tr>
<tr>
<td>Indications</td>
<td>288</td>
</tr>
<tr>
<td>Information Address Length</td>
<td>308, 326</td>
</tr>
<tr>
<td>Initialization file</td>
<td>137</td>
</tr>
<tr>
<td>Input Locked</td>
<td>86</td>
</tr>
<tr>
<td>Integrated link</td>
<td>91, 173</td>
</tr>
<tr>
<td>Interlocking Data</td>
<td>272</td>
</tr>
<tr>
<td>Internal Indications</td>
<td>326</td>
</tr>
<tr>
<td>IS</td>
<td>77, 105</td>
</tr>
<tr>
<td>IU</td>
<td>176, 217, 331</td>
</tr>
</tbody>
</table>

### K
- **KEEP_FILE_VERSION_1_DATABASE_FILES** | 72
- **Known Protocols** | 144
- **KP** | 144

### L
- **LA** | 66, 86, 213
- **LAN** | 98
- **LAN link** | 91
- **LAN Node Name** | 98
- **Language** | 66, 86
- **Last Error Status** | 230
- **Last Transaction** | 101
- **Last Transaction number** | 166
- **LC** | 153, 296
<table>
<thead>
<tr>
<th>System Object</th>
<th>Reference Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCU</td>
<td>107, 151</td>
</tr>
<tr>
<td>LCU500</td>
<td>173</td>
</tr>
<tr>
<td>LD</td>
<td>120</td>
</tr>
<tr>
<td>LE</td>
<td>234</td>
</tr>
<tr>
<td>Length</td>
<td>234</td>
</tr>
<tr>
<td>Length of Cause of Transmission Info</td>
<td>307</td>
</tr>
<tr>
<td>Length of Time-out</td>
<td>230</td>
</tr>
<tr>
<td>LF</td>
<td>121</td>
</tr>
<tr>
<td>LI</td>
<td>83, 98, 149, 218, 296, 315, 332</td>
</tr>
<tr>
<td>LIN</td>
<td>15, 17, 91, 95, 98</td>
</tr>
<tr>
<td>Line</td>
<td>137, 169, 173</td>
</tr>
<tr>
<td>Line Number</td>
<td>118, 149, 218, 296, 315, 332</td>
</tr>
<tr>
<td>Lines per Page</td>
<td>118</td>
</tr>
<tr>
<td>Link</td>
<td>91</td>
</tr>
<tr>
<td>Link Layer Confirmations Enabled</td>
<td>213</td>
</tr>
<tr>
<td>Link Number</td>
<td>98</td>
</tr>
<tr>
<td>Link Type</td>
<td>92, 176, 210</td>
</tr>
<tr>
<td>LK</td>
<td>176, 210</td>
</tr>
<tr>
<td>LL</td>
<td>121</td>
</tr>
<tr>
<td>LM</td>
<td>153, 277</td>
</tr>
<tr>
<td>LMK</td>
<td>151, 276</td>
</tr>
<tr>
<td>LMK Stations</td>
<td>153</td>
</tr>
<tr>
<td>LN</td>
<td>118</td>
</tr>
<tr>
<td>Load Control Policy</td>
<td>105, 112</td>
</tr>
<tr>
<td>Load Control Unit</td>
<td>153</td>
</tr>
<tr>
<td>LOCAL</td>
<td>45</td>
</tr>
<tr>
<td>Local application</td>
<td>44</td>
</tr>
<tr>
<td>Log Directory</td>
<td>120</td>
</tr>
<tr>
<td>Log Flush Timeout</td>
<td>121</td>
</tr>
<tr>
<td>Log Length</td>
<td>121</td>
</tr>
<tr>
<td>Logged In</td>
<td>83</td>
</tr>
<tr>
<td>Logical application number</td>
<td>49</td>
</tr>
<tr>
<td>Logical device number</td>
<td>50</td>
</tr>
<tr>
<td>Lon Message</td>
<td>277</td>
</tr>
<tr>
<td>Lon Point</td>
<td>279</td>
</tr>
<tr>
<td>LonTalk protocol</td>
<td>173</td>
</tr>
<tr>
<td>LonTalk protocol line</td>
<td>205</td>
</tr>
<tr>
<td>LONWORKS channels</td>
<td>169</td>
</tr>
<tr>
<td>Loop Configuration Table</td>
<td>203</td>
</tr>
<tr>
<td>Loop Control</td>
<td>296</td>
</tr>
<tr>
<td>Loop Mode In Use</td>
<td>204, 250</td>
</tr>
<tr>
<td>Loop Reversal Unit</td>
<td>201</td>
</tr>
<tr>
<td>Loop Status</td>
<td>204</td>
</tr>
<tr>
<td>Loop Switch</td>
<td>250</td>
</tr>
<tr>
<td>Loop Timeout</td>
<td>296</td>
</tr>
<tr>
<td>LP</td>
<td>105, 112, 118, 279</td>
</tr>
<tr>
<td>LS</td>
<td>204, 230, 249</td>
</tr>
</tbody>
</table>
LT ................................................................. 92, 101, 166, 296
LU ............................................................................. 204, 250
LW .............................................................................. 250

M
MA ................................................................. 166, 326
Mail Update Identification .............................................. 166
Mailbox ........................................................................ 166
Master Address .......................................................... 326
Master mode ............................................................... 297
Maximum Delayed Response Time ................................. 326
Maximum Message Length ............................................ 212, 311, 326
Maximum Random Delay for Retransmission .................. 212
Maximum Response Delayed Time ............................... 311
Maximum Scan Time .................................................... 205
MC ............................................................................ 199, 234
MD ............................................................................... 204
ME .............................................................................. 46, 239
Memory ................................................................. 29, 239
Memory area ............................................................. 231, 237
Memory Area Definitions .............................................. 231
Memory Block Size ..................................................... 31
Memory Blocks Free ................................................... 31
Memory Blocks Used ................................................... 32
Memory Configuration .................................................. 234
Memory Pool Sizes ...................................................... 31
Memory pool supervision Enabled .................................. 46
Memory Rung ............................................................. 234
Message Application .................................................. 147, 179, 221, 335
Message Identification ................................................ 146, 179, 220, 221, 334
Message Split ............................................................ 237, 239
Message split ............................................................. 219, 238
MF ............................................................................. 31
MI ........................................................ 145, 146, 179, 220, 221, 334
Mirroring ................................................................. 56, 59, 76, 104
Mirroring Role ............................................................ 106
ML .............................................................................. 212, 311, 326
MM ............................................................................... 293
MO .............................................................................. 51, 89
MODBUS .............................................................. 173
Modbus Protocol Converter .......................................... 315
Modem Command ....................................................... 199
Modem Delay ........................................................... 204
Modem Signal ............................................................. 188
MON ................................................................. 15, 17, 19, 81
MON_EVENT ........................................................... 83
Monitor Alarm Signal Size .......................................... 67
Monitor Mapping ................................................................. 51
Monitor Number .................................................................. 89
Monitor Stop ....................................................................... 86
Monitors .............................................................................. 81
MP .......................................................................................... 31
MR ................................................................. 106, 231, 234
MS .................................................................................. 31, 67, 86, 145, 147, 166, 179, 221, 335
MT .................................................................................. 205, 311, 326
MU ................................................................................ 32, 145, 166
MW ................................................................................... 82

N
NA ................................................................. 45, 93, 112, 186, 230
NAK ............................................................................. 93, 186
NAK Limit ................................................................. 93, 186
Name ............................................................................. 13, 45
NC .................................................................................. 205
ND ................................................................. 22, 48, 106, 116
NE .................................................................................. 141, 149
NET ................................................................. 106, 133, 134, 135, 141, 149
NET line attributes .......................................................... 170
NET unit line ................................................................. 170
NETCONF .............................................................................. 137
Network Variable Configuration ........................................ 205
NN ................................................................. 23, 98, 142, 267, 276
No restrictions ................................................................. 20, 140
NO_ALARM_BY_OR_AND_OF ......................................................... 72
NO_ALIAS_CHECKING .......................................................... 71
NO_QUALITY_ATTRIBUTE_SEMANTICS ..................................... 71
NOD ................................................................. 15, 17, 97, 133, 134, 135, 141, 149
Node ............................................................................. 91, 135, 149, 161
Node Name ..................................................................... 23
Node Number ............................................................... 22, 25, 48, 106, 116, 142, 267, 276
Node numbers .................................................................. 134
Node Type ..................................................................... 98, 144
NORMAL .............................................................................. 114
Not configurable, otherwise no limitations ......................... 20
Not preconfigurable, otherwise no restrictions .................. 139
NT .................................................................................. 98, 144
NUDAQ PCI-7250 ................................................................. 37
NUDAQ PCI-7256 ................................................................. 37
Number of Enquiries .......................................................... 230
Number of NAKs ............................................................. 230

O
Object notation .............................................................. 12, 14
Object number .............................................................. 15, 137
### System Objects

#### Technical Description

<table>
<thead>
<tr>
<th>Object Status</th>
<th>222, 327, 335</th>
</tr>
</thead>
<tbody>
<tr>
<td>OD</td>
<td>121</td>
</tr>
<tr>
<td>OE</td>
<td>41</td>
</tr>
<tr>
<td>OJ</td>
<td>122</td>
</tr>
<tr>
<td>OL</td>
<td>282</td>
</tr>
<tr>
<td>OM</td>
<td>23</td>
</tr>
<tr>
<td>ON_COMMAND_EXPANSION</td>
<td>70</td>
</tr>
<tr>
<td>On-line</td>
<td>137</td>
</tr>
<tr>
<td>On-line configuration</td>
<td>138</td>
</tr>
<tr>
<td>OP</td>
<td>36, 102</td>
</tr>
<tr>
<td>OPC Alarm and Event Server</td>
<td>97, 108</td>
</tr>
<tr>
<td>OPC Server</td>
<td>36, 102</td>
</tr>
<tr>
<td>Open on Job Basis</td>
<td>122</td>
</tr>
<tr>
<td>Operating System</td>
<td>24</td>
</tr>
<tr>
<td>Operating System Event Handler</td>
<td>41</td>
</tr>
<tr>
<td>Operating System Event Handler Filter</td>
<td>41</td>
</tr>
<tr>
<td>Operating System Minor Version</td>
<td>23</td>
</tr>
<tr>
<td>Operating System Version</td>
<td>24</td>
</tr>
<tr>
<td>OS</td>
<td>24, 186, 222, 327, 335</td>
</tr>
<tr>
<td>OT</td>
<td>41</td>
</tr>
<tr>
<td>Output Destination</td>
<td>121</td>
</tr>
<tr>
<td>Output Synchronisation</td>
<td>186</td>
</tr>
<tr>
<td>OV</td>
<td>24</td>
</tr>
<tr>
<td>Overflow Limit</td>
<td>282</td>
</tr>
</tbody>
</table>

#### P

| P214 protocol          | 173, 174 |
| PA                     | 36, 224, 308 |
| PAC                    | 107, 151  |
| PAC - 5                | 153       |
| PAC-5                  | 172       |
| Page Number            | 119       |
| Parallel Queues        | 60        |
| Parameter Reservation  | 258       |
| Parity                 | 181       |
| Paths                  | 46        |
| PC                     | 32, 86, 153, 154, 289, 308, 327 |
| PCL                    | 107, 151  |
| PCLTA card             | 175       |
| PC-NET                 | 169       |
| PD                     | 187, 193  |
| PE                     | 337       |
| Peer Node Number       | 164       |
| PH                     | 33, 46, 223 |
| Phone Number           | 223       |
| Phone number           | 223       |
| Physical application number | 49 |
MicroSCADA Pro
System Objects
Technical Description

- Physical device number ................................................. 50
- Picture Cache Size .................................................... 32
- Picture Cache Used .................................................... 32
- Picture Containers ...................................................... 86
- Picture containers ...................................................... 86
- Pixel Characters .......................................................... 338
- PL ............................................................................... 194, 309
- PLC Station ................................................................. 315
- PM ............................................................................... 60, 177
- PN ............................................................................... 119, 164
- PO ............................................................................... 172, 173
- Poll Delay ................................................................... 193
- Polling .......................................................................... 191
- Polling Address .......................................................... 224
- Polling Address Length, Link Address Length ............. 309
- Polling Address, Address of Link .................................. 308
- Polling Attributes
  - ANSI ........................................................................... 224
  - Polling Delay ............................................................ 193
  - Polling frequency .................................................... 194
  - Polling limit .............................................................. 194
  - Polling Period .......................................................... 194
  - Polling Ratio ............................................................. 195
- Post-processing Policy for Object Status 2 ................. 65, 106
- PP ............................................................................... 65, 106, 194, 195
- PQ ............................................................................... 60
- PR ............................................................................... 24, 52, 158, 258
- Preconfiguration ............................................................ 137
- PRI ............................................................................... 15, 17, 113, 115, 120, 134, 136, 152, 158, 331
- Primary Application ....................................................... 36
- Print Enable ................................................................. 337
- PRINT_TRANSPARENT .................................................. 117
- Printer .......................................................................... 113, 115, 116, 134, 136, 158, 331
- Printer Control ............................................................. 122
- Printer control .............................................................. 336
- Printer Log .................................................................. 113, 119
- Printer log ................................................................... 119
- Printer Mapping ............................................................ 52
- Printer Queue ............................................................... 119
- Printer Spool Stop ......................................................... 68
- Printer Spooler Queue Length Maximum .................. 60
- Printer Spooler Queue Length Used .......................... 60
- Printer State ................................................................. 123
- Printer status ............................................................... 335
- Printer Type ................................................................. 332
- PRIVATE ....................................................................... 66
- Process Communication ............................................... 244
<table>
<thead>
<tr>
<th>System Object</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process communication</td>
<td>133</td>
</tr>
<tr>
<td>Process Data</td>
<td>244</td>
</tr>
<tr>
<td>Process Data Confirmation</td>
<td>308, 327</td>
</tr>
<tr>
<td>Process Query Length</td>
<td>62</td>
</tr>
<tr>
<td>Procontrol P214</td>
<td>154</td>
</tr>
<tr>
<td>PROD_QUERY</td>
<td>62</td>
</tr>
<tr>
<td>Product Name</td>
<td>24</td>
</tr>
<tr>
<td>Program Revision</td>
<td>25</td>
</tr>
<tr>
<td>Program Version</td>
<td>144</td>
</tr>
<tr>
<td>Protocol</td>
<td>144, 170, 172</td>
</tr>
<tr>
<td>Protocol Mode</td>
<td>177</td>
</tr>
<tr>
<td>PS</td>
<td>68, 178</td>
</tr>
<tr>
<td>PT</td>
<td>195, 327, 332</td>
</tr>
<tr>
<td>PU</td>
<td>32, 60, 200</td>
</tr>
<tr>
<td>Pulse Counter</td>
<td>289</td>
</tr>
<tr>
<td>Pulse Dialling</td>
<td>200</td>
</tr>
<tr>
<td>Pulse Length for Control Relay Output Block</td>
<td>327</td>
</tr>
<tr>
<td>PX</td>
<td>338</td>
</tr>
<tr>
<td>PY</td>
<td>181</td>
</tr>
<tr>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>QD</td>
<td>61</td>
</tr>
<tr>
<td>QE</td>
<td>46</td>
</tr>
<tr>
<td>QL</td>
<td>62</td>
</tr>
<tr>
<td>QM</td>
<td>62, 119</td>
</tr>
<tr>
<td>QO</td>
<td>61</td>
</tr>
<tr>
<td>QP</td>
<td>63</td>
</tr>
<tr>
<td>QU</td>
<td>63, 119</td>
</tr>
<tr>
<td>Queue Dedication</td>
<td>61</td>
</tr>
<tr>
<td>Queue Length Maximum</td>
<td>119</td>
</tr>
<tr>
<td>Queue Length Used</td>
<td>119</td>
</tr>
<tr>
<td>Queue Maximum</td>
<td>62</td>
</tr>
<tr>
<td>Queue Priority</td>
<td>63</td>
</tr>
<tr>
<td>Queue supervision Enabled</td>
<td>46</td>
</tr>
<tr>
<td>Queue Used</td>
<td>63</td>
</tr>
<tr>
<td>Queued Objects</td>
<td>61</td>
</tr>
<tr>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Radio clock</td>
<td>37</td>
</tr>
<tr>
<td>Radio Connection Wait Time</td>
<td>200</td>
</tr>
<tr>
<td>Radio Disconnection Delay</td>
<td>199</td>
</tr>
<tr>
<td>RAM</td>
<td>93</td>
</tr>
<tr>
<td>RAM device</td>
<td>93</td>
</tr>
<tr>
<td>RAM links</td>
<td>91</td>
</tr>
<tr>
<td>RC</td>
<td>32, 69, 200</td>
</tr>
<tr>
<td>RCOM</td>
<td>173</td>
</tr>
<tr>
<td>RCT</td>
<td>107</td>
</tr>
</tbody>
</table>
## System Objects

### Technical Description

<table>
<thead>
<tr>
<th>Object</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD</td>
<td>25, 181, 245</td>
</tr>
<tr>
<td>RE</td>
<td>25, 93, 181</td>
</tr>
<tr>
<td>Read, conditional write</td>
<td>20, 139</td>
</tr>
<tr>
<td>Read-only, configurable</td>
<td>20</td>
</tr>
<tr>
<td>Reads Transparent Data</td>
<td>245</td>
</tr>
<tr>
<td>Receive Interrupt Enable Delay</td>
<td>187</td>
</tr>
<tr>
<td>Receive Quota</td>
<td>270</td>
</tr>
<tr>
<td>Receiver Data Bit Count</td>
<td>181</td>
</tr>
<tr>
<td>Recovery</td>
<td>225</td>
</tr>
<tr>
<td>Redundancy</td>
<td>93, 181</td>
</tr>
<tr>
<td>Redundant Frontend Counters</td>
<td>165</td>
</tr>
<tr>
<td>Redundant Frontend Diagnostic Cycle</td>
<td>165</td>
</tr>
<tr>
<td>Redundant Frontend timeout</td>
<td>165</td>
</tr>
<tr>
<td>Redundant front-ends</td>
<td>163</td>
</tr>
<tr>
<td>Redundant Line Station</td>
<td>297, 315</td>
</tr>
<tr>
<td>Reference</td>
<td>152, 156, 158</td>
</tr>
<tr>
<td>Registration Time</td>
<td>101</td>
</tr>
<tr>
<td>Remote Calls Enabled</td>
<td>200</td>
</tr>
<tr>
<td>Repeat Time Delay</td>
<td>187</td>
</tr>
<tr>
<td>Reply Polling</td>
<td>196</td>
</tr>
<tr>
<td>Reply Time-out</td>
<td>226, 253, 316, 328</td>
</tr>
<tr>
<td>Reply Timeout</td>
<td>245, 274, 278</td>
</tr>
<tr>
<td>Reply Window Size</td>
<td>314</td>
</tr>
<tr>
<td>Report Cache Size</td>
<td>32</td>
</tr>
<tr>
<td>Report Cache Used</td>
<td>33</td>
</tr>
<tr>
<td>Report Task Stop</td>
<td>68</td>
</tr>
<tr>
<td>Representation Libraries</td>
<td>34, 47</td>
</tr>
<tr>
<td>Request to Send</td>
<td>182</td>
</tr>
<tr>
<td>Reset Event Queues</td>
<td>304, 328</td>
</tr>
<tr>
<td>Response Timeout</td>
<td>212</td>
</tr>
<tr>
<td>Retry Limit</td>
<td>188</td>
</tr>
<tr>
<td>Revision Compatibility</td>
<td>69</td>
</tr>
<tr>
<td>Revision compatibility switch</td>
<td>71</td>
</tr>
<tr>
<td>Revision Date</td>
<td>25</td>
</tr>
<tr>
<td>Revision of the Product</td>
<td>24</td>
</tr>
<tr>
<td>Revision State</td>
<td>25</td>
</tr>
<tr>
<td>REVISION_COMPATIBILITY</td>
<td>69</td>
</tr>
<tr>
<td>REX</td>
<td>107, 151</td>
</tr>
<tr>
<td>REX Stations</td>
<td>154, 267</td>
</tr>
<tr>
<td>RI</td>
<td>187</td>
</tr>
<tr>
<td>RK</td>
<td>187, 188</td>
</tr>
<tr>
<td>RL</td>
<td>34, 47, 188, 251</td>
</tr>
<tr>
<td>RM</td>
<td>164, 274, 311, 327</td>
</tr>
<tr>
<td>RM00</td>
<td>93</td>
</tr>
<tr>
<td>RM01</td>
<td>93</td>
</tr>
<tr>
<td>RN</td>
<td>99, 100</td>
</tr>
</tbody>
</table>
## MicroSCADA Pro

### System Objects

#### Technical Description

<table>
<thead>
<tr>
<th>Object</th>
<th>Page References</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>63</td>
</tr>
<tr>
<td>Router LMK</td>
<td>251</td>
</tr>
<tr>
<td>Routing</td>
<td>27</td>
</tr>
<tr>
<td>Routing Node</td>
<td>99</td>
</tr>
<tr>
<td>RP</td>
<td>24, 196</td>
</tr>
<tr>
<td>RP570 slave protocol</td>
<td>172</td>
</tr>
<tr>
<td>RP570/571 master protocol</td>
<td>172</td>
</tr>
<tr>
<td>RQ</td>
<td>270</td>
</tr>
<tr>
<td>RS</td>
<td>25, 68, 163, 304, 328</td>
</tr>
<tr>
<td>RT</td>
<td>101, 154, 226, 245, 253, 274, 278, 293, 309, 316, 328</td>
</tr>
<tr>
<td>RTS</td>
<td>182</td>
</tr>
<tr>
<td>RTS Keep Up Padding Characters</td>
<td>187</td>
</tr>
<tr>
<td>RTU</td>
<td>107, 151, 154, 240</td>
</tr>
<tr>
<td>RTU Configuration Attributes</td>
<td>242</td>
</tr>
<tr>
<td>RTU Loop Status</td>
<td>249</td>
</tr>
<tr>
<td>RU</td>
<td>33, 297, 315</td>
</tr>
<tr>
<td>Running Mode</td>
<td>164, 274, 311, 327</td>
</tr>
<tr>
<td>Running Objects</td>
<td>63</td>
</tr>
<tr>
<td>Running Status</td>
<td>163</td>
</tr>
<tr>
<td>RW</td>
<td>200, 314</td>
</tr>
<tr>
<td>RX</td>
<td>154</td>
</tr>
</tbody>
</table>

### S

| S.P.I.D.E.R. RTU | 249 |
| S.P.I.D.E.R. RTUs | 240 |
| S.P.I.D.E.R. SCADA Master Stations | 155 |
| SA | 23, 100, 142, 149, 223, 241, 251, 281, 300, 316, 328 |
| SAT | 107, 151 |
| SATTCON COMLI Stations | 155 |
| SB | 182 |
| SC | 53, 92, 245, 268 |
| SCIL | 12, 14 |
| Screen Size | 84 |
| SD | 40, 54, 84, 93, 116, 175, 304, 328 |
| SE | 145, 148, 222 |
| Semigraphic | 84 |
| Serial lines | 169 |
| Session Idle Timeout | 268 |
| Session in Sequence Response Delay | 268 |
| Session Keepalive Timeout | 269 |
| Session Nack Timeout | 268 |
| Session Retransmit Timeout | 269 |
| Session Setup Handling | 268 |
| SET | 137 |
| SETTING_LA_AND_AG_DOES_NOT_ALARM | 70 |
| SF | 54, 211 |
| SG | 84, 188 |
### MicroSCADA Pro

#### System Objects

**Technical Description**

<table>
<thead>
<tr>
<th>Object</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH</td>
<td>34, 164, 268</td>
</tr>
<tr>
<td>Shadow Dump Slowdown</td>
<td>55</td>
</tr>
<tr>
<td>Shadow the Event Channel Queue</td>
<td>56</td>
</tr>
<tr>
<td>Shadowing</td>
<td>34, 53</td>
</tr>
<tr>
<td>Shadowing Connection time</td>
<td>53</td>
</tr>
<tr>
<td>Shadowing Diagnostic Interval</td>
<td>55</td>
</tr>
<tr>
<td>Shadowing Flush time</td>
<td>54</td>
</tr>
<tr>
<td>Shadowing Number</td>
<td>55</td>
</tr>
<tr>
<td>Shadowing Phase</td>
<td>56</td>
</tr>
<tr>
<td>Shadowing Receive Timeout</td>
<td>57</td>
</tr>
<tr>
<td>Shadowing State</td>
<td>57, 164</td>
</tr>
<tr>
<td>Shadowing Watchdog</td>
<td>58</td>
</tr>
<tr>
<td>SHARED</td>
<td>66</td>
</tr>
<tr>
<td>SI</td>
<td>55, 155, 268, 321</td>
</tr>
<tr>
<td>SID</td>
<td>107, 151</td>
</tr>
<tr>
<td>SIN</td>
<td>107, 151</td>
</tr>
<tr>
<td>SINDAC ADLP180 Stations</td>
<td>155</td>
</tr>
<tr>
<td>SINDAC ADLP80 Stations</td>
<td>155</td>
</tr>
<tr>
<td>Single Char Response</td>
<td>314</td>
</tr>
<tr>
<td>Single Indication</td>
<td>321</td>
</tr>
<tr>
<td>SK</td>
<td>269</td>
</tr>
<tr>
<td>SL</td>
<td>55, 238, 309</td>
</tr>
<tr>
<td>Slave Address</td>
<td>328</td>
</tr>
<tr>
<td>Slave Mode</td>
<td>298</td>
</tr>
<tr>
<td>SM</td>
<td>155, 246, 258, 270</td>
</tr>
<tr>
<td>SN</td>
<td>55, 155, 267, 276</td>
</tr>
<tr>
<td>SO</td>
<td>155</td>
</tr>
<tr>
<td>SP</td>
<td>40, 56, 156, 239, 256, 274</td>
</tr>
<tr>
<td>SPA</td>
<td>107, 151, 156</td>
</tr>
<tr>
<td>SPA Message</td>
<td>258, 270</td>
</tr>
<tr>
<td>SPA Point</td>
<td>256, 274</td>
</tr>
<tr>
<td>SPA point</td>
<td>255</td>
</tr>
<tr>
<td>SPA point definitions</td>
<td>254</td>
</tr>
<tr>
<td>SPA points</td>
<td>274</td>
</tr>
<tr>
<td>SPA protocol</td>
<td>172</td>
</tr>
<tr>
<td>SPACOM</td>
<td>40, 156, 250</td>
</tr>
<tr>
<td>SPACOM Driver Name</td>
<td>40</td>
</tr>
<tr>
<td>SPACOM Protocol</td>
<td>40</td>
</tr>
<tr>
<td>SPACOM STA Attributes</td>
<td>258</td>
</tr>
<tr>
<td>SPI</td>
<td>151, 281</td>
</tr>
<tr>
<td>Split</td>
<td>231, 237</td>
</tr>
<tr>
<td>Split Destination List</td>
<td>238</td>
</tr>
<tr>
<td>Spontaneous Data</td>
<td>304, 328</td>
</tr>
<tr>
<td>Spontaneous message</td>
<td>218, 237</td>
</tr>
<tr>
<td>SQ</td>
<td>56</td>
</tr>
<tr>
<td>SR</td>
<td>57, 200, 269, 314</td>
</tr>
<tr>
<td>SS</td>
<td>57, 211, 269</td>
</tr>
<tr>
<td>Object</td>
<td>References</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>ST</td>
<td>52, 107, 111, 123, 156, 223, 259, 294, 309, 329</td>
</tr>
<tr>
<td>STA</td>
<td>15, 17, 26, 103, 134, 136, 151, 152, 156, 215, 217</td>
</tr>
<tr>
<td>STA Attributes</td>
<td>240, 250, 267, 297</td>
</tr>
<tr>
<td>ANSI</td>
<td>223</td>
</tr>
<tr>
<td>STA Objects</td>
<td>215</td>
</tr>
<tr>
<td>Start Address</td>
<td>231</td>
</tr>
<tr>
<td>Start Command</td>
<td>92</td>
</tr>
<tr>
<td>State</td>
<td>259</td>
</tr>
<tr>
<td>Station</td>
<td>103, 133, 134, 136, 151</td>
</tr>
<tr>
<td>Station Address</td>
<td>23, 100, 142, 148, 241, 251, 281, 316</td>
</tr>
<tr>
<td>ANSI</td>
<td>223</td>
</tr>
<tr>
<td>Station address</td>
<td>143, 300</td>
</tr>
<tr>
<td>station address</td>
<td>281</td>
</tr>
<tr>
<td>Station Address Length</td>
<td>309</td>
</tr>
<tr>
<td>Station Address Table</td>
<td>148</td>
</tr>
<tr>
<td>Station Communication Parameters</td>
<td>230</td>
</tr>
<tr>
<td>Station Mapping</td>
<td>52</td>
</tr>
<tr>
<td>Station mapping</td>
<td>51</td>
</tr>
<tr>
<td>Station number</td>
<td>53</td>
</tr>
<tr>
<td>Station Suspension</td>
<td>225</td>
</tr>
<tr>
<td>Station suspension</td>
<td>253</td>
</tr>
<tr>
<td>Station Type</td>
<td>107</td>
</tr>
<tr>
<td>ANSI</td>
<td>223</td>
</tr>
<tr>
<td>Station Types</td>
<td>216</td>
</tr>
<tr>
<td>Status Check Request</td>
<td>245</td>
</tr>
<tr>
<td>Status code</td>
<td>229</td>
</tr>
<tr>
<td>Stop Bits</td>
<td>182</td>
</tr>
<tr>
<td>STY</td>
<td>15, 18, 109</td>
</tr>
<tr>
<td>SU</td>
<td>159, 160, 226, 315</td>
</tr>
<tr>
<td>Subnet Number</td>
<td>267, 276</td>
</tr>
<tr>
<td>Substation</td>
<td>156</td>
</tr>
<tr>
<td>Summer Time</td>
<td>315</td>
</tr>
<tr>
<td>Supervision Configuration</td>
<td>46</td>
</tr>
<tr>
<td>Suspended station</td>
<td>194</td>
</tr>
<tr>
<td>Suspension</td>
<td>225</td>
</tr>
<tr>
<td>Suspension Attribute</td>
<td>253</td>
</tr>
<tr>
<td>Suspension Time</td>
<td>226</td>
</tr>
<tr>
<td>SV</td>
<td>42, 79, 87, 123</td>
</tr>
<tr>
<td>SW</td>
<td>58, 159, 161</td>
</tr>
<tr>
<td>SX</td>
<td>143</td>
</tr>
<tr>
<td>SY</td>
<td>58, 150, 240, 244, 305, 329</td>
</tr>
<tr>
<td>Sync Format</td>
<td>211</td>
</tr>
<tr>
<td>Sync Mode</td>
<td>246</td>
</tr>
<tr>
<td>Sync Status</td>
<td>211</td>
</tr>
<tr>
<td>Synchronise</td>
<td>244, 305, 329</td>
</tr>
<tr>
<td>SYS</td>
<td>15, 19, 21</td>
</tr>
<tr>
<td>SYS Command Multiplier</td>
<td>293</td>
</tr>
</tbody>
</table>
## Technical Description

**SYS Reply Timeout** ................................................................. 293
**SYS Waiting Time** .............................................................. 329
**Sys Waiting Time** ................................................................. 309
**SYS_BASCON.COM** ............................................................... 19, 21
**SYS_EVENT** ............................................................................. 64
**System Device Name** ............................................................. 84, 93, 116
**System Device name** .............................................................. 175
**System Event Enabled** ........................................................... 64
**System message** ................................................................. 145, 146, 219, 220, 334
**System Messages Enabled** .................................................... 148, 222
**System object** ...................................................................... 11, 133, 134, 152
**System object notation** ......................................................... 13
**System object number** .......................................................... 134
**System Messages Enabled** ......................................................
**System Variables** ................................................................... 42, 79, 87, 123
**SZ** ........................................................................................... 84

### T

- **TA** .......................................................................................... 246
- **TB** .......................................................................................... 108
- **TC** .......................................................................................... 310
- **TD** .......................................................................................... 47, 182, 247
- **TE** .......................................................................................... 248

**Terminal Event** ...................................................................... 248
**Terminal Event Message** ........................................................ 295
**Terminal event message** ........................................................ 295
**Terminal Message** .................................................................. 248
**Terminal Status** ..................................................................... 248, 294
**Terminal status messages** ...................................................... 294
**Text Databases** ...................................................................... 47
**TF** ........................................................................................... 27
**TI** ............................................................................................. 27, 94, 186, 188, 189, 212, 230, 294

**Time** ....................................................................................... 163, 166
**Time Bias** .............................................................................. 108
**Time Format** .......................................................................... 27
**Time Initialised** ..................................................................... 294
**Time Master** ........................................................................... 27
**Time Reference** ...................................................................... 28, 108
**Time Season** .......................................................................... 29
**Time Stamp** ........................................................................... 237
**Time Synchronisation** ............................................................ 310
**Time synchronisation** ............................................................ 240
**Time Synchronisation Interval** ............................................... 58
**Time Zone** .............................................................................. 29
**Time Zone Correction** ............................................................ 167
**TIMEOUT** ............................................................................. 27
**Timeout** .................................................................................. 101
**Timeout between Retries** ...................................................... 189
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeout Length</td>
<td>27, 94, 188</td>
</tr>
<tr>
<td>TL</td>
<td>143</td>
</tr>
<tr>
<td>TM</td>
<td>27, 163, 166, 248</td>
</tr>
<tr>
<td>TN</td>
<td>48, 108, 116</td>
</tr>
<tr>
<td>Topic Configuration</td>
<td>317</td>
</tr>
<tr>
<td>Topics</td>
<td>315</td>
</tr>
<tr>
<td>TP</td>
<td>317</td>
</tr>
<tr>
<td>TQ</td>
<td>270</td>
</tr>
<tr>
<td>TR</td>
<td>28, 92, 108, 291</td>
</tr>
<tr>
<td>Transfer Address</td>
<td>246</td>
</tr>
<tr>
<td>Translated Object Number</td>
<td>48, 108, 116</td>
</tr>
<tr>
<td>Translation Limit</td>
<td>143</td>
</tr>
<tr>
<td>Translation Type</td>
<td>45, 82, 89, 109, 115</td>
</tr>
<tr>
<td>Transmission rate</td>
<td>179</td>
</tr>
<tr>
<td>Transmission Wait Delay</td>
<td>190</td>
</tr>
<tr>
<td>Transmit Quota</td>
<td>270</td>
</tr>
<tr>
<td>Transmitter Data Bit Count</td>
<td>182</td>
</tr>
<tr>
<td>TRANSPARENT</td>
<td>115</td>
</tr>
<tr>
<td>Transparent Data</td>
<td>247, 306</td>
</tr>
<tr>
<td>Transparent data</td>
<td>245</td>
</tr>
<tr>
<td>Transparent Data Address</td>
<td>290</td>
</tr>
<tr>
<td>Transparent Data Response</td>
<td>291</td>
</tr>
<tr>
<td>Transport</td>
<td>92</td>
</tr>
<tr>
<td>Transport Layer Timeout</td>
<td>329</td>
</tr>
<tr>
<td>TS</td>
<td>29, 237, 248</td>
</tr>
<tr>
<td>TT</td>
<td>45, 82, 89, 109, 115, 329</td>
</tr>
<tr>
<td>TV</td>
<td>295</td>
</tr>
<tr>
<td>TW</td>
<td>190</td>
</tr>
<tr>
<td>Type</td>
<td>14</td>
</tr>
<tr>
<td>Type Name</td>
<td>112</td>
</tr>
<tr>
<td>TZ</td>
<td>29, 167</td>
</tr>
<tr>
<td>U</td>
<td></td>
</tr>
<tr>
<td>UC</td>
<td>68</td>
</tr>
<tr>
<td>UN</td>
<td>251, 267</td>
</tr>
<tr>
<td>Unacknowledged Alarm Count</td>
<td>68</td>
</tr>
<tr>
<td>Unit Number</td>
<td>251, 267</td>
</tr>
<tr>
<td>Unit Type</td>
<td>251, 267, 276</td>
</tr>
<tr>
<td>UP</td>
<td>259</td>
</tr>
<tr>
<td>Update Points</td>
<td>259</td>
</tr>
<tr>
<td>User Variables</td>
<td>42, 79, 88, 123</td>
</tr>
<tr>
<td>UT</td>
<td>251, 267, 276</td>
</tr>
<tr>
<td>UV</td>
<td>42, 79, 88, 123</td>
</tr>
<tr>
<td>V</td>
<td></td>
</tr>
<tr>
<td>VE</td>
<td>144</td>
</tr>
<tr>
<td>Vector parameters</td>
<td>303</td>
</tr>
</tbody>
</table>
MicroSCADA Pro
System Objects

Technical Description

ADDR ................................................................. 303
COT ................................................................. 303
PRI ................................................................. 303
QU ................................................................. 303
RM ................................................................. 303
RT ................................................................. 303
TYPE .............................................................. 303
VAL ................................................................. 303
Value range ..................................................... 303
Virtual printer .................................................. 120
VS ................................................................. 82

W
WARM ............................................................ 44, 45
Watchdog ......................................................... 37
Watchdog application ....................................... 58
WC ............................................................... 87
Window Color .................................................. 87
Write-only ....................................................... 20, 139
Write-only, not preconfigurable ....................... 140

X
X ................................................................. 82
XA ............................................................... 206
XC ............................................................... 165
XR ............................................................... 212
XS ............................................................... 165
XT ............................................................... 165