1MRS751264-MEN
Issue date: 29.02.00
Program revision: 3.0
Documentation version: A

COM 500
COM 500 Engineering

Notice 1

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

Notice 2

This document version complies with the program revision 3.0.

Notice 3

Additional information such as Release Notes and Last Minute Remarks can be found on the program distribution media.

Trademarks

Related MicroSCADA Technology and COM 500 Manuals

The following MicroSCADA technology manuals are published with this software release:

Connecting LONWORKS Devices to MicroSCADA	1MRS751249-MEN
System Configuration	1MRS751248-MEN
System Objects	1MRS751252-MEN
Application Objects	1MRS751253-MEN
Programming Language SCIL	1MRS751250-MEN
Status Codes	1MRS751251-MEN

The following COM 500 manuals are published with this software release:

COM 500 Engineering	1MRS751264-MEN
Configuring MicroSCADA for IEC 60870-5-101 Master Protocol	1MRS751260-MEN
Configuring MicroSCADA for DNP V3.00 Slave Protocol	1MRS751703-MEN
Configuring MicroSCADA for DNP V3.00 Master Protocol	1MRS751456-MEN
Configuring MicroSCADA for IEC 60870-5-101 Slave Protocol	1MRS751702-MEN

The following COM 500 manuals are delivered only on request (in Word 97 document format):

Communication Programming Interface (CPI)	1MRS751365-MEN
Configuring MicroSCADA for Modbus Slave Protocol	1MRS751263-MEN

Contents

Page

1 Introduction

About this Chapter

This chapter provides some general information about COM 500.

COM 500 3.0 Manuals

This manual is a part of the COM 500 3.0 User Manuals. In addition to this manual, there are 6 other manuals published with COM 500 revision 3.0.

Using This Manual

This manual is the one, which should be read when you want to use the COM 500 product to make a MicroSCADA-based communication gateway. This manual describes how to install, configure and engineer COM 500. Detailed descriptions are given to help the user to provide deeper knowledge about the functionality of COM 500.

Referenced Manuals

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

- System Configuration manual.
- System Objects manual.
- Application Objects manual.

The following COM 500 manuals provide more detailed information about communication protocols:

- Configuring MicroSCADA for DNP V3.00 Slave Protocol manual.
- Configuring MicroSCADA for DNP V3.00 Master Protocol manual.
- Configuring MicroSCADA for IEC 60870-5-101 Slave Protocol manual.
- Configuring MicroSCADA for IEC 60870-5-101 Master Protocol manual.

COM 500 as Communication Gateway

COM 500 is an ABB software product. It is a communication gateway running on MicroSCADA platform. The purpose of COM 500 is to provide a gateway between process devices and up to four upper level systems (NCCs). The main tasks of COM 500 are signal rerouting and protocol conversions. Figure 1 illustrates the use of COM 500 as a gateway in a substation.

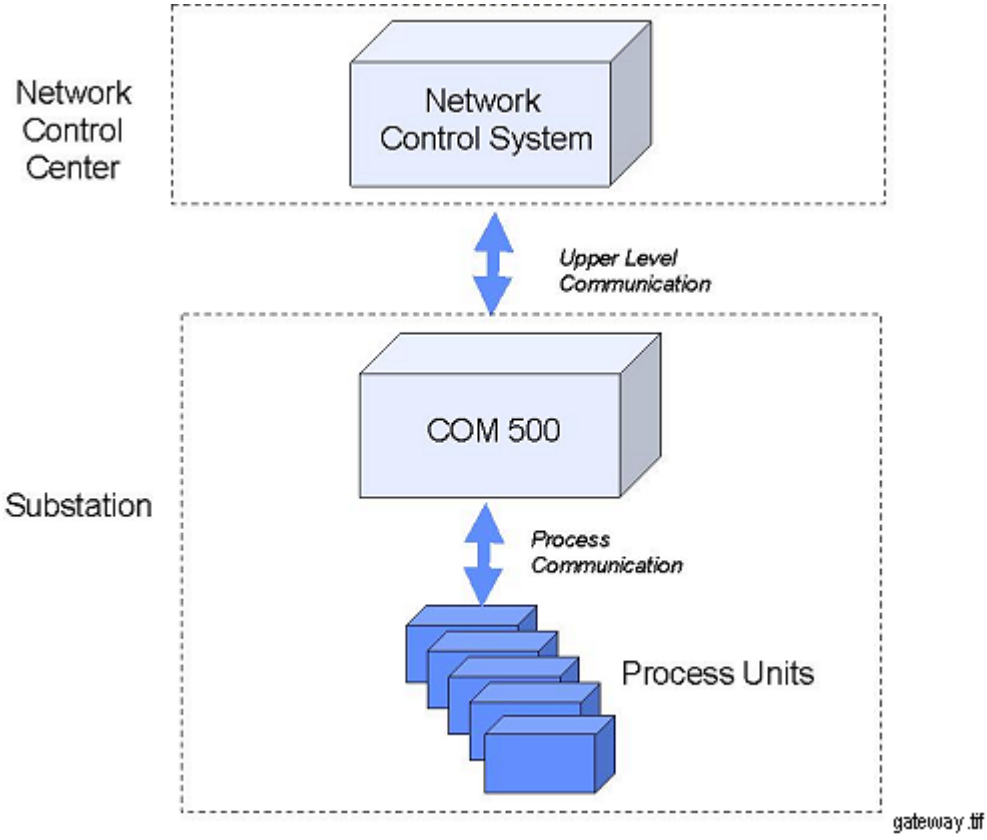

Figure 1. COM 500 as a communication gateway

COM 500 provides a variety of protocols for both process communication and upper level communication. COM 500 functionality can be combined to a SYS 500 system server with a substation-level human-machine interface.

COM 500 Engineering Process

The engineering process of COM 500 contains the following steps:

1 Software installation.

2 System configuration.

3 Signal generation.

4 Signal engineering.

Further details about the engineering steps are given in Chapter 3 of this document.

Signal Engineering Process

Signal engineering in COM 500 Signal engineering contains the following steps:

1 Add the NCCs to the tool and define information related to it, for example the protocol to be used. Define also the alarm groups you want to use.

2 Check that all the indications and commands (i.e. input and output process objects), that are needed are shown in the tool. If they are not, add them.

3 Check that all the necessary attributes for indications and commands are shown in the tool correctly. If they are not, add the missing attributes or change their definitions.

4 Define to which NCCs COM 500 should send the indications. Connect the indications to alarm groups, if needed.

5 Configure the commands to be received from the NCCs.

6 Adjust the system and application parameters.

2 Safety Information

About this Chapter

This chapter gives information about prevention of hazards.

2.1 Back Up Copies

Taking Back Up Copies

Taking back up copies is suggested before making any changes, especially changes that might have side effects. Software and data need to be copied to another place, usually to a back up tape. A DAT tape is commonly used.

Back up copy facilitates restoration of the application software in case of disk crash or other severe failure when stored data is lost. It is therefore recommended that back up copies are taken regularly.

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

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

System Back Up

Usually a system back up is taken after the application is made. It should be taken again when changes are made to MicroSCADA system. For example this is needed when driver configuration or network setup is changed.

Application Back Up

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

A fatal error is an error that causes a break-down or a locked situation in the MicroSCADA program execution.

2.2 Fatal Errors

Handling

In case of a fatal error:

1 Write down possible MicroSCADA error messages.

2 Shut down the MicroSCADA main program. If this cannot be done in the MicroSCADA Control Panel, try to end the task in the Task Manager of Windows NT.

3 Shutting down base system computers by switching off the power might damage the files.

4 In Windows NT, the data kept in the main memory at the moment of a fatal error is placed into drwtsn32.log file. It is placed into the system folder, for example Winnt. Analyze and copy the data in it.

5 Restart the system.

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

Status Codes

Error messages in SCIL are called status codes. The list of status codes and short explanations are in the Status Codes manual.

3 Instructions

About this Chapter

This chapter gives instructions on how to install, configure and engineer COM 500.

3.1 COM 500 Software

General

COM 500 is a communication server, which provides communication gateway functions for mapping signals between process devices and up to four upper level systems. The upper level system is called here network control centre (NCC). It is the system to which COM 500 sends information for supervising and controlling the process of the customer and from which COM 500 receives process control commands. COM 500 also handles system co-ordination tasks, such as dynamic assignment of control command authorities and communication supervision.

COM 500 supports a variety of protocols for connecting to upper level systems. It is typically connected to the network by some telecontrol protocol, for example RP 570. Process devices can be connected to it using different protocols, such as SPA and IEC 870-5-103.

COM 500 is based on MicroSCADA technology, like SYS 500. It can be integrated to the SYS 500 for cost savings in compact system solutions. It provides the user interface needed for signal engineering and communication diagnostics.

Function

The main task of COM 500 is to handle data transfer between process devices and up to four network control centres. Data transfer usually involves protocol conversion. Also other tasks as communication supervision and command authority checking are involved.

Both the process devices and the network control system may be a ABB or third party product as long as it implements the interface defined the communication protocol used. Documents such as interoperability lists and device profiles can be used for verifying the compatibility between COM 500 and other systems.

Example System

COM 500 runs on MicroSCADA platform and can communicate via both DCP-NET and PC-NET units. Several communication frontends may be connected to increase the system performance. COM 500 can be used as a stand-alone gateway, in co-operation with SYS 500 or as a combined SYS-COM system. See Figure 2.

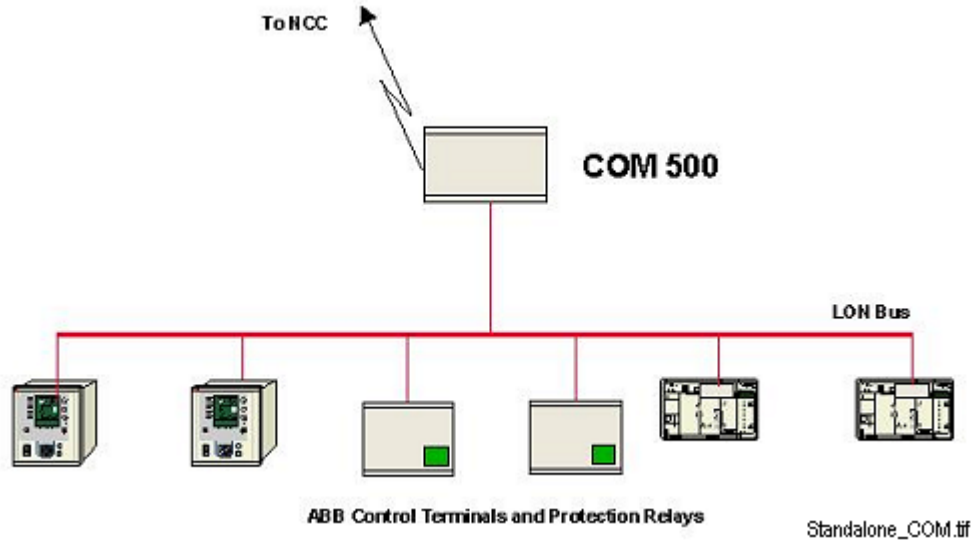

Figure 2. A typical stand-alone COM 500 system

The Role of a COM 500 Application

The purpose of a COM 500 application depends on the type of the system. In case of a stand-alone gateway the application is used merely for signal rerouting, but in case of a combined SYS-COM the application is used also for process control and management, and it acts both as a communication gateway and a substation controlling system (SCS).

This is based on the layered structure of MicroSCADA software presented in Figure 3. The application layer can have different functions independent of the lower layers.

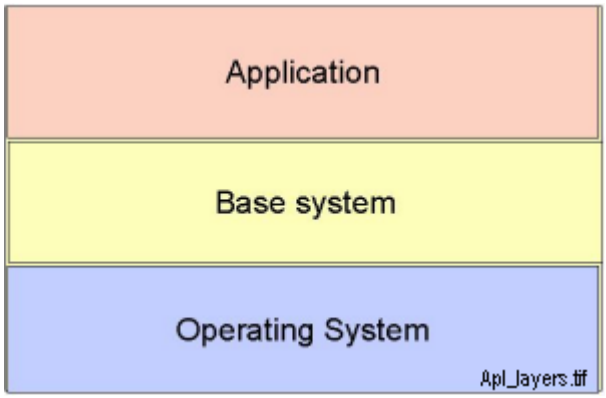

Figure 3. Software layers in COM 500

From the communication point of view COM 500 application sees each process unit and NCC as a system object. Setting the attributes of the system objects can change the properties of the communication channels.

Communication between COM 500 and a NCC is based on command procedures implemented in the SCIL programming language. These procedures send information from the COM 500 application to a NET unit for protocol conversion and

transmission. Typically one command procedure is needed for each type of data. Data can be transferred also based on application and system commands such as general interrogation command, and at special situations such as application start-up and after communication disturbances. When the system is running, process events generally activate the command procedures via event channels.

Commands and setpoints from a NCC to a substation are brought into the COM 500 application via process objects. They activate command procedures via event channels. Command procedures send the actual control commands to the process units. Also system and application commands are received using process objects. The content of the COM 500 command procedures is described in the Chapter 4.

Functional Environment

Functional environment of a COM 500 application can be described using the environment model shown in Figure 4. COM 500 application communicates through the base system and NET unit. In addition to NCCs and COM 500 application, base system can also communicate with a SCS application at the same time.

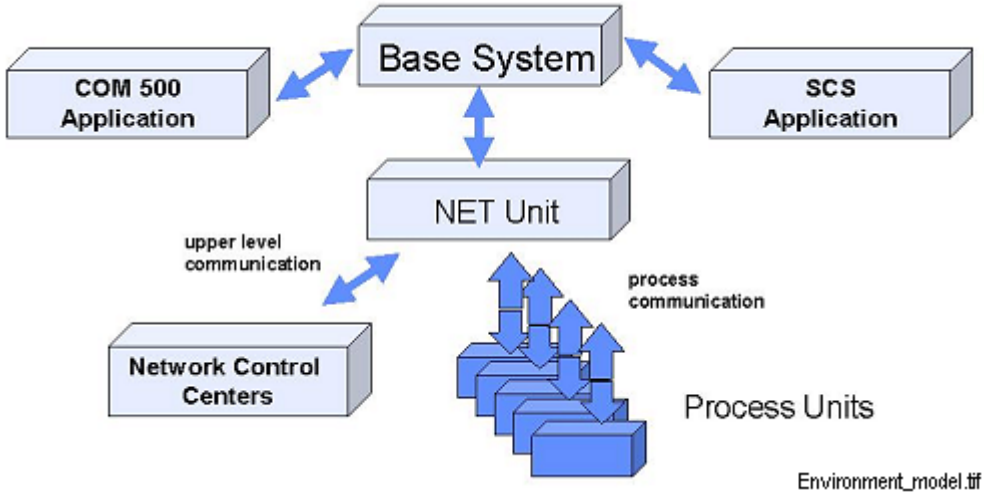

Figure 4. COM 500 environment model

Communication Programming Interface

COM 500 provides support for the Communication Programming Interface (CPI) which is a protocol development environment that can be used for implementing new protocols to MicroSCADA. CPI is a collection of functions programmed in the C language for making communication software that converts between the MicroSCADA internal protocol and other protocols. CPI is available on request.

CPI based communication software can be used for process communication or for upper level communication. The CPI library contains functions to send and receive messages to or from COM 500. It also contains functions to pack and unpack data. The CPI based communication software and COM 500 communicate through TCP/IP network. The program that uses CPI interface in COM 500 must emulate a RTU

device placed into a communication fronted. CPI is described in more details in Chapter 4 of this document.

3.2 Installation

3.2.1 General

Upgrading COM 500 Rev. 1.0...2.0A to 3.0

If you want to upgrade an existing COM application to COM 500 revision 2.0 A, please observe the following notes.

If the previous revision of COM 500 is 1.0..1.0C, you must prepare the application for both Base Tools and COM 500 by using the Control MicroSCADA Application dialog box before starting MicroSCADA. When upgrading from revision 2.0 preparing is not needed.

COM 500 revision 3.0 has a mechanism that makes all the required modifications to the application to update it to revision 3.0. This mechanism is started when a monitor is started to a COM 500 application first time after the installation of COM 500 3.0. After the mechanism is run you must restart MicroSCADA to take all modifications in use. The modifications are described in detail in section 3.2.3.

If any project specific modifications have been made to the command procedures of the previous COM 500 revision, the modifications must be copied to the matching new command procedures. The signal configuration, that is the contents of the cross-reference tables, does not need any changes. The content of command procedures is described in the Chapter 4.

For changes needed in the configuration files see section 3.3.1 for details.

3.2.2 Software Installation

Installation Procedure

1 Install the required software products as follows:

- If the system is used only as a communication gateway only COM 500 software needs to be installed.
- If the system is used also as a substation control system with a local HMI, install also SYS 500.
- If the application is built by using LIB 5xx application libraries, install also LIB 500 and other LIB 5xx products needed.

2 Prepare the application for Base Tools, LIB 500 (if needed) and COM 500 using the Control MicroSCADA Application dialog box. For more information on preparing the application, see MicroSCADA Installation manual.

3.2.3 COM 500 Start-up

Actions at Start-up

COM 500 automatically creates all the necessary application objects (event and time channels, command procedures etc.) when a monitor is opened to a COM 500-prepared application for the first time after the installation of COM 500 software.

COM 500 also creates the directory \sc\apl\'name'\com500 that is used for storing cross-reference and parameter files.

Updating from Revision 1.0…1.0 C

The application objects (event channels, command procedure etc.) of COM 500 revision 3.0 have been renamed not to follow LIB 5xx conventions, as was the case in older COM 500 revisions. Also the cross-reference pointer is stored in a different attribute in revision 3.0.

COM 500 3.0 provides an automatic application update script that goes through the process database of a COM 500 application and does the following changes:

- The cross-reference pointer of a process object is moved from the FI attribute to the new TI attribute.
- Event channel name is changed from BNU_* to COM_*.
- The old COM 500 command procedures BNU_*:C are maintained, the new corresponding command procedures are named COM_*:C.
- The predefined command procedure APL_INIT_1:C is modified to execute the new COM 500 initialisation procedure.

The cross-reference text files BNU_XR*.TXT are also renamed to COM_*.TXT. Indication cross-reference data is moved to the free type process objects for permanent storage. The icons of COM 500 tools in Tool Manager are modified to point to /COM/ACTIVE/COM_.

Updating from Revision 2.0 and 2.0 A

The command procedures are changed from the previous releases as listed below. The modified command procedures will be automatically updated in the application, and as a safety measure the existing command procedures COM_*:C in the application will be copied to the name TMP_*:C. Note that only the command procedures included in the list below are copied.

The following command procedures have changed from revision 2.0:

COM_AUTHCH:C	COM_RDDATA:C
COM_BOCMD:C	COM_RDGEN:C
COM_CNTINT:C	COM_RDXREF:C
COM_COMINI:C	COM_RESPRC:C
COM_CPISS:C	COM_RPSFT:C
COM_DSAO:C	COM_RPSS:C

COM_DSBO:C
COM_DSDO:C
COM_DSXREF:C
COM_GENINT:C
COM_GRPAL:C
COM_GRPSND:C
COM_IECTRM:C
COM_IESA:C
COM_IESS:C
COM_IESEI:C
COM_NETINI:C
COM_SUSSTA:C
COM_USPC:C
COM_USAI:C
COM_USBI:C
COM_USDB:C
COM_USDI:C
COM_USBS:C
COM_USPC:C
COM_USXREF:C

The following command procedures have changed from revision 2.0 A:

COM_BOCMD:C
COM_GENINT:C
COM_IESEI:C
COM_NETINI:C
COM_RDDTA:C
COM_RDGEN:C
COM_RDXREF:C
COM_REVDTA:C
COM_RESPRC:C
COM_RPSFT:C
COM_RPSS:C
COM_SUSSTA:C
COM_USAI:C
COM_USBI:C
COM_USDB:C
COM_USDI:C
COM_USBS:C
COM_USPC:C
COM_USPC:C

The following new command procedures are included in COM 500 revision 3.0:

COM_RPSSY:C

The Signal Cross-reference Tool will automatically add digital input and digital output process objects to the standard view definition. Indication cross-reference data is moved to the free type process objects for permanent storage. The icons of COM 500 tools in Tool Manager are modified to point to /COM/ACTIVE/COM_.

3.2.4 Presentation Order of Pictures and Dialogs

Depending on which software the COM 500 application is prepared for the login dialog shown when opening a monitor and the picture/dialog shown after login is different. The following two cases can be found:

- When the application has been prepared for Base Tools and COM 500, the COM 500 login dialog is shown when a monitor is opened to the application. After login the Communication Diagnostics dialog is opened. Both the Tool Manager and the Signal Cross-reference Tool can be accessed directly from this dialog.
- When the application has been prepared also for LIB 500, the LIB 500 login dialog is used. After login the picture defined in the LIB 500 Application Settings dialog as the fist picture is opened. The Signal Cross-reference Tool and Communication Diagnostics dialog can be opened from the Tool Manager after they have been added to it, see the next two sections 3.5.1 and 3.5.2

3.3 System Configuration

3.3.1 General

Configuration Files

In COM 500 base system is configured with SYS_BASCON.COM. COM 500 revisions 1.0...1.0C provided a specific base system configuration file SYS_COM500.COM.

The SYS_BASCON.COM and SYS_COM 500.COM files are ASCII files, which can be edited with a text editor, for example with Notepad in the Windows NT or with SCIL Program Editor.

Base System Objects

Each base system has a set of objects that specify base system and its environment, hardware and software, as well as the physical and logical connections of the base system and its applications.

Base system objects are defined with SCIL commands in the SYS_BASCON.COM file, which is executed each time the base system is started. With a few limitations, you can also define and modify base system objects any time when COM 500 is running. During the operation, the base system objects are in the primary memory of the base system computer.

Communication System Objects

Each NET unit contains a set of system objects, which specify line properties, connected devices etc. These objects can be created, modified and deleted by SCIL, and setting the attributes of the objects can change the properties. Each communication line is represented by one object as well as each station created on a line.

In case of a DCP-NET units communication system objects are defined by using a so called preconfiguration. During operation, the system objects are in the memory of the DCP board, which is the protocol hardware used.

In case of a PC-NET unit process communication system objects can be defined using the System Configuration Tool or by using SCIL statements.

3.3.2 Base System Configuration

COM 500 Base System

Basic configuration of base system, for example the base system itself, nodes, links and MicroSCADA monitors, are defined in SYS_BASCON.COM. Base system configuration is described in detail in the MicroSCADA System Configuration manual.

SYS_BASCON.COM

The new SYS_BASCON.COM template provides all the necessary definitions for COM 500. If the new template is used the characters -; must be removed from the beginning of the lines shown below in order to activate the definitions for a COM 500 application:

```
-; PQ = 15,-                                  ;Number of parallel queues
-; QD = (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),-    ;Parallel queue dedication
```

If the system has no older revision of COM 500 installed and the old SYS_COM500.COM is wanted to be used, the application definition (without -; in the beginning) must be added to the lines shown above. Also the following station type definitions must be added:

```
#CREATE STY22:B = LIST(NA = "SPI", DB = "STA", CX = "S.P.I.D.E.R/RP570")
#SET STY29:BCX  = "IEC"
#SET STY30:BCX  = "DNP"
```

If the new System Configuration Tool included in the MicroSCADA 8.4.2 A base tools is wanted to be used, the new SYS_BASCON.COM must be used.

SYS_COM500.COM

The definitions made in this file have been added to the new SYS_BASCON.COM template included in MicroSCADA 8.4.2 A so SYS_COM500.COM is no more needed. If, however, the new template is not used, the following correction should be made to SYS_COM500.COM:

```
#SET SYS:BRC = 0
```

should be changed to

```
#SET SYS:BRC = 2000.
```

3.3.3 Communication System Configuration

Protocols

COM 500 revision 3.0 supports several protocols. For example protocols SPA, LONTALK AND IEC 60870-5-101 master can be used for process communication and RP 570 slave and IEC 60870-5-101 slave protocols for upper level communication with network control centres (NCC). Some protocols are implemented in both DCP-NET and PC-NET units whereas some are supported in PC-NET only. For more information about the protocols, see Chapter 4.

The following chapters provide information for the configuration of RP-570 slave, IEC 60870-5-101 slave and DNP V3.00 slave lines and stations needed for

communication with upper level systems. Communication system configuration needed for a CPI application is described in the document CPI Programming Manual.

When the upper level communication is configured it is very important to match the parameters (attributes) of the line and station to the parameters of the upper level system. The attributes of line and station system objects are described in detail in the System Objects Manual.

The communication system configuration needed for the process communication (master protocols) is similar as in SYS 500 and is described in the System Configuration manual.

DCP-NET Unit

DCP-NET program contains a preconfiguration. It contains system objects and attributes that provide the default configuration. Each time DCP-NET unit is loaded and started the preconfiguration becomes valid.

The preconfiguration can be viewed, edited and documented off-line or on-line. Off-line operations are done with NETCONF tool, which runs in the DOS environment. During operation base system is configured using a preconfiguration tool.

The preconfiguration has following limitations:

- A maximum of 20 process units can be preconfigured for a NET unit.
- All attributes cannot be preconfigured.

IEC 60870-5-101 slave protocol or DNP V3.00 slave protocol is not supported in DCP-NET.

The easiest way to configure RP-570 slave line and station for a DCP-NET unit is to use the preconfiguration. An example of DCP-NET preconfiguration line attributes for RP-570 slave protocol is shown in Table 1.

Table 1. DCP-NET preconfiguration for RP-570 slave protocol

Attribute	Description	Value
PO	Protocol	16
IU	In Use	1
MS	Message Application	1
MI	Message Ident.	0
LK	Link Type	0
BR	Baud Rate	9600
SB	Stop Bits	1
PY	Parity	2
RD	Receiver Data Bits	8
TD	Transm. Data Bits	8
OS	Output Synchroniz.	1
RE	Redundancy	2
TI	Timeout Length	3
NA	NAK Limit	3
EN	ENQ Limit	3
DE	CTS Delay Length	40
ER	Embedded Response	0
RP	Reply Poll Count	10
PD	Poll Delay	100
PS	Buffer Pool Size	20
PP	Polling Period	3
CN	Connection	Ign

An example of DCP-NET preconfiguration process unit attributes for RP-570 slave protocol is shown in Table 2.

Table 2. DCP-NET preconfiguration for RP-570 slave protocol

Attribute	Description	Value
	Device Type	SPI
	Physical Device Number	1
LI	Line Number	1
AL	Allocation	1
AS	Allocating Applic.	1
IU	In Use	1
MI	Message Ident.	0
MS	Message Application	1
SA	Station Addr. (Dec)	1
DE	Diagnostics Enable	0
DI	Diagnostic Interval	0
FS	Fast Sel. to Susp.	0
RT	Reply Timeout	20
SP	Message Split	0
SU	Suspension Time	60
	Name	Ign

PC-NET Unit

PC-NET units can be configured either using the System Configuration Tool or with SCIL. Using the System Configuration Tool is preferred when configuring COM 500. Detailed information about using the System Configuration Tool is provided in the System Configuration manual.

In some cases SCIL statements are wanted to be used for making the configuration. COM 500 provides standard command procedures for creating lines and stations for RP 570 slave, IEC 60870-5-101 slave and DNP V3.00 protocols. Command procedure COM_RPSCR contains line and station definitions for RP-570 slave protocol. It needs

to be executed every time PC-NET unit is restarted, for example from the predefined command procedure APL_INIT_1:C.

As input the command procedure COM_RPSCR needs the NET number, line number, vector of stations and message application number.

The following command defines RP 570 slave protocol to line 1 of NET 1 and adds master station with number 8 into it:

```
#EXEC COM_RPSCR:C (@NET=1, @LINE=1, @STATIONS=VECTOR(8), @APPLIC=1)
```

Command procedure COM_101SCR defines lines and stations for IEC 60870-5-101 slave protocol. It also needs to be executed every time the PC-NET unit is started. As input the command procedure needs NET number, line number, communication mode (0 = balanced, 1 = unbalanced), vector of stations and message application number.

The following command defines a balanced IEC 60870-5-101 slave protocol to line 2 of NET 1 and adds a master station with number 9 into it:

```
#EXEC COM_101SCR:C (@NET=1, @LINE=2, @MODE=0, @STATIONS=VECTOR(9),-
@APPLIC=1)
```

For DNP V3.00 slave protocol lines and stations can be created using the command procedure COM_DNPSCR, which also should be executed every time the PC-NET unit is started. As input this command procedure needs NET number, line number, vector of stations and message application number.

The following command defines a DNP V3.00 slave protocol to line 3 of NET 1 and adds a master station with number 10 into it:

```
#EXEC COM_DNPSCR:C (@NET=1, @LINE=2, @MODE=0, @STATIONS=VECTOR(10),-
@APPLIC=1)
```

3.3.4 Configuration Tips

IEC 60870-5-101 Protocol

When configuring IEC 60870-5-101 slave lines and stations it is very important to match the message field length attributes (IL, CL, PL and SL, see system configuration manual for details) to the corresponding parameters of the master system. A mismatch with these attributes can lead to a situation where communication appears to be running properly, but the messages are incorrectly interpreted or not set to the process database at all.

The semantics of the cause of transmission is different in MicroSCADA and in IEC 60870-5-101. Therefore if MicroSCADA is used as the IEC master, the following definition should be made to the station type:

```
#SET STY29:BCT(3) = "UNKNOWN"
```

If this definition is not made, spontaneous events (COT = 3) are not registered in process database.

Application and system commands and unrecognized messages sent from the NCCs are received in process objects and interpreted by COM 500 command procedures. To ensure that these commands are received and executed properly, the following things should be checked:

- The MI attribute of each slave station should have its default value (29000 + station number).
- The CA attribute of each IEC 60870-5-101 slave station should have its default value (32000).

The value of the PC (Process Data Confirmation) attribute of each IEC slave station should be set to 0 in order to ensure proper confirmation and termination of IEC commands.

If COM 500 is not synchronized from the NCC, the value of the RM (Running Mode) attribute of the IEC slave station should be so that bit 1 of the value is set (RM = 2 if no other bits are set).

DNP V3.00 Protocol

Some requests and unknown messages sent from the NCC are received in process objects and interpreted by COM 500 command procedures. To ensure that these messages are received and interpreted properly, the following things should be checked:

- The MI attribute of each slave station should have its default value (30000 + station number).
- The CA attribute of each IEC 60870-5-101 slave station should have its default value (32000).

When data is send as a double binary signal, two binary data objects with consecutive addresses are used for the two bits of the value.

3.4 Signal Generation

The signals i.e. process objects for COM 500 need to be generated somehow. The signals can be generated by:

- Re-using an existing application or parts of an existing application.
- Importing signals.
- Building a COM 500 application.

3.4.1 Re-using an Existing Application

Re-using an existing application is recommended especially when SYS 500 and COM 500 are used in parallel. After the SYS 500 application has been built it (directory \sc\apl\'name') can be copied as such to the COM 500 computer.

If only gateway functionality is required (no MMI, printouts etc.) it is enough to copy only the directory \sc\apl\'name'\apl_ that contains the necessary databases. If an

existing COM 500 application is re-used, for example in another substation, also the directory \sc\apl\'name'\COM500 must be copied.

After copying the application it must be prepared for COM 500 in the COM 500 computer. Make also the required changes in SYS_BASCON.COM.

When re-using a complete application that has been prepared for LIB 500 you must do one of the following:

- Install LIB 500 to the COM 500 computer.
- Rename the file \sc\apl\'name'\apl_\apl_lib.txt.

3.4.2 Importing Signals

Importing signals means that process objects are imported from another application using load files and tools made for this purpose. Importing process objects can take place in the following ways:

- By using the MicroSCADA DB <--> text tool in both source and target (COM 500) applications.
- By using the LIB 500 Database Import/Export Tool in both source and target applications.
- By using the LIB 500 Database Import/Export Tool to import a load file generated by the SigTOOL software.

When using the LIB 500 Database Import/Export Tool the required scale objects must be created manually in the target application.

3.4.3 Building a COM 500 Application

Building a COM 500 application (creating the process database) is similar as building a SYS 500 application, the actual gateway functionality is provided by the COM 500 software and the definitions made using the Signal Cross-reference Tool in the signal engineering phase. A COM 500 application can be built in the following ways:

- Manually by creating process object one by one.
- Using LIB 5xx application library software.

Building a COM 500 application manually is quite troublesome and comes into question only in case the number of signals is very small. Process objects can be created by the Object Navigator of the base tools, using the Signal Cross-reference Tool or by SCIL.

Building the application using LIB 5xx application software is strongly recommended even if no MMI is required since LIB 5xx provides an effective graphical engineering environment where the process database can be built fast and easily. LIB 5xx provides

support for most types of process devices, and the LIB 5xx picture functions create the correct number of process objects with suitable attribute values.

For building a COM 500 application using LIB 5xx software the following software must be installed:

- LIB 500 application library base software.
- Any LIB 5xx software containing the needed functionality.

To create the application:

1 Create pictures for the application. These pictures consist of background and picture functions for controlling and supervising the process units. Use Picture Editor for creating the pictures. In addition to supporting creation of a background, it provides tools for handling picture functions. If the COM 500 application is to be used only as a gateway (no MMI) the pictures can be created as drafts, i.e. picture editing should not be given much effort.

2 Install the necessary standard functions from application libraries. The picture function is simply given an identification and placed into the picture.

3 Configure the picture function. Set the parameters of the picture function according to the configuration of the process unit and the application. Use Standard Configuration Tool for this purpose. For more information related to these three first steps, see Chapter 7 in the manual Picture Editing.

Each picture function creates all application objects it requires. These application objects are usually created at the configuration phase. In addition to Standard Configuration Tool, Process Object Definition Tool is used to define process objects.

Application Check List

Following things must be noted when building and maintaining COM 500 application:

- Table Index (TI) attribute of the process objects has been reserved for COM 500 cross-reference purposes. Changing any TI value in the application may result in a severe malfunction.
- Free Integer (FI) attribute of those output process objects included in a command in the Signal Cross-reference Tool is used for database queries. This attribute should not be changed.
- COM 500 uses event channels named COM_* to activate command procedures. These must not be removed from the process objects. If any other event channel is to be attached to a process object, the corresponding command procedure (command procedure name = event channel name) must be attached as a secondary object of this event channel.
- Application objects with logical name COM_* or BNCC* are internal COM 500 objects, and must not be modified.
- APL:BSV elements 20..29 are reserved for COM 500.
- Station local/remote switch must be in “remote” position to enable commands from any NCC.

- Names of the NCCs to perform commands must be in the “Authorized Centres” list of the Station Authority dialog (accessed via the Stations menu of a LIB 5xx station picture).

3.5 Signal Engineering

Signals

Signals are divided into indications and commands, i.e. input and output process objects. Indications are sent from process units to COM 500 where they are re-routed to one or several NCCs. Typically there are double indications and measurements that need to be forwarded to NCCs.

Commands are sent from NCC to COM 500 where they are re-routed to process units. Secured commands, direct commands and setpoints are typical commands that are sent. Re-routing of indications and commands is presented in Figure 5.

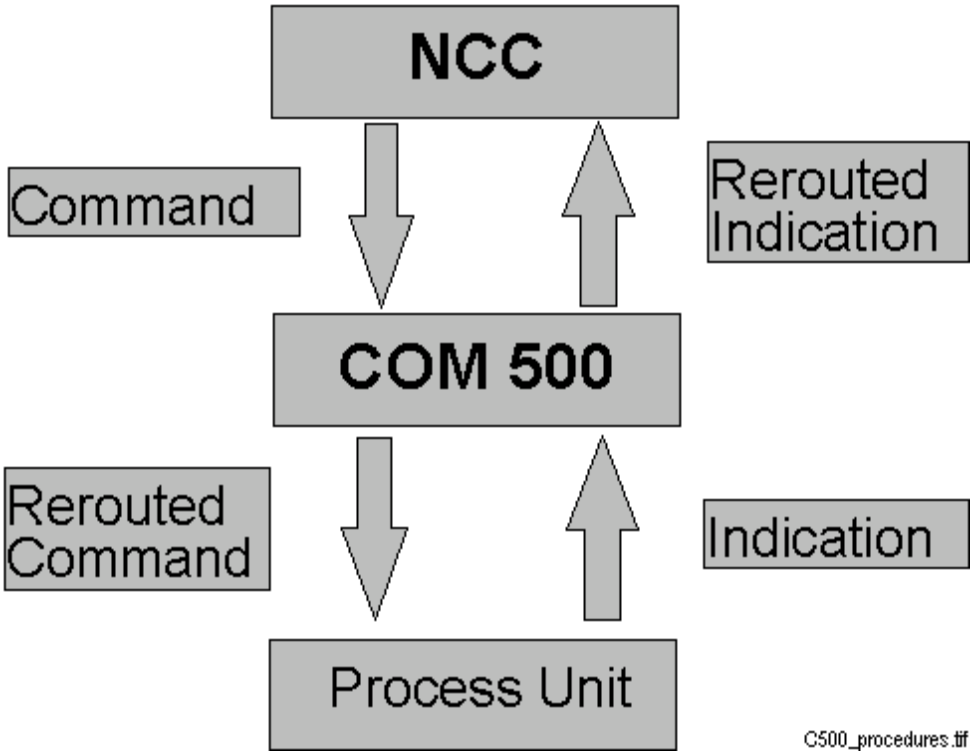

Figure 5. Signals are sent from process units to NCC and commands from NCC to process unit. COM 500 reroutes the signals.

Signal Engineering Process

Signal Engineering means that COM 500 is told how to reroute the signals. This isdone using the Signal Cross-reference Tool. Signal engineering contains the following steps:

1 Add the NCCs to the tool and define information related to it, for example the protocol that should be used. Define also the alarm groups you want to use. This step is described in sections 3.5.6 and 3.5.7

2 Check that all the indications and commands that are needed are shown in the tool. If they are not, add them. This step is described in the section 3.5.12.

3 Check that all the necessary attributes for indications and commands are shown in the tool correctly. If they are not, add the missing attributes or change their definitions. This step is described in the section 3.5.9.

4 Define where COM 500 should send the indications, either the address or alarm group. This step is described in the section 3.5.13.

5 Define the address to which COM 500 should send the commands. If you need to send the command in a specific form or you need a reply for the command, define those also. This step is described in the section 3.5.14.

Signal Engineering is done in the Signal Cross-reference Tool, which is described in the following sections.

3.5.1 Adding Icon for the Signal Cross-reference Tool

To add icon for the Signal Cross-reference Tool to the Tool Manager:

1 Select the System Configuration page.

2 Choose **Insert Tool** from the Edit menu.

3 Choose **Signal X-references** from the list of tools. Tool Properties dialog appears.

4 Click **OK**. The Signal Cross-reference Tool is added to Tool Manager.

3.5.2 Adding Icon for the Communication Diagnostics Dialog

To add icon for the Communication Diagnostics dialog to the Tool Manager:

1 Select the System Configuration page.

2 Choose **Insert Tool** from the **Edit** menu.

3 Choose **Signal COM 500 Diagnostics** from the list of tools. Tool Properties dialog appears.

4 Click **OK**. The Communication Diagnostics dialog is added to Tool Manager.

3.5.3 Using the Signal Cross-reference Tool

Signal Cross-reference Tool is the tool that is used for mapping the signals to and from the process devices to the signals to and from the NCCs. Signal Cross-reference Tool can also be used for making NCC and alarm group definitions and setting the system and application parameters.

3.5.4 Parts of the Signal Cross Reference Tool

General

The Signal Cross-reference Tool is shown in Figure 6. It contains a menu bar at the top with six menus, which are **Cross-reference** menu, **Edit** menu, **Signal** menu, **View** menu, **Settings** menu and **Help** menu.

Below the menu bar there is a toolbar with twelve short-cut keys, a combo pop-down menu for the views and a short-cut key for opening the view definitions dialog. The Signal cross-reference Tool contains four notebook pages, which are **Indications** page, **Commands** page, **NCCs** page and **Parameters** page.

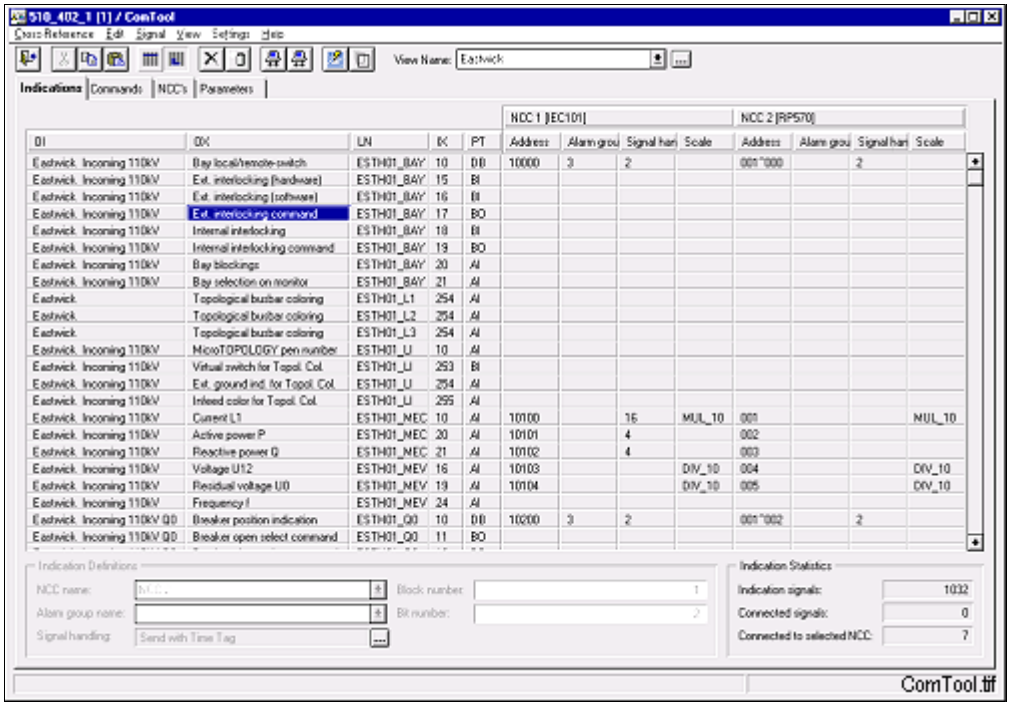

Figure 6. Signal cross-reference Tool, which contains menu bar, toolbar and four different pages for signal engineering.

Cross-reference Menu

Cross-reference menu is used for opening the Cross-reference Import and Export dialogs, the Print and Page Setup, and Print dialogs. If you choose **Exit**, Signal Cross-reference Tool is closed.

Edit Menu

Edit menu contains functions for cutting, copying and pasting text between text boxes located in pages and dialogs of the Signal Cross-reference Tool. It is also used for opening the Find dialog that can be used for finding text strings from the signals and the cross-reference data.

Signals Menu

By using the **Signals** menu, signals (process objects) can be added and edited. The scale of analog input process objects can be edited. **Signals** menu is also used for opening the Attribute Definitions dialog. The Attribute Definitions dialog is used for defining and ordering attributes to be shown in Indications and Commands pages.

View Menu

You can use the **View** menu to change the view in the Signal Cross-reference Tool. When you choose a view name from the **View** menu, the signals are read from the process database both to indications and commands page. The **View** menu can also be used for opening the View Definitions dialog. The View Definitions dialog is used to define search conditions for signals displayed in **Indications** and **Commands** pages. The signals are searched from the process database.

Settings Menu

The **Settings** menu is used for setting the toolbar visible and invisible and for choosing the select method to be used when editing signals or cross-reference information. **Settings** menu contains an item for enabling and disabling the address overlap check of input signals. The Settings menu can also be used for opening the Auto Addressing dialog. By using the Auto Addressing dialog the address offsets used in copy-paste of cross-reference information can be selected.

Help Menu

Help menu displays information concerning the Signal Cross-reference Tool, such as identification, version number, revision and license information.

Tool Bar

The tool bar of the Signal Cross-reference Tool shown in Figure 7 contains twelve short-cut keys for quick access of the different functions. From left to right the functionality of the keys is: exit, edit-cut, edit-copy, edit paste, selection method-line, selection method-free, edit-delete, signal-add, cross-reference-import, cross-reference-export, signal-edit and signal-scale. Tool bar also provides a combo-popdown menu for selecting the view and a short-cut key for opening the View Definitions dialog.

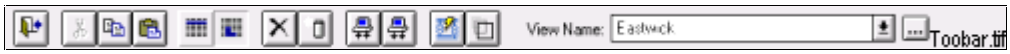

Figure 7. The toolbar of the Signal Cross-reference Tool

Indications Page

The **Indications** page is used for defining signal cross-references for indications i.e. input process objects. You can define address and the signal handling attribute of the NCC to which the signal should be sent or you can define the alarm group the signal activates, or both.

By default, indication signals include columns for five attributes. The included attributes are:

- Object Identifier.
- Object Text.
- Logical Name.
- Index.

- Process Object Type.

The page has also NCC specific columns:

- Address.
- Alarm group.
- Signal handling.

Signal related definitions are displayed inside the indication signal definition area at the bottom of the **Indications** page. The indication signal statistics area includes numeric information concerning signals of the application. For example a number of indication signals, connected (cross-referenced) signals and the number of signals that are connected to a selected NCC are shown.

Commands Page

The **Commands** page is used for defining signal cross-references for commands, i.e. output process objects. The page includes columns for same attributes as the indication page. The following information is entered for each command signal per NCC:

- Command type.
- Purpose.
- Command group.
- Return indication.
- Address.
- Signal handling attributes.

Commands page includes also statistics just as the indication page.

NCCs Page

The **NCCs** page is used for adding or deleting NCCs or for defining properties of the NCCs. The page contains a list of the NCC names. At the bottom of the page there are buttons for adding and deleting NCCs. NCC specific information is displayed on the right side of the page. Protocol, station number, name, comment text operation mode and group alarm information are displayed for the selected NCC.

Alarm groups are listed inside the alarm information area. You can modify alarm groups by clicking corresponding **Add**, **Edit** or **Delete** buttons on the page, when some NCC is selected. Group alarms can be reset using the **Reset** button.

Parameters Page

A separate page for common parameters is included in the Signal Cross-reference Tool to enable defining the COM 500 system and application parameters. These common parameters include time-out parameters and authorisation check parameters.

These attributes and their values are saved into a parameter file. The parameter file is taken as input both for the command procedures and the Signal Cross-reference Tool, when they are started.

3.5.5 Opening and Closing the Signal Cross-reference Tool

Opening

To open the Signal Cross-reference Tool, double-click Signal Cross-reference Tool icon in the System Configuration page of the Tool Manager. You can also click the icon and choose **Open** from the **File** menu. If there is no icon for Signal Cross-reference Tool, see the section 3.5.1 to add an icon for the tool.

The Signal Cross-reference Tool can also be opened from the **Engineering** menu of the COM 500 Communication Diagnostics dialog.

Closing

To close the Signal Cross-reference Tool, choose **Exit** from the **Cross-reference** menu or double-click its close box.

Properties

When you open or close the Signal Cross-reference Tool, a Progress Indicator appears on screen to display progress of reading indication and command signals from process database. See Figure 8. If the number of indication or command signals according to the current view definition exceeds 10.000, a notification dialog box appears. In this case you should reduce the number of signals included in the current view definition.

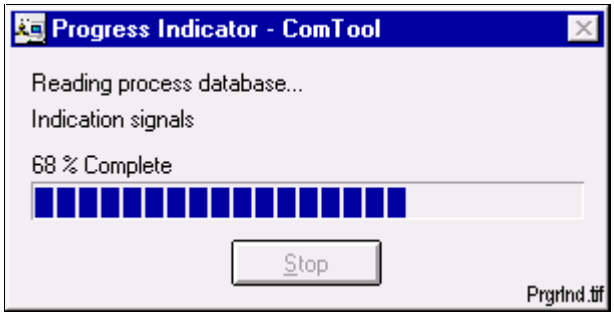

Figure 8. Progress Indicator shows the progress of reading information from process database.

3.5.6 Defining NCC Properties

NCC definitions are definitions concerning the upper level systems COM 500 is meant to be communicating with. COM 500 can be connected up to four NCCs. NCC definitions should be made before any alarm group or signal definitions are made.

NCC properties are defined on the **NCCs** page. See Figure 9. If you edit these fields, the alarm group names and NCC names drop-down lists that are shown in indications and commands pages are also changed.

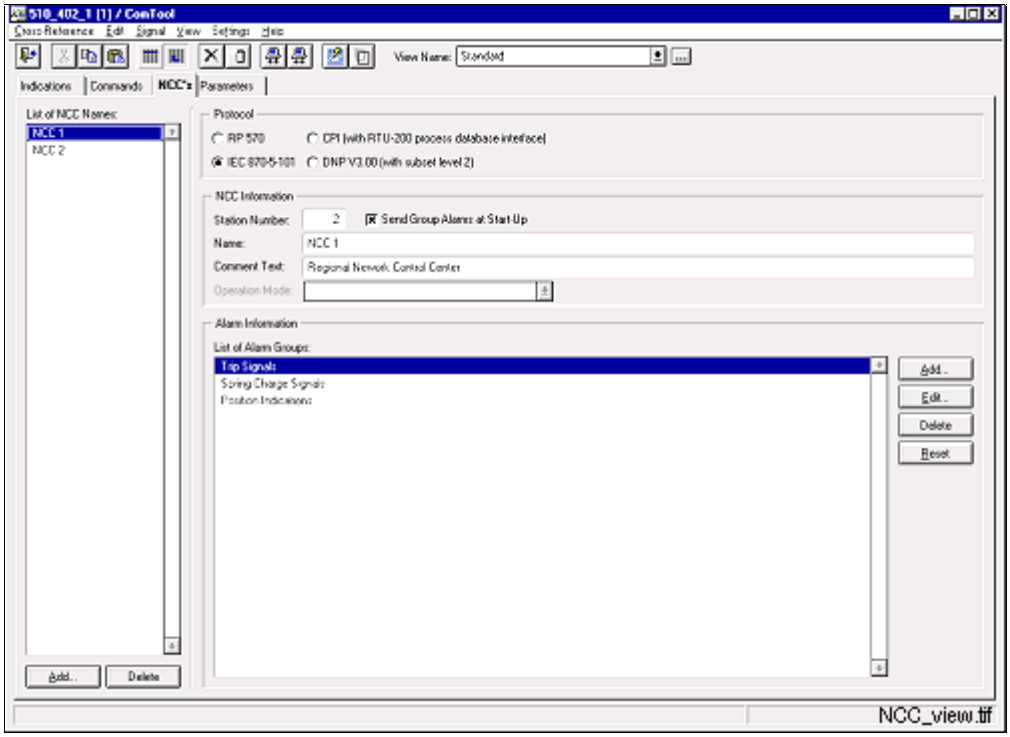

Figure 9. NCC page of the Signal Cross-reference Tool

Adding a NCC

To add a NCC, click the **Add** button at the bottom of the NCCs page. A new NCC is added to the NCC name list. By default, the protocol of the added NCC is RP-570. The first free station number in the system is set as the default station number. NCC name is generated according to the following convention:

name [number]

where:

‘name’	”NCC”
‘number’	NCC number 1..4

This name can be edited and its maximum length is 10 characters.

Deleting NCC

To delete NCC:

1. Click its name in the NCC name list.
2. Click **Delete**. A notification dialog appears.
3. Click **Yes** and the NCC disappears.

When a NCC is deleted all alarm group and signal definitions related to the NCC in question are also deleted.

Defining NCC Properties

Each NCC should have the following properties defined:

- **Protocol**, either RP-570, IEC 60870-5-101, DNP V3.00 or CPI. This is the communication protocol used for communication with the upper level system.
- **Station number**. This is the number of the STA:S object number representing the upper level system.
- **Name**, default name NCC 'n' is given when a new NCC is added, see above. This name is used also as the command source name in the COM 500 command authority check mechanism.
- **Description**. This is a free text with the maximum length of 30 characters.
- **Send Group Alarms at Start-up**. If this parameter is set on, the value of the group alarm signal is sent to the upper level system as non-time-tagged binary signal.
- **Operation Mod**e. This parameter is valid only for DNP V3.00 protocol and it describes how messages are sent between the slave (COM 500) and the master (NCC). The descriptions of the operation modes are as follows (please refer to DNP V3.00 documentation for details):
    - **Quiescent Operation**. In this mode the master does not poll the slave, all communication is based on unsolicited report-by-exception messages. The master can send application layer confirmations to the slave.
    - **Unsolicited Report-by-Exception Operation**. The communication is basically unsolicited, but the master occasionally sends integrity polls for Class 0 data to verify that its database is up to date.
    - **Polled Report-by-Exception Operation**. The master frequently polls for event data and occasionally for Class 0 data.
    - **Static Report-by-Exception Operation.** The master polls only for Class 0 data or the specific data it requires.

To define the properties of a NCC:

1 Click either RP 570, IEC 60870-5-101, CPI or DNP V3.00 **Protocol** check box to select the protocol the NCC uses.

2 Enter **Station number**, **Name** and **Description** for the NCC,

3 Click the **Send Group Alarms at Start-up** check box if this feature is wanted.

4 In case of the DNP V3.00 protocol, select the **Operation Mode**.

3.5.7 Alarm Groups

A group alarm collects several inputs into a single binary alarm signal, which is forwarded to the NCCs. For example all alarms within one bay, or all trip signals can be grouped to reduce the amount of signals sent to the NCCs. The group alarm can also be used to differentiate alarms. For example according to priority of the alarm the signals can be divided into different groups.

The COM 500 group alarm works as follows:

- When the first of the connected signals gets into alarming state, the group alarm is set, i.e. binary value 1 is sent to the alarm group address.
- As long as at least one of the connected signals is alarming the group alarm remains set.
- When the last of the connected signals gets into non-alarming state, the group alarm is reset, i.e. binary value 0 is sent to the alarm group address.

Each NCC can have up to 255 alarm groups and 65535 signals can be connected to an alarm group. Note that it is possible to connect a signal to an alarm group without mapping the signal itself to a NCC, meaning that the value of the signal is not sent.

Adding Alarm Group

To add an alarm group:

1 Click **Add**. Alarm Group Definitions dialog appears. See Figure 10 below.

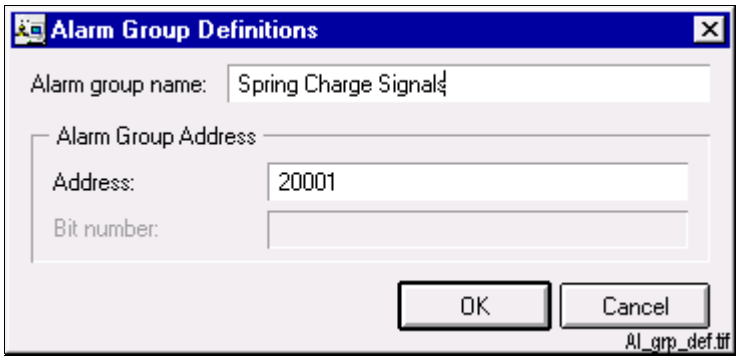

Figure 10. Alarm Group Definitions

2 Type **Alarm group name**.

3 Enter the alarm group **Address** depending on the NCC protocol.

4 Click **OK**.

5 A new alarm group is added to the alarm group list. If an alarm group with the same address already exists, a notification dialog box appears. Change the address of the new or the existing alarm group.

Editing Alarm Group

To edit an alarm group:

1 Select an alarm group in the list.

2 Click **Edit** to open Alarm Group Definitions dialog.

3 Change the definitions in the corresponding text fields.

4 Click **OK**.

Deleting Alarm Group

To delete an alarm group:

1 Click an alarm group in the list to select it.

2 Click **Delete** button. The notification dialog appears.

3 Click **Yes**. The alarm group disappears.

Resetting Alarm Group

To reset an alarm group:

1 Click an alarm group in the list to select it.

2 Click **Reset** button. The notification dialog appears.

3 Click **Yes**. The group alarm is reset.

Resetting the group alarm does not reset the alarms of the signals connected to the group alarm.

3.5.8 Defining Views

The query conditions of the signals included in the indications and commands pages are modified in the View Definitions dialog.

Opening View Definitions Dialog

To open the View Definitions dialog select **Define** from the **View** menu. See Figure 11.

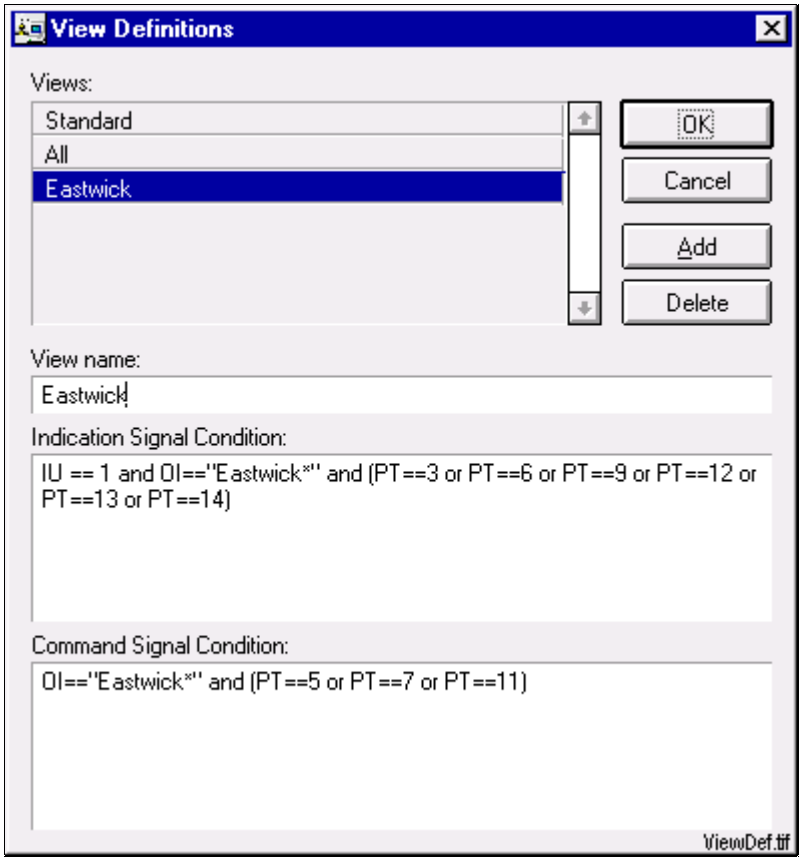

Figure 11. View Definitions dialog

View Definitions

By default, a non-editable view called Standard is assigned with The Signal Cross-reference Tool. To view the indication and command signal query conditions defined for a view, click the view name in the list.

Adding View Definition

To add new view definition, click **Add**. A new view name is added to the list based on the convention:

name [view number]

where:

‘name’	”Name for view”
‘view number’	number of the view in View Definitions

By default the query conditions of the Standard view are copied to the new view. These conditions can be modified to suit the purposes of the new view.

Closing the View Definitions Dialog

Click **OK** to close the View Definitions dialog. The validity of the new view condition is checked, and if it is invalid, a dialog is shown. The names of valid view definitions are added as separate menu items to the **View** menu.

You can modify column attributes attached to the indications and commands pages using the Attribute Definitions dialog.

3.5.9 Defining Column Attributes

Opening Attribute Definitions Dialog

To open the Attribute Definitions dialog select **Column Attributes** from the **Signal** menu, see Figure 12 below.

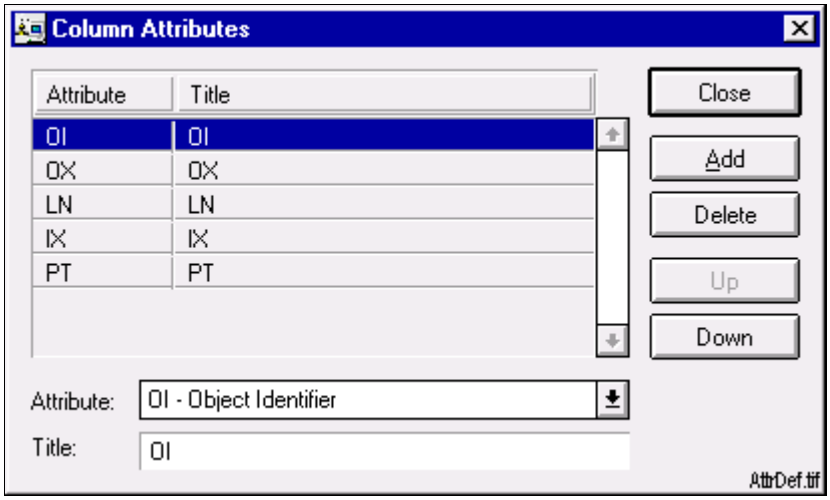

Figure 12. Attribute Definitions

By default, five columns are attached to the indications and commands pages. These are:

- Object Identifier (OI).
- Object Text (OX).
- Logical Name (LN).
- Index (IX).
- Process Object Type (PT).

Order of Attributes

To change the order of attributes click an attribute and then the **Up** or **Down** buttons.

Adding New Attributes

To add a new attribute:

1 Click **Add.**

2 Choose the attribute you want to add to the list from the Attribute drop-down list.

3 Type a **Title** for the new attribute. If no title is specified for the new attribute, a two-char attribute is used as the default title. The new attribute is added to the list.

Deleting Attributes

To delete an attribute from the Attribute Definitions dialog, select the attribute in the list and click **Delete**.

It is not possible all the column attributes, at least one attribute must be defined.

Editing Attributes

To edit some attribute in the Attribute Definitions dialog, select the attribute in the list. Then modify the attribute information either in **Attribute** drop-down list or in title text field.

Closing the Attribute Definitions Dialog

Click **OK** to close the dialog. If new attributes were added to the list, new columns and rulers have been added to Indications and Commands pages. These are located on the right side of the page.

Defining Attribute Column Widths

To define attribute column widths on **Indications** and **Commands** pages, move the rulers. The width of the column is set, when dropping the ruler.

If you change the width of a column in **Indications** page, the corresponding column is also changed in **Commands** page and vice versa.

3.5.10 Defining Auto Address Parameters

Auto addressing parameters are used when cross-reference signals are copied and pasted in the Signal Cross-reference Tool. The purpose of the auto addressing mechanism is to define address offsets to avoid address overlaps when cross-reference data is copied and pasted. For example if the cross-references of an indication connected to a RP-570 NCC is copied and the address of the signal is 002^001 while the RP-570 address offsets are 1 (block number) and 2 (bit number), the NCC address of the signal where the cross-reference is pasted will be 003^003.

For the different NCC protocols the auto addressing parameters are as follows:

- For RP-570 also CPI block address 1 … 255 and bit address 0 … 15.
- For IEC 60870-5-101 IEC address 1 … 65535.
- For DNP V3.00 index 1 … 65535.

Auto addressing parameters are defined using the Auto Addressing dialog shown in Figure 13.

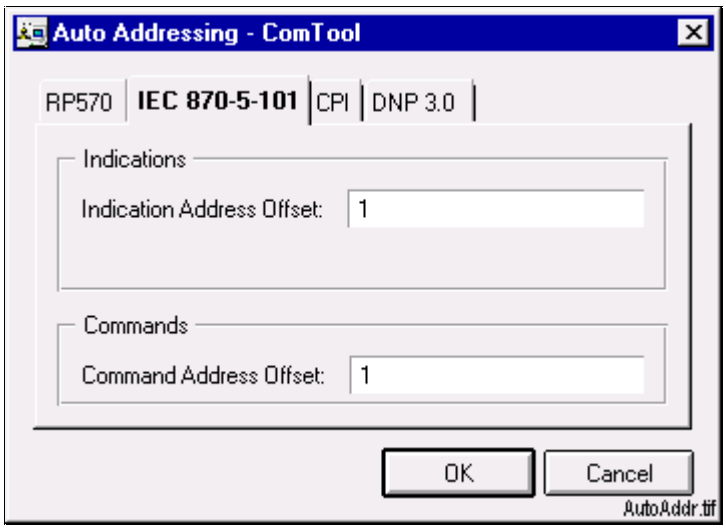

Figure 13. The Auto Addressing dialog

Opening Attribute Definitions Dialog

To open the Auto Addressing dialog select **Auto Addressing** from the **Settings** menu.

Defining Auto Addressing Parameters

To define the auto addressing parameters, select the used NCC protocols and write the address offsets in the corresponding text fields.

Closing the Attribute Definitions Dialog

Click **OK** to close the dialog. The defined auto addressing parameters will be used when cross-reference information is pasted in the Signal Cross-reference Tool.

3.5.11 Enabling and Disenabling the Indication Address Overlap Check

By selecting the corresponding item in the **Settings** menu the address overlap check of indications can be taken in use. When in use, this feature checks for address overlaps every time a new address is given, and if an overlap is detected, the user is notified with a dialog. De-selecting the corresponding menu item can disable the address overlap check.

Note that when the address overlap check is in use it may decrease the performance of the Signal Cross-reference Tool, especially if the number of indication signals is great.

3.5.12 Adding, Deleting and Modifying Signals

It is possible to add, delete and signals, i.e. process objects, by using the signal Cross-reference Tool. Also the attributes of the existing process object can be modified.

Adding Signals

To add signals the following steps should be taken:

1. Select **New** from the **Signal** menu. This opens the New Signal dialog shown in Figure 14.
2. Enter the logical name and index of the new signal. Click **OK** and the signal appears in the Signal Cross-reference Tool.

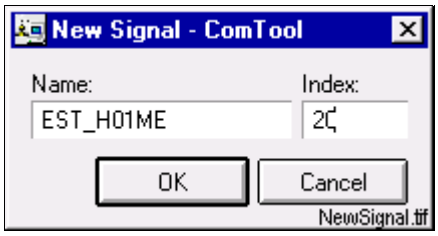

Figure 14. New Signal dialog

Deleting Signals

In order to delete a signal take the following steps:

1. Change the **Selection method** to **Line** from the **Settings** menu.
2. Select the line of the signal to be deleted.
3. Select **Delete** from the **Edit** dialog or press Ctrl + B. This opens a confirmation dialog.
4. Click **OK** and the signal is deleted.

Modifying Signals

To modify the attributes of an existing signal the following steps should be taken:

1. Double click the row of the signal to be edited. This opens the Process Object Tool.
2. Modify the attributes.
3. Click **OK**.

3.5.13 Defining Indication Cross-references

Cross-references for indication signals are defined in the **Indications** page, see Figure 15. Cross-referencing of indication signal means defining either address or alarm group or both. It is also possible to define the signal handling of cross-referenced signal. When an address has been defined for the signal, the indication is sent to the

NCCs. When an alarm group has been defined for the signal, the alarm group is activated, however see the note below.

For each indication (input process object) the following properties should be defined:

- NCC address, i.e. the address in the upper level system where the signal is sent. In RP-570 and CPI the address consists of a block number (1..255) and possibly a bit number (0..15). In IEC 60870-5-101 and DNP V3.00 protocols the address is an integer (IEC address) the range of which is determined by the Information Address Length (IL) attribute of the corresponding station. The address can be omitted if the signal is only connected to a group alarm and the value itself is not sent.
- Alarm group information, i.e. the alarm group to which the signal is connected. This information can be omitted if the signal is not sent to any alarm group. Note that the alarm group is presented as a number in the Signal Cross-reference Tool.
- Signal handling attributes. These attributes define how the signal is handled before it is sent to the NCCs, e.g. a double binary signal can be sent as a singe indication. Note that the alarm group is presented as a number (a bitmask of the numbers of the selected signal handling attributes) in the Signal Cross-reference Tool.

For each analog signal you can also select that is used when the signal is sent to the NCC. If no signal is selected, the signal is not scaled at all.

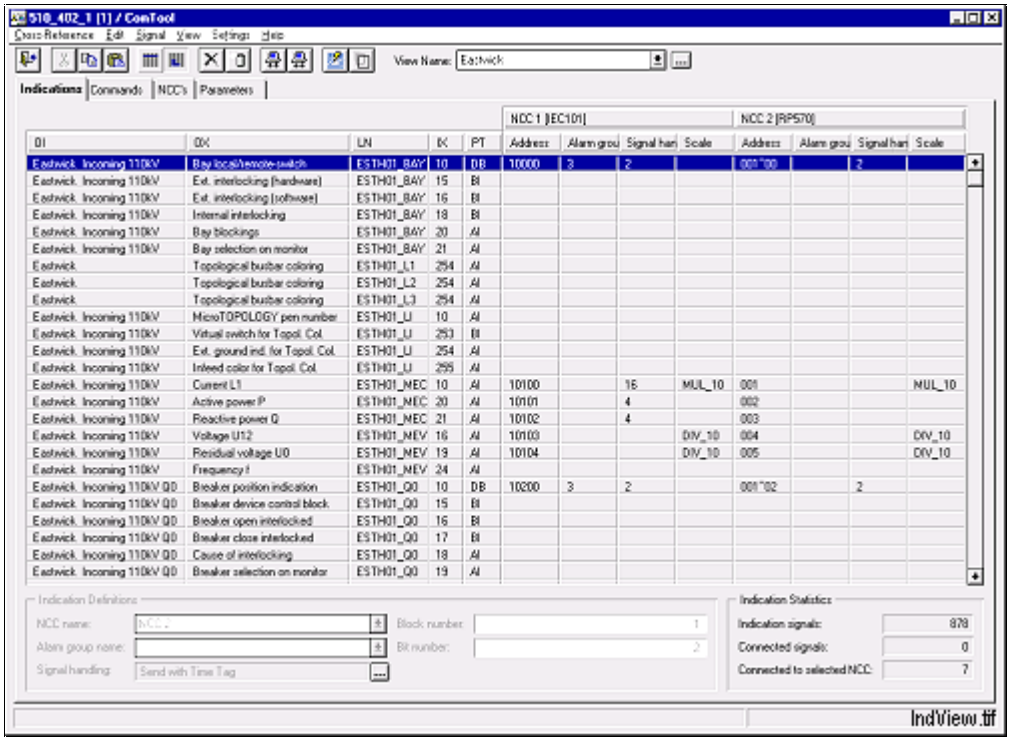

Figure 15. The indication page of the Signal Cross-reference Tool

Adding Cross-reference

To add cross-reference for indication signal:

1 Click the correct signal row in the indication signals list.

2 Choose a NCC from NCC name drop-down list or click a field in a NCC column. Note that if the datatype of the selected signal is not supported by the protocol of the NCC, a message is shown on the status bar and the selections described below are inhibited.

3 Choose alarm group from the drop-down list. The alternatives are the alarm groups that have been added to NCC in the NCCs page.

4 Define address based on block and bit number for RP-570 protocol or CPI, IEC address for IEC 60870-5-101 protocol and index for DNP V3.00 protocol.

5 In case of an analog input signal define a scale. It is possible to select a different scale for each signal and for each NCC.

6 Click the button on the right side of the signal-handling field and the Signal Handling Attributes dialog appears, see Figure 16. Select one or several signal handling attributes from the list of attributes. To select several attributes, hold Ctrl -key down while clicking attributes.

7 Click **OK**. The cross-reference of the indication signal is ready. It is displayed under cross-referenced NCC name.

Indication signal types and the corresponding signal handling attribute values are listed in Table 3. When the signal handling attribute “Project Specific” is selected, a specific block in the corresponding command procedure is executed to enable project specific modifications, see Chapter 4 for details.

Table 3. The signal handling attributes related to different data types

Data type	RP-570/CPI	IEC 60870-5-101	DNP V3.00
Binary input	Project specific Send with time tag Send as inverse value Send as double binary None	Project specific Send with time tag Send with long time tag Send as inverse value Send as double binary Send with and without time tag None	Project specific Send as inverse value Send as double binary Send change Send change with time Send change with relative time None
Double binary	Project specific Send with time tag Send as inverse value Send as single indication None	Project specific Send with time tag Send with long time tag Send as inverse value Send as single indication Send with and without time tag None	Project specific Send as inverse value Send as single indication Send change Send change with time Send change with relative time None
Digital input	Project specific Send with time tag Send as analog value None	Project specific Send with time tag Send with long time tag Send as analog value Send with and without time tag* None	
Analog input	Project specific Send with time tag	Project specific Send with time tag	Project specific Send as 16-bit value

Data type	RP-570/CPI	IEC 60870-5-101	DNP V3.00
	None	Send with long time tag Send as floating point value Send as scaled value Send with and without time tag None	Send without flag Send change with time None
Pulse counter	Project specific None	Project specific Send with time tag Send with long time tag Send with and without time tag None	Project specific Send as 32-bit value Send as delta counter Send without flag Send change with time None

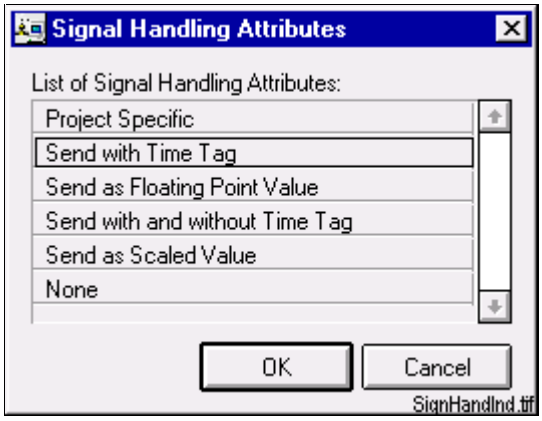

Figure 16. The Signal Handling Attribute dialog

In order to select a scale for an analog input signal, click the **Scale** column of the selected NCC to change the Signal Handling field into the Scale field. Click the button on the right side of this field to open the Scale dialog shown in Figure 17. Select a scale from the list and click **OK**. Note that name of the scale can not be longer that ten characters.

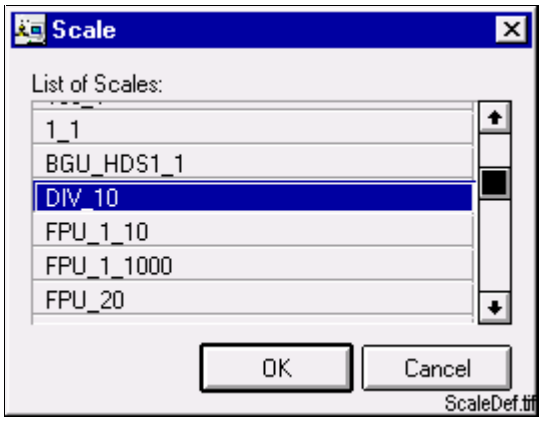

Figure 17. The Scale dialog

You can edit the attributes of a selected scale object by selecting **Scale** from the **Signals** menu when a cell containing a scale name is selected. This opens the Scale

Object dialog shown in Figure 18. The scaling of analog signals in COM 500 is described I detail in Chapter 4 of this document.

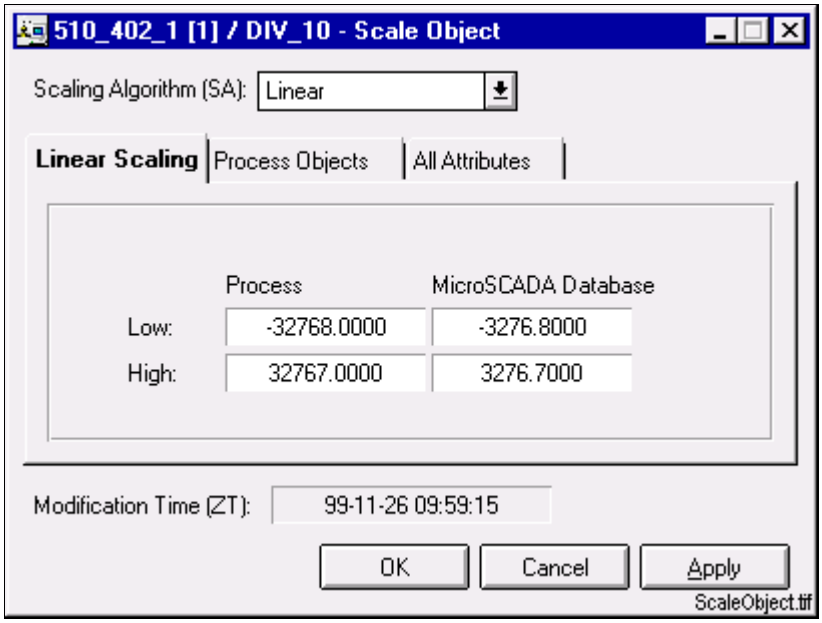

Figure 18. The Scale Object dialog

Deleting Cross-reference

To delete cross-reference from the signal:

1. Click an indication signal row.
2. Choose an NCC from the **NCC Name** drop-down list.
3. Clear the **Address** and **Alarm group** fields. If a signal handling attribute is used, set it to **None.**
4. If the alarm group has been defined for indication signal, select **None** from the **Alarm group** drop-down list.
5. Cross-reference is deleted between indication signal and corresponding NCC.

Cross-reference information can also be deleted using the delete function as follows:

1. Set the **Selecting method** as **Free** from the **Settings** menu or the tool bar.
2. Mark the cross-reference data to be deleted with mouse on the indication page.
3. Delete the cross-reference data by selecting **Delete** from the **Edit** menu, using the corresponding short-cut key of the tool bar or by selecting Ctrl+D on the keyboard.

Cut or Copy and Paste of Cross-reference

Cross-reference data can be cut, copied and pasted from one signal to another, or from a group of signals to another by taking the following steps:

1. Set the **Selecting method** as **Free** from the **Settings** menu or the tool bar.

2 Mark the cross-reference data to be cut/copied and pasted with mouse on the indication page.

3 Cut or copy the cross-reference data by selecting **Cut** or **Copy** from the **Edit** menu, using the corresponding shortcut key of the tool bar or by selecting Ctrl+X/ Ctrl+C on the keyboard.

4 Click the field in the upper left corner of the area where the cross-reference data is to be pasted.

5 Paste the cross-reference data by selecting **Paste** from the **Edit** menu, using the corresponding short-cut key of the tool bar or by selecting Ctrl+V on the keyboard.

3.5.14 Defining Command Cross-references

Cross-references for commands (output process objects) are defined in the command page of the Signal Cross-reference Tool, see Figure 19.

In MicroSCADA different protocols are “seen” in a different way to the application level. Different protocols have also different command philosophies. This means that in addition to being a protocol converter COM 500 has to be able to make conversions between different methods of command handling. For this purpose the following parameters must be given in the Signal Cross-reference Tool for each command.

1 Command type, values:

For binary output, analog output and digital output process objects:

- Direct command. Note that the object commands of IEC 870-5-103 devices must be made using this command type.

For binary output process objects:

- Secured command with two output objects (e.g. REF 542, tap changer raise/ lower commands).
- Secured command with four output objects (e.g. SPACOM devices, REF 543).
- Secured command with one output object (e.g. IEC 870-5-103 devices, RTU 200).

For analog output process objects:

- Secured command with one output object (e.g. REC 561 devices).

2 Purpose, function of an individual signal (output process object), values depend on the command type and are as follows:

- Open command (secured command/2 output objects).
- Close command (secured command/2 output objects).
- Open select (secured command/4 output objects).
- Close select (secured command/4 output objects).
- Execute (secured command/4 output objects).
- Cancel (secured command/4 output objects).

For direct commands the purpose has no meaning and selecting it is disabled.

3 **Command group**, given if there are several controlled objects (e.g. breaker and truck) with output process objects sharing the same logical name, the output objects of the different object should be given different command group numbers. Value: 1..5.

For each command signal (output process object) the following properties should be defined:

General, signal-related information (common for all NCCs):

- Command type, i.e. how the command is presented in the COM 500 process database.
- Purpose of the signal.
- Number of the command group.
- Return indication. This is the input process object that is updated as the result of the command, e.g. the position indication object of a breaker is the return indication of the breaker (open/close) command. This information ensures that the IEC 60870-5-101 messages are sent in the correct order to the NCC when commands are received from the NCC. If IEC 60870-5-101 protocol is not used to connect any of the NCCs, this information can be omitted.

NCC-related information:

- Signal handling attributes. These attributes define how the signal is handled before it is sent to the process devices, e.g. an object command can be received as an inverse value. Note that the alarm group is presented as a number (a bit mask of the numbers of the selected signal handling attributes) in the Signal Cross-reference Tool.

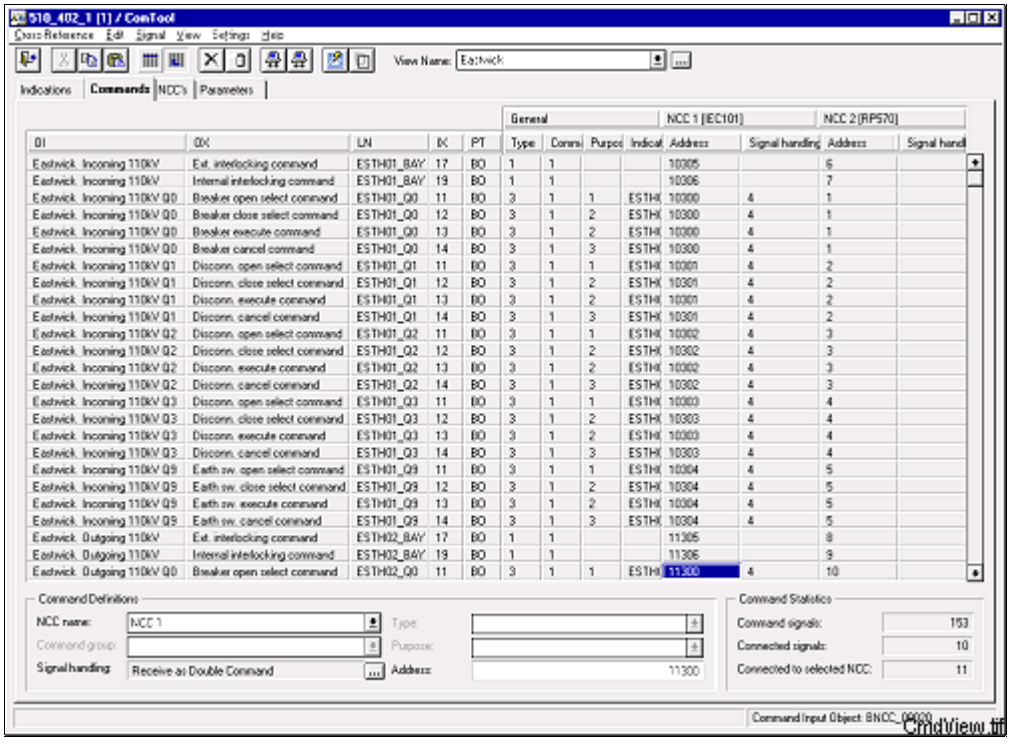

Figure 19. Commands page

Adding Cross-references

To add cross-reference for a command signal:

1. Click a row in the command signals list.
2. Select **General** column by clicking same command signal row under **General** column or selecting **General** from **NCC** name drop-down list inside the command definition area.
3. Choose command type from **Type** drop-down list.
4. Choose signal purpose from the **Purpose** drop-down list.
5. Enter the **Command Group** number. When the command type is set to direct or Secured / 1 output object, the command group of the cross-referenced signal is automatically set to 1. Note that if you want to change the command group number, you must first set the command type to **None** and then back.
6. Click the button next to the **Indication** field and the Indications dialog is opened, see Figure 20.

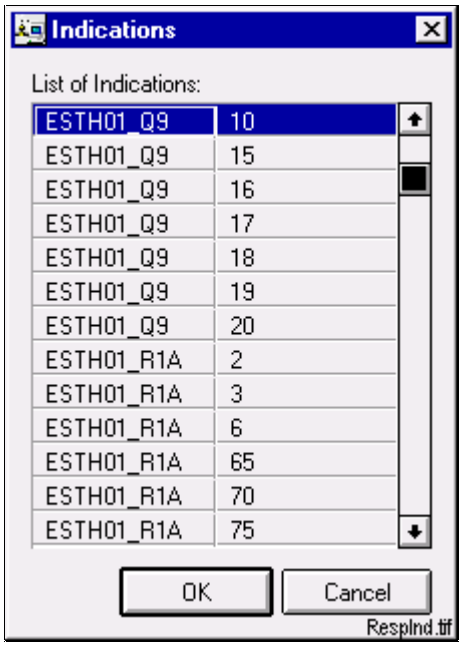

Figure 20. Indications dialog

7. Select the correct indication from the list and click **OK**. The selected return indication appears in the indication field.
8. Enter the **Address** to the command field. Note that if there are several objects constituting one command (e.g. type secured/4 output objects) the same address is copied to all the signals of the command. Note also that if the datatype of the selected signal is not supported by the protocol of the NCC, a message is shown on the statusbar, and entering the address is inhibited.
9. Select the **Signal Handling Attributes** using the Signal Handling Attributes dialog. Note that if there are several objects constituting one command (e.g. type secured/4 output objects) the same address is copied to all the signals of the command. The available signal handling attributes are presented in Table 4.

Table 4. Command signal handling attributes

Data type	RP-570/CPI	IEC 870-5- 101	DNP V3.00
Binary output	Project specific Inverse value Receive as regulation command None	Project specific Inverse value Receive as double command None	Project specific Inverse value Receive as double command Report status to master None
Analog output	Project specific None	Project specific None	Project specific Report status to master None
Digital output	Project specific None	Project specific None	

An example of a cross-referenced command consisting of four signals (secured/4 output objects) is presented in Figure 21. This example is for a breaker open/close command of a REF 543 unit.

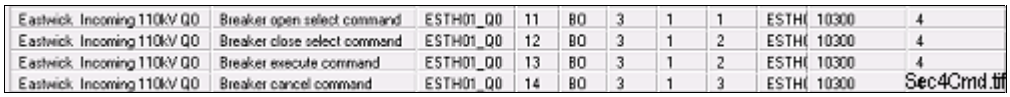

Figure 21. Example of a secured-4-output-object command

Another example shown in Figure 22 shows how a secured command with two output objects is defined for raise/lower command of a SPACOM tap changer.

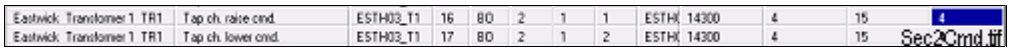

Figure 22. Example of a secured-2-output-object command

IEC 60870-5-101 regulating step commands are handled as double commands in COM 500.

Deleting Cross-references

To delete cross-reference from a command signal:

1 Click a command signal in the command signals list.

2 Select first NCC column, which includes cross-reference to this signal by clicking the row under that NCC column or selecting NCC from **NCC Name** drop-down list.

3 Clear the **Address** field. If a signal handling attribute has been defined, choose None from the Signal handling attributes dialog.

4 Repeat address and possible signal handling attribute for removing all the NCC columns, which have been connected for the selected signal.

5 To choose **General** column, click same command signal row under General column or select **General** from **NCC Name** drop-down list.

6 Choose None from the **Type** drop-down list.

7 The command signal cross-reference is deleted, and the **General** column is cleared.

If command signal cross-reference is secured/2 output objects or secured/4 output objects, Address, Signal handling and Type definitions are removed from the cross-referenced indices, which are in the same command group.

Cut or Copy and Paste of Cross-reference

Delete, cut, copy and paste functions for commands are as for indications except for the fact that only one command is cut or deleted at a time.

3.5.15 Defining Parameters

Parameters are defined and viewed in the **Parameters** page shown in Figure 23. It enables defining the COM 500 configuration. These common parameters include the following information:

- Time-out parameters.
- Authorisation checking parameters.

These attributes and their values are saved into a parameter file. The parameter file is taken as input both for the command procedures and the Signal Cross-reference Tool, when they are started.

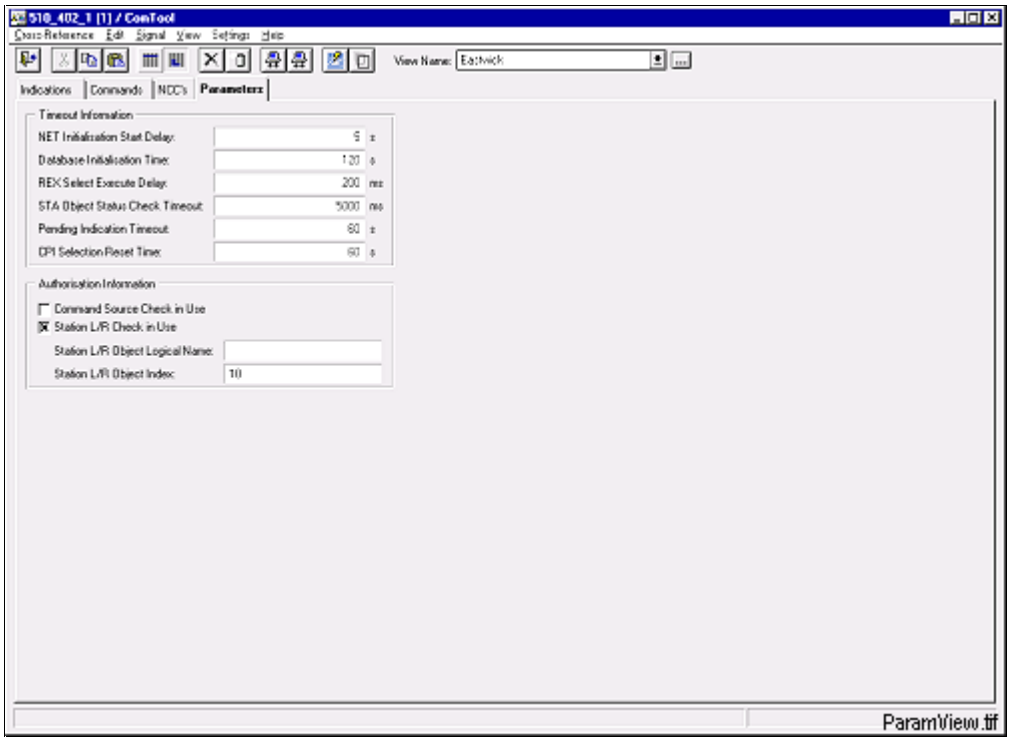

Figure 23. Parameters page

The following parameters can be edited in the Signal Cross-reference Tool:

- **Station L/R Check In Use.** This parameter states whether the station local/remote switch check is in use. When this parameter is set on, COM 500 command procedures check that the value of the station local/remote switch process object has the value corresponding to remote position of the switch. Default value TRUE.
- **Command source Check In Use**. This parameter states whether the LIB 500 command source check is in use. When this parameter is set on, the name of the NCC (e.g. NCC 1) must be in the list of authorised command centres of the application. This list is edited using the Command Authority dialog that can be opened from the LIB 500 Stations menu. If the COM 500 application is not built using LIB 5xx or no MMI (station pictures) is used, this parameter must be set. Default value TRUE.
- **Hard coded STA LR object LN**. Logical name of the station local/remote switch process object. This parameter must be given if the station local/remote switch process object is not created by LIB 5xx otherwise it should be set to "", an empty text string. Default value is an empty text string.
- **Hard coded STA LR object IX**. Index of the station local/remote switch process object. This parameter must be given if the station local/remote switch process object is not created by LIB 5xx, otherwise it should be set to zero. Default value 0.
- **NET Initialisation Start Delay**. Time (seconds) after which the initialisation of the protocol converters in NET is started. This parameter should be set to be the time from MicroSCADA start-up to the moment when all NET lines and stations have been created. Default value 60 s.
- **Database Initialisation Time**. Time (seconds) in which COM 500 database is considered as initialised, i.e. all input process objects connected to process devices have been updated. After this time the COM 500 sends "database initialised" message to the NCCs and accepts RP-570 commands from the NCCs. This parameter should be measured using the actual system with all devices connected. Default value 120 s.
- **STA Object Status Check Timeout**. Timeout (in seconds) used when checking the state of a station reported as SUSPENDED. Default value 5000 ms.
- **CPI Selection Reset Time**. Time (seconds) after which the internal selection of a CPI command is reset. Default value 60 s.
- **REX Select Execute Delay**. This parameter defines the delay (in milliseconds) between consecutive select and execute commands sent to a REX device. Default value 200 ms.
- **Pending Indication Timeout**. Time (in seconds) after which the command connected to a pending indication is terminated. This is time waited for an indication connected to a command to update, before negative command termination is sent to the IEC 60870-5-101 NCC that sent the command. Default value 60 s.

3.5.16 Import and Export of Cross-reference

Export of Cross-reference

Cross-referencesignal information can be exported from the Signal Cross-reference Tool into delimited text files. This functionality can be used for producing documentation of the cross-references or to modify the existing cross-references by editing the exported files with a spreadsheet or ASCII editor. It is also possible to create new cross-references. The exported files have following names and purposes:

- COM_XRNCC.XRF NCC information.
- COM_XRGRP.XRF Alarm group information.
- COM_XRIND*.XRF Indication information.
- COM_XRCMD*.XRF Command information.

Depending on the number of indications or commands there can be multiple text files. If one indication text file includes more than 10000 lines, the second text file, called COM_XRIND1.XRF, is taken into use and so on.

Export operation can be started from menubar selecting **Export...** from the **Cross-reference** menu, which opens Export dialog shown in Figure 24. User can select one of the alternatives listed below.

- **Indications**. Exported information includes all attributes included in the column attributes, and for each NCC address, alarm group number, signal-handling attribute and scale.
- **Commands**. Exported information includes all attributes included in the column attributes, command type, purpose, logical name and index of the response indication, and for each NCC address and signal handling attribute.
- **NCCs.** Exported information includes station number, protocol, NCC name, comment text, operation mode, and the parameter send group alarms at start-up. When NCCs are exported alarm groups are exported to a separate file including the following information: NCC number, alarm group name, alarm group number and address.
- All. This includes all the alternatives listed above.

The names of export files are fixed and the directory is the PICT directory of the current application. It is possible to select whether the first row in export file includes column names, by default this is set. Also the following parameters can be changed:

- **Field separator**. This is the character that separates consecutive fields in the export file. The possible choices are comma, semicolon and tabulator.
- **Text delimiter.** The possible choices are “” (double quote) and none (no delimiter).

Clicking **OK** starts exporting. During export of signals a progress indicator is shown to display the percentage of exported signals. Clicking **No** cancels the operation.

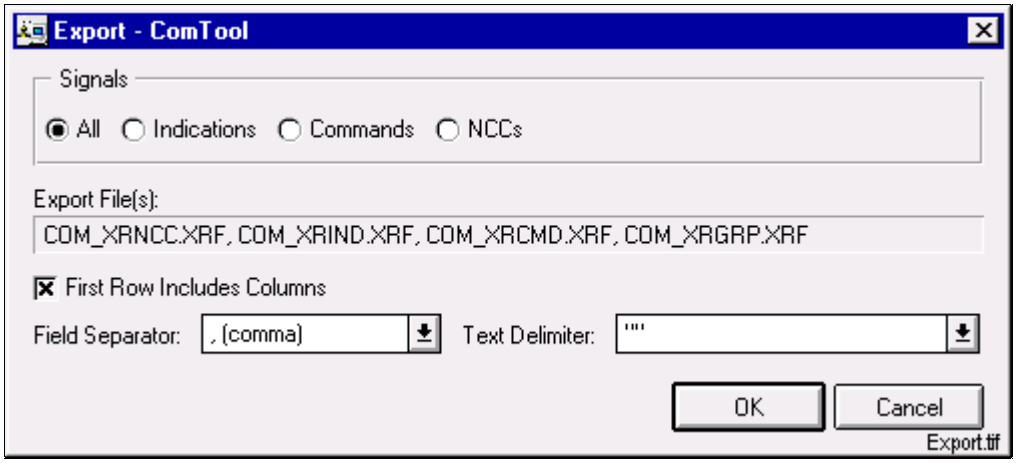

Figure 24. Export dialog

Import of Cross-reference

Importing cross-references from files modifies the cross-information of the current application. Import function has two modes:

- **Modify existing cross-references**. In this mode only the NCCs and signals that are found both in the imported files and the current application. No new NCCs, alarm groups or signals are created.
- **Construct new cross-references**. When cross-reference data is imported by using this mode, the existing cross-references are first reset and then re-created based on the information in the imported files. If new NCCs, alarm groups or signals are detected, they are created.

Import operation can be started from menubar selecting **Cross-reference** and **Import...**, which opens an Import dialog shown in Figure 25. One of the following alternatives can be selected.

- **All.**
- **Indications.**
- **Commands.**
- **NCCs .**

The information included in the alternatives is the same as when exporting. The names of the imported files are the same as when exporting. Parameters **First Row Includes Columns**, **Field Separator** and **Text Delimiter** should be selected according the file to be imported.

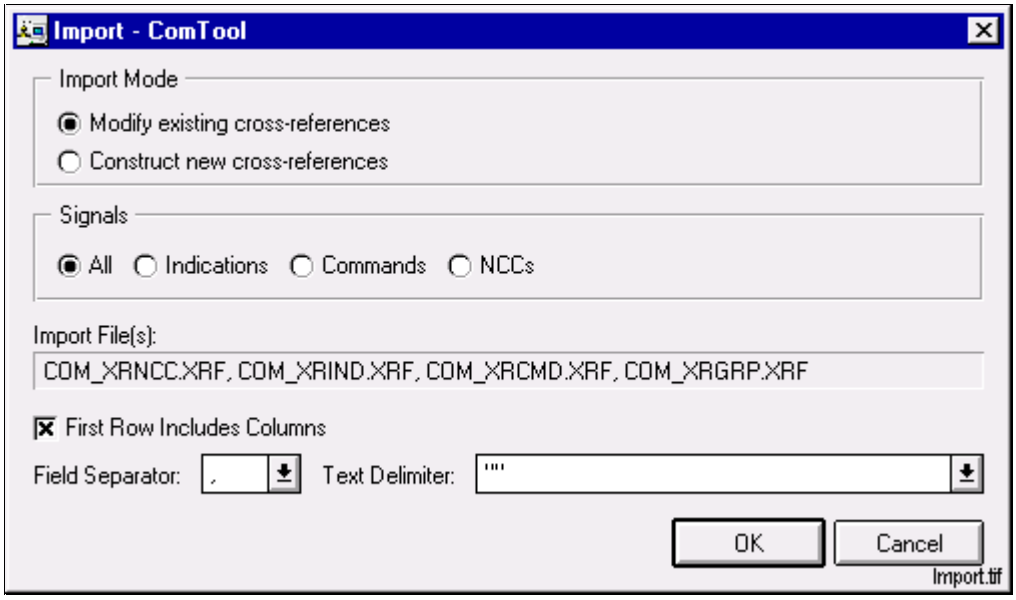

Figure 25. Import dialog

Clicking the **OK** button starts importing. During the import operation a progress indicator is shown to display the amount of imported signals. Clicking **No** cancels the operation.

3.5.17 Printing Cross-references

General

It is possible to print cross-references from the Signal Cross-reference Tool. This functionality can be made for producing documentation of the application to be sent to the customer or the supplier of the network control system.

Configuring a Printer

The availability of printers and the configuration work needed depends on which context the MicroSCADA monitor is opened to. In COM 500 a monitor is by default opened to the context of the MicroSCADA user i.e. the user with the user name “MicroSCADA”. In this case only local printers can be used for printing from the Signal Cross-reference tool. Local printers are:

- Printers connected directly to computer's serial or parallel port.
- Network printers defined as local port.

The procedure how to define a network printer as local port is described in section 3.4 of the System Management Manual.

If a monitor is opened from command prompt or from SCIL with an operating system call, it can be opened to the context of the current operating system user. This requires that the command line option start_as_logon_user is used when opening the monitor. Further details of opening monitors can be found in section 2.6.2 of the System Management Manual. When a monitor is opened to the context of the current

operating system user, the printers that are provided by the operating system to the user are available also for printing from the Signal Cross-reference Tool.

When local printers are used, the MicroSCADA user should have access to these printers.

Since printouts from The Signal Cross-reference Tool are made by using Visual SCIL dialogs, the resolution of the display affects the printout. The resolution should be 1024 x 768 or higher to get all the data properly printed.

Printing

When printing from he Signal Cross-reference Tool the user can select one of several options These options and the information they contain are listed below.

NCC printout:

- Name of the NCC.
- Description of the NCC.
- Protocol of the NCC.
- Station number of the NCC.
- Alarm group names.
- Alarm group addresses.

Indications printout:

- Logical name and index.
- Unit number and address.
- Object identification and object text.
- NCC address.
- Signal handling attributes.

Commands printout:

- Logical name and index.
- Unit number and address.
- Object Identification and Object Text.
- Command type, purpose and command group.
- NCC address.

- Signal handling attributes.

Parameters printout:

- Name of the parameter.
- Value of the parameter.
- Unit of the parameter.

Cross-reference information can be printed by selecting **Cross-reference** and **Print** from the menubar. This opens the Print dialog shown in Figure 26. Print selection can be:

- **Current page**. When this option is selected, the printout contains the active notebook page, i.e. indications, commands, NCCs or parameters.
- **All pages**. This option includes all cross-reference information in the printout.
- **Selected page**. Printout is made according to the further selections made.

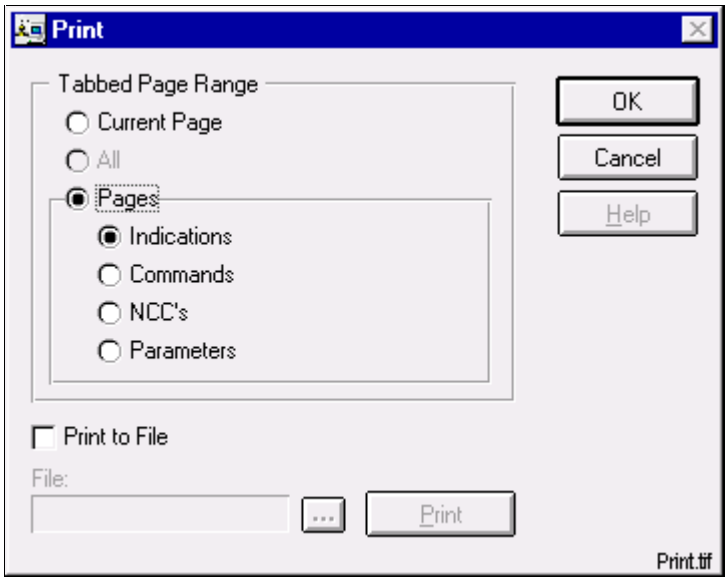

Figure 26. Print dialog

All the printed pages contain header, number of columns, number of column titles and footer. Title is centred in the header of each paper. Column titles are the names of the column attributes. Under each column is cross-reference information printed for each signal. Certain fixed signals are allocated into each printed page. Footer contains the name and number of application, number of each page and total number of pages.

Page Setup

It is possible to change the settings of the printed page by selecting **Cross-reference** and **Page Setup** from the menubar. This opens the Page Setup dialog shown in Figure 27. Page settings include the width of the top, bottom, left and right margins. By default these settings have values: 20, 20, 50 and 20, respectively.

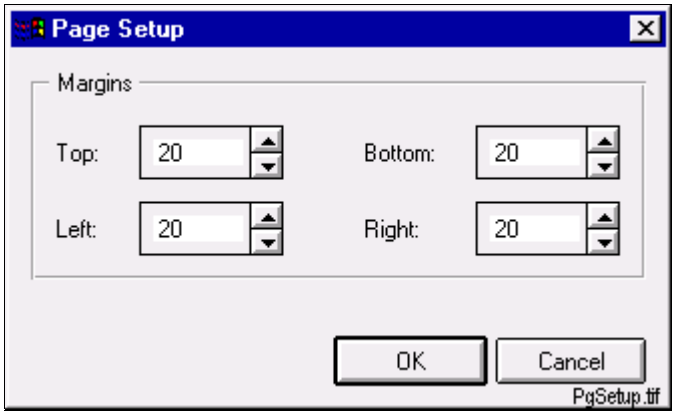

Figure 27. Page Setup dialog

Print Setup

Printer settings can be changed by selecting **Cross-reference** and **Print Setup** from the menubar. This open Print setup dialog shown in Figure 28. This dialog contains options for printer name, properties, paper size, source and orientation.

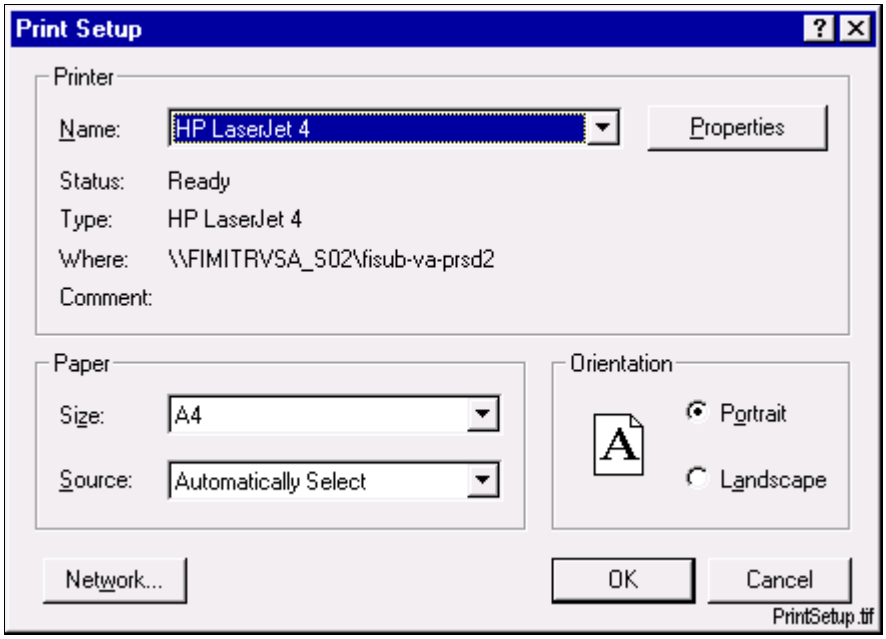

Figure 28. Print Setup dialog

3.6 Using COM 500

3.6.1 Communication Diagnostics Dialog

Communication diagnostics dialog is a tool for displaying communication diagnostics for each NCC and system information in COM 500. From the menubar of tool, it is possible to open the Signal Cross-reference Tool and Tool Manager. It is also possible to change the used font.

NCC Communication Diagnostics

For each NCC of COM 500 a separate tab is included into Communication diagnostics dialog. Each NCC tab includes general information found from the cross-reference tables and diagnostics information. General information displays NCC's name and description, station number of NCC and the protocol of NCC. Diagnostics information displays diagnostic counter names and their values in this NCC. Depending on the used protocol of NCC, the number of diagnostic counters may be different according to the following table below.

Table 5. NCC diagnostics counters

NCC (RP 570 Slave)	NCC (DNP V3.00)	NCC (IEC 60870-5-101)
Transmitted Telegrams	Transmitted Telegrams	Transmitted Telegrams
Failed Transmissions	Failed Transmissions	Failed Transmissions
Timeout Errors	Transmitted Commands	Transmitted Commands
Received Telegrams	Transmitted Replies	Transmitted Replies
Parity Errors	Received Telegrams	Received Telegrams
Overrun Errors	Parity Errors	Parity Errors
Redundancy Errors	Overrun Errors	Overrun Errors
Framing Errors	Check Sum Errors	Check Sum Errors
Buffer Overflow Errors	Framing Errors	Framing Errors
	Buffer Overflow Errors	Buffer Overflow Errors

For NCC of protocol type CPI only general NCC information is displayed, not the diagnostic counter names and values.

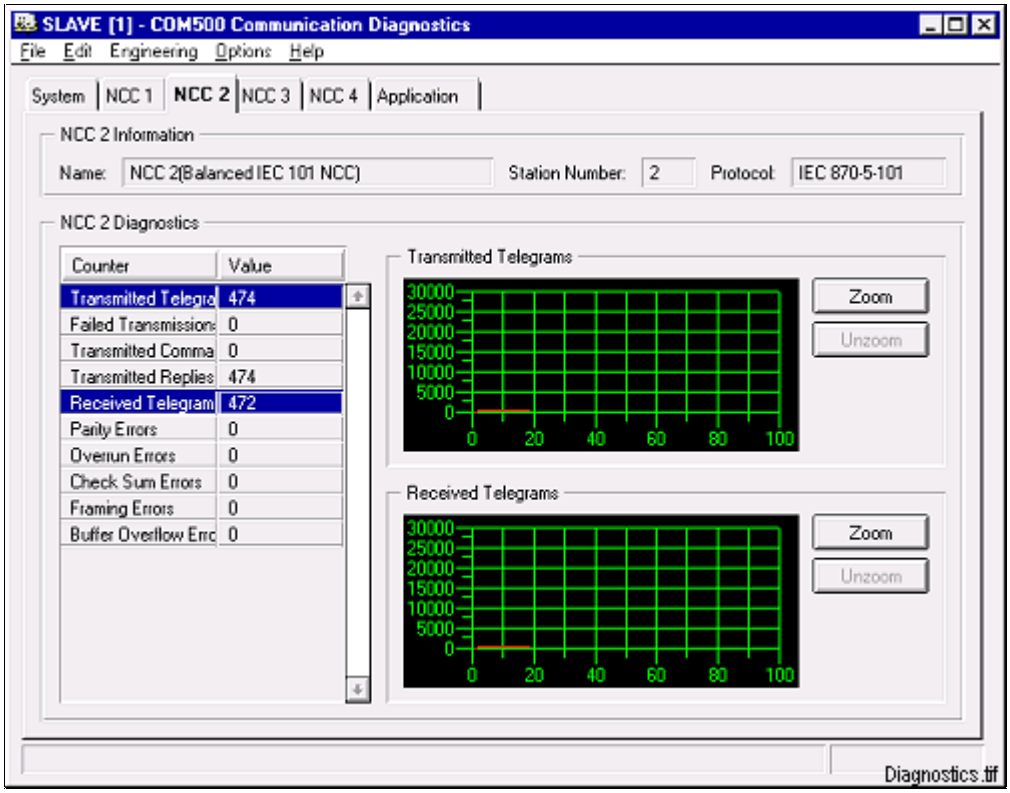

Figure 29. NCC Communication Diagnostics

In the right hand side of each NCC page, there are two graphs for displaying the counter values in a plot diagram, see Figure 29. The diagnostic counters, which are displayed in these graphs can be changed by selecting two appropriate counter names

in the list. The first selected counter is displayed in the upper graph and the second in the lower graph. During selection of the second counter with a mouse click, the Ctrl -key is pressed down. By default the first two counters are being selected, when Communication diagnostics dialog is opened. The update interval for refreshing counter values and these graphs can be selected from the menubar **Options** and **Update Interval**. If there is no need to update the graphs, unselect the last active counter name in the list by pressing the space -key. If counter value exceeds its maximum value 30000, the value starts again from 0.

For each plot graph there is zoom functionality that allows zooming a selected plot area in each graph. To zoom in some graph, click the **Zoom** inside appropriate graph, and select zoom area by pressing and releasing the mouse button for some rectangular area in this graph. When releasing the mouse button, the selected area becomes zoomed. To zoom back the selected graph click **Unzoom**, see Figure 30.

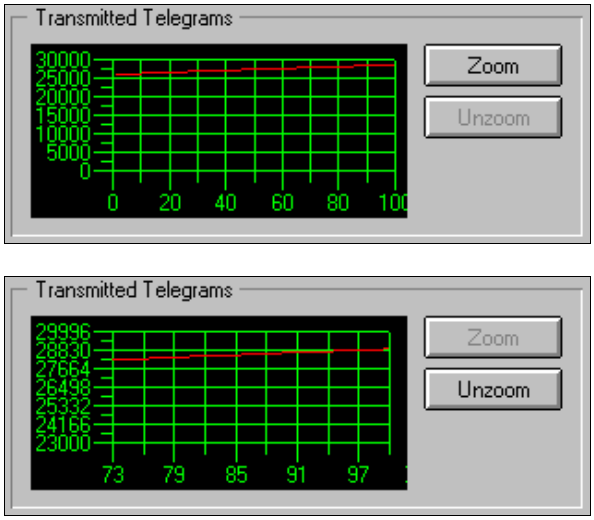

Figure 30. Zoom Functionality of a plot graph

System Information

System page of Communication diagnostics dialog Shown in Figure 31 includes license, application and system information. The license information is displayed with text “This product is licensed to” appended with the name of the COM 500 license owner. Application information displays the application name and number, e.g. Application: COM500, 1. System information displays the license site info together with system node number and station address in parenthesis.

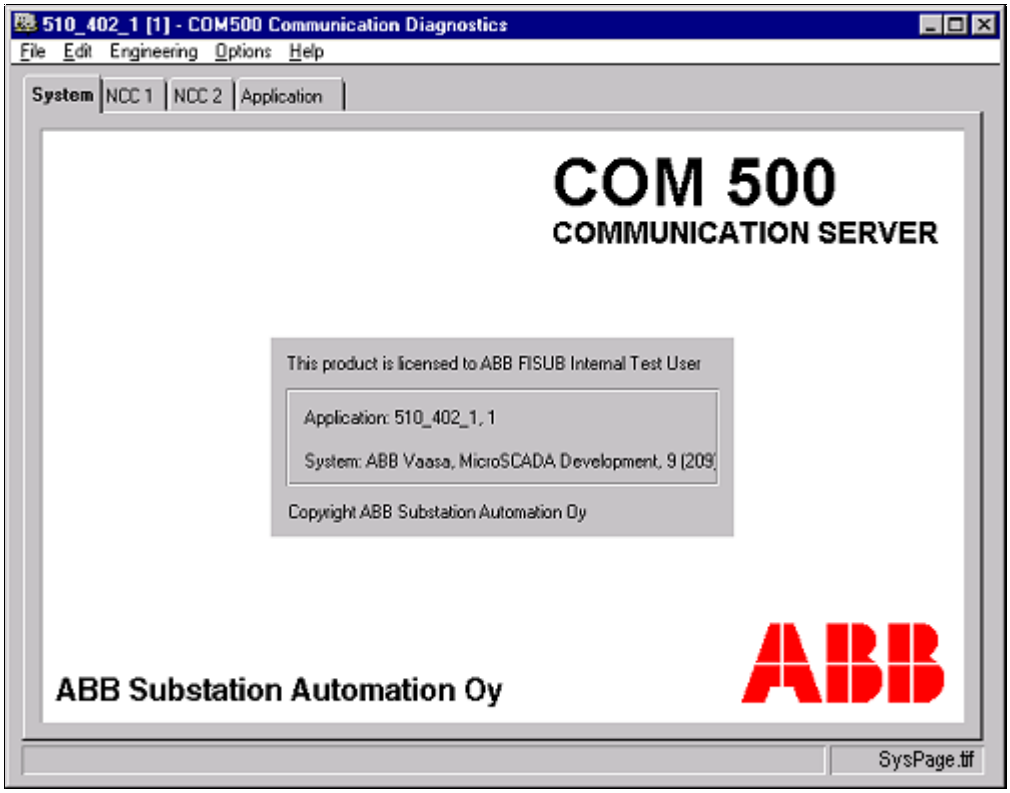

Figure 31. System page

3.6.2 Command Authorisation

COM 500 checks the command authority of a NCC when a command is received. Command authority is determined by the following factors:

- Station local/remote switch must be in REMOTE position, if the parameter Station L/R Check In Use is set on.
- NCC must be given the authority to make commands.

They can both be used at the same time, or only one of them can be used.

In case a command is not authorised or fails, a negative termination is sent to the NCC if IEC 60870-5-101 protocol is used.

4 Technical Description

About this Chapter

This chapter contains descriptions about the functionality, design and configuration of COM 500. Detailed description is given to help the user to understand the function of COM 500.

4.1 HW Requirements

COM 500 requires at least:

- Processing power of a 300 MHz Pentium. Computers with multi-processors may also be used.
- RAM size of 128 MB, smaller RAM size may result in degraded performance.
- COM 500 application requires approximately 100 MB of disk space. The recommended total disk capacity is at least 1 GB. Any SCSI or IDE controller supported by Windows NT may be used.

Options:

- A CD-ROM device is recommended for the installation of Windows NT and the COM 500 software.
- A 3.5”/1.44 MB floppy drive or a 4/8 GB DAT tape drive is recommended for system backup purposes.
- Any Ethernet adapter supported by Windows NT may be used for connecting the base system computer to the LAN.
- For time synchronisation, a PC 31/32 radio clock board from Meinberg Funkuhren, Germany, may be used. The board contains a radio receiver for the Frankfurt DCF-77 77 kHz radio transmitter. Optionally the PC 32 board can be connected with a serial line to a GPS receiver. LON Clock Master card in the LON Star Coupler, RER 111.

4.2 Available Protocols

Protocols that are supported in COM 500 are shown in the Table 5. The master protocols are used in process communication and slave protocols are used in upper level communication.

Also CPI is listed. It is actually not a protocol but rather an interface that can be used for implementing new protocols to MicroSCADA environment. If protocols that are not listed above need to be used, there is possibility to program the protocol conversion using CPI. The programming is done using C language.

Table 5. Protocols that are supported in COM 500

Protocol	Master or Slave	Support (PC-NET or DCP-NET)
SPA	Master	PC-NET and DCP-NET
LAG 1.2 (LON)	Master	PC-NET
IEC 870-5-101	Master and slave	PC-NET
IEC 870-5-103	Master	PC-NET
ANSI X3.28 - AB (ABB)	Master	DCP-NET
RP 570	Master and slave	PC-NET and DCP-NET
DNP V3.00	Slave	PC-NET
CPI	(supports both)	PC-NET and DCP-NET

4.2.1 Information about Protocols

Some of the protocols listed above are master protocols, which means that the protocol is used for communicating with process devices whereas slave protocols are used for communicating with upper level systems. The following sections provide reference documentation about the protocols supported by COM 500. In addition to the documents mentioned below, the MicroSCADA technology documentation can be used as reference.

SPA

SPA-Bus Communication protocol V2.4.

LAG 1.4 (LON)

LON Application Guidelines for Substation Automation. Version 1.4. This is the ABB standard for using LON as a substation automation protocol.

RP-570 Master and Slave

This protocol is used for communication to the RTU 200 family process devices and between MicroSCADA systems.. Protocol Specification: RTU PROTOCOL 570 and 571 (RP 570 and RP 571). Doc. id: 1KSE 300000-VW. M. Vänskä 95-10-18.

ANSI X3.28 Master

This protocol is used for communication with Allen-Bradley PLC devices. It is also used for communication with SRIO 1000M and SRIO 500M devices. Protocol Specifications:

- PLC-2-Family/RS-232-C Interface Module. Doc. Id: 1771-6.5.8.
- SRIO 1000M and 500MDoc. Id: 34 SRIO 100M 2 EN1 D.

IEC 60870-5-103 Master

This protocol is used for connecting to protection and control devices. Protocol Specification: INTERNATIONAL STANDARD IEC 60870-5-103.

IEC 60870-5-101 Master and Slave

Protocol specification: INTERNATIONAL STANDARD IEC 870-5-101.

DNP V3.00 Slave

The following protocols describe the DNP V3.00 protocol:

- DNP V3.00 DATA LINK LAYER version 0.02 (P009-0PD.DL).
- DNP V3.00 APPLICATION LAYER version 0.03 (P009-0PD.APP).
- DNP V3.00 DATA OBJECT LIBRARY version 0.02 (P009-OBL).
- DNP V3.00 TRANSPORT FUNCTIONS version 0.01 (P009-0PD.TF).
- DNP V3.00 SUBSET DEFINITIONS version 2.00 (P009-01G.SUB).

4.2.2 CPI

CPI software can be used implementing both master and slave protocols. The CPI library contains functions to send and receive messages. It also contains functions to pack and unpack data. The CPI based communication software and COM 500 communicate through TCP/IP network. The communication program that uses CPI interface, must emulate RTU or SPA device placed into a communication frontend.

When using a CPI program with COM 500 revision 2.0 for upper level communication (slave protocol) the following rules apply:

- The STA object attribute interface of the device is as in RP-570 slave.
- The process database interface is of RTU 200 type.
- System messages are as in RP-570 slave.

The CPI interface is designed to support connections to several applications in a base system or in several base systems. All applications in one base system can be reached using the same TCP/IP socket.

When using CPI, the manual “Communication Programming Interface CPI” should be used as a guideline. This manual is delivered on request.

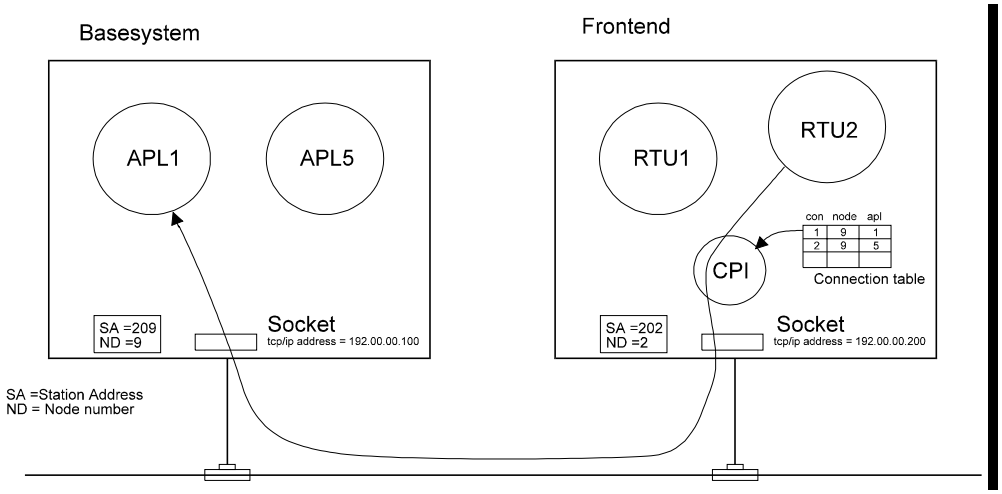

Figure 32. A process unit can communicate with the base system through CPI

4.3 Data Flow

Data flow through COM 500 is handled by the signal routing mechanism that consists of the following parts:

- Cross-reference mechanism, i.e. cross-reference tables created and maintained by the Signal Cross-reference Tool.
- Parameter files.
- Command procedures.

The other parts of the signal routing are:

- System message handling, application and system command handling.
- Command authority check.
- Group alarm handling.

Queue Length Handling

When the signal flow from the process devices to COM 500 is much larger than the signal flow from COM 500 to the NCCs, for example in case of a start-up with a slow or damaged NCC connection, the event channel queue may fill up, which can lead to communication disturbances. This is prevented by implementing a queue length handling mechanism which works as follows:

- When the event channel queue length is 95% of its maximum analogue values and pulse counters are not sent if the registration of the process object has changed, i.e. newer event is in the queue.
- When the event channel queue has its maximum length, other data types are not sent if the registration of the process object has changed.

If a signal is not sent, a message is sent to the notification window. The maximum length of the event channel queue can be set by using the EM attribute of the

application. Reading the EU attribute of the application indicates the current length of the event channel queue.

4.3.1 Indications

4.3.1.1 Mechanism and Datatypes

Indications (input process objects) receive data from the process devices. This data is then sent to the NCCs by COM 500. Data flow from a process device to one or several NCCs is shown in Figure 33.

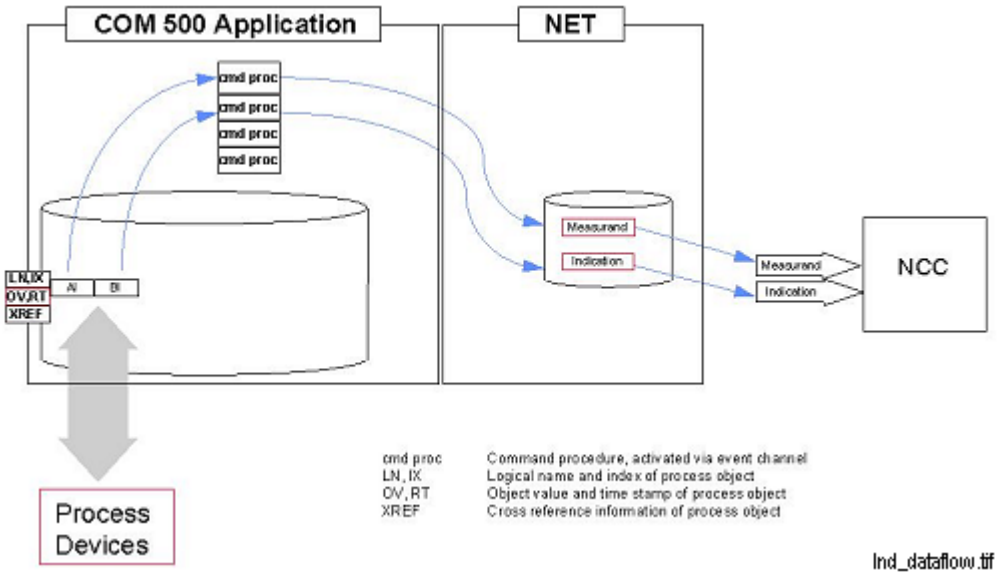

Figure 33. Indications from process units pass through COM 500 application and NET unit to NCC.

The following input data types are supported by COM 500.:

- Binary input.
- Double binary input.
- Digital input.
- Analog input.
- Pulse counter.

Sending input signals to a NCC does not require any additional process objects, except for group alarm, since the data is sent directly to the NET unit using a set of command procedures based on the data stored in the cross reference tables. This data includes:

- Address to which the data is sent (NCC address).
- Alarm group information.
- Information about the handling of the signal (signal handling attributes).
- The scale object used in scaling of analog values.

4.3.1.2 Telegrams and Parameters

The following table presents the different IEC 870-5-101 Application Service Data Unit (ASDU) types that are used in COM 500 when sending data to a NCC. The ASDU used depends on the signal handling attributes selected for the signal in the Signal Cross-reference Tool. Only those signal handling attributes that affect the ASDU used are mentioned.

Table 5. IEC 870-5-101 ASDU types in COM 500

Data type	Signal Handling Attr.	ASDU Type
Binary input	None	M_SP_NA_1(1)
	Send with time tag	M_SP_TA_1(2)
	Send with long time tag	M_SP_TB_1 (30)
	Send as double binary	M_DP_NA_1(3)
	Send with time tag + send as double binary	M_DP_TA_1(4)
	Send with long time tag + send as double binary	M_DP_TB_1 (31)
	Send with and without time tag	M_SP_NA_1(1) + M_SP_TA_1(2)
	Send with and without time tag + send with long time tag	M_SP_NA_1(1) + M_SP_TB_1(30)
	Send with and without time tag + send as double binary	M_DP_NA_1(3) + M_DP_TA_1(4)
	Send with and without time tag + send with long time tag + send as double binary	M_DP_NA_1(3) + M_DP_TB_1(31)
Double binary	None	M_DP_NA_1(3)
	Send with time tag	M_DP_TA_1(4)
	Send with long time tag	M_DP_TB_1(31)
	Send as single indication	M_SP_NA_1(1)
	Send with time tag + send as single indication	M_SP_TA_1(2)
	Send with long time tag + send as single indication	M_SP_TB_1 (30)
	Send with and without time tag	M_DP_NA_1(1) + M_DP_TA_1(2)
	Send with and without time tag + send with long time tag	M_DP_NA_1(1) + M_DP_TB_1(31)
	Send with and without time tag + send as single indication	M_SP_NA_1(1) + M_SP_TA_1(2)
	Send with and without time tag + send with long time tag + send as single indication	M_SP_NA_1(1) + M_SP_TB_1(30)
Digital input	None	M_ST_NA_1(5)
	Send with time tag	M_ST_TA_1(6)
	Send with long time tag	M_ST_TB_1 (32)
	Send as analog value	M_ME_NA_1(9)
	Send with time tag + send as analog value	M_ME_TA_1(10)
	Send with long time tag + send as analog value	M_ME_TD_1(34)
	Send with and without time tag	M_ST_NA_1(5) + M_ST_TA_1(6)
	Send with and without time tag + send with long time tag	M_ST_NA_1(5) + M_ST_TB_1(32)
	Send with and without time tag + send as analog value	M_ME_NA_1(9) + M_ME_TA_1(10)
	Send with and without time tag + send with long time tag+ send as analog value	M_ME_NA_1(9) + M_ME_TD_1(34)

Data type	Signal Handling Attr.	ASDU Type
Analog input	None	M_ME_NA_1(9)
	Send with time tag	M_ME_TA_1(10)
	Send with long time tag	M_ME_TD_1(34)
	Send as floating point value	M_ME_NC_1(13)
	Send as scaled value	M_ME_NB_1(11)
	Send with time tag + send as floating point value	M_ME_TC_1(14)
	Send with time tag + send as scaled value	M_ME_TB_1(12)
	Send with time tag + send as floating point value	M_ME_TF_1(36)
	Send with and without time tag	M_ME_NA_1(9) + M_ME_TA_1(10)
	Send with and without time tag + send with long time tag	M_ME_NA_1(9) + M_ME_TD_1(34)
	Send with and without time tag + send as floating point value	M_ME_NC_1(13) + M_ME_TC_1(14)
	Send with and without time tag + send as scaled value	M_ME_NB_1(11) + M_ME_TB_1(12)
	Send with and without time tag + send with long time tag+ send as floating point value	M_ME_NC_1(11) + M_ME_TF_1(36)
Pulse counter	None	M_IT_NA_1(15)
	Send with time tag	M_IT_TA_1(16)
	Send with long time tag	M_IT_TB_1(37)
	Send with and without time tag	M_IT_NA_1(15) + M_IT_TA_1(16)
	Send with and without time tag + send with long time tag	M_IT_NA_1(15) + M_IT_TB_1(37)

When “Send with and without Time Tag” has been selected, two consecutive messages are sent, one with and one without time tag. This feature is suitable for masters that handle time-tagged and non-time-tagged data separately.

The table below presents the IEC 870-5-101 cause of transmission values that are possible in COM 500 for different types of data.

Table 6. IEC 870-5-101 cause of transmission values used in COM 500

Data type	COT	Explanation
Binary input	3	Spontaneous
	5	Requested
	11	Return information caused by remote command
	20	Interrogated by general interrogation
Double binary	3	Spontaneous
	5	Requested
	11	Return information caused by remote command
	20	Interrogated by general interrogation
Digital input	3	Spontaneous
	5	Requested
	11	Return information caused by remote command
	20	Interrogated by general interrogation
Analog input	3	Spontaneous
	5	Requested
	11	Return information caused by remote command
	20	Interrogated by general interrogation
Pulse counter	3	Spontaneous
	37	Requested by general counter request

The following table describes the input data objects and variations used with DNP V3.00 slave protocol. The data object and variation used depends on the signal handling attributes. The variations described in the table below are default variations,

i.e. variations that are used if no variation is specified by the master in the data request.

Table 7. DNP V3.00 input data objects and variations in COM 500

Data type	Signal Handling Attr.	Obj.	Var.
Binary input*	None	1	1 (2)*
	Send change	2	1
	Send change with time	2	2
	Send change with relative time	2	2
Double binary*	None	1	1 (2)*
	Send change	2	1
	Send change with time	2	2
	Send change with relative time	2	2
Binary output	Report Status to Master	10	2
Analog input	None	30	1
	Send as 16-bit value	30	2
	Send without Flag	30	3
	Send as 16-bit Value+Send without Flag	30	4
	Send Change Event without Time	32	1
	Send as 16-bit Value+ Send Change Event without time	32	2
Pulse counter	None	20 (21)**	2
	Send as 32-bit Value	20 (21)**	1
	Send as Delta Counter	20	4
	Send as 32-bit Value+Send as Delta Counter	20	3
	Send without Flag	20 (21)**	6 (10)**
	Send as 32-bit Value+Send without Flag	20 (21)**	5 (9)**
	Send as Delta Counter + Send without Flag	20	8
	Send as 32-bit Value+ Send as Delta Counter + Send without Flag	20	7
	Send Change Event without Time	22	2
	Send as 32-bit Value+ Send Change Event without Time	22	1
Analog output	Report Status to Master	40	2

* Variation 2 is used when the status of the process object changes.

** Used if counters have been frozen by the master.

4.3.1.3 Handling of Analog and Digital Values

Since some data types have a different value range in MicroSCADA and in the slave protocols supported in COM 500 revision 2.0, some kind of scaling is needed. Analog input values are handled as follows:

In COM 500 you can define the scaling of an analog signal separately for each NCC. This is done by selecting an existing scale object to the signal in question in the Signal Cross-reference Tool.

The scaling algorithm is as follows:

- The “Process” value range is scaled to the “MicroSCADA database” value range of the scale object.

- The value sent to the NCC is limited to the value range defined by the message type of the NCC protocol.
- If the value is over this value range, the overflow bit of the analog telegram is set in IEC 60780-5-101 and DNP V3.00 protocols, and in RP-570 protocol the status of the signal is marked as invalid.

If for example the value of an analog signal sent to an IEC 870-5-101 master (NCC 2) as a scaled value is wanted to be divided by ten, the parameters of scale object should be as shown in Figure 34.

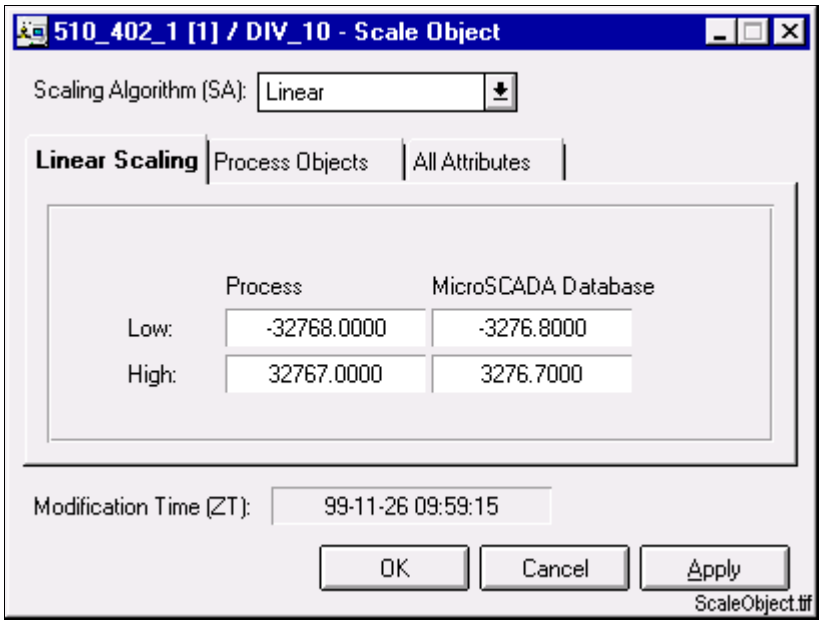

Figure 34. Scale that divides by ten

Digital input values are handled as follows:

- In MicroSCADA the value is 0…65535.
- In RP-570 and CPI the value is 0…65535 or -2000…2000 (sent as analog value).
- In IEC 870-5-101 the value (as seen in SCIL) is 0..127 (step point information) or -32767…32767 (sent as analog value).
- If the value range of the slave protocol is different than in MicroSCADA, a digital input value is limited as follows: the value sent to the NCC is limited between 0 and the maximum of the value range of the slave protocol.

4.3.2 Commands

4.3.2.1 Mechanism and Datatypes

COM 500 receives a command from a NCC as an input, which activates a set of command procedures that sends the command to the process objects that are connected to the process units based on the information stored in the cross-reference tables. This data includes:

- Logical names and indexes of the output process objects.

- Logical name and index of the return indication, if any connected.
- Information about the handling of the signal (signal handling attributes).

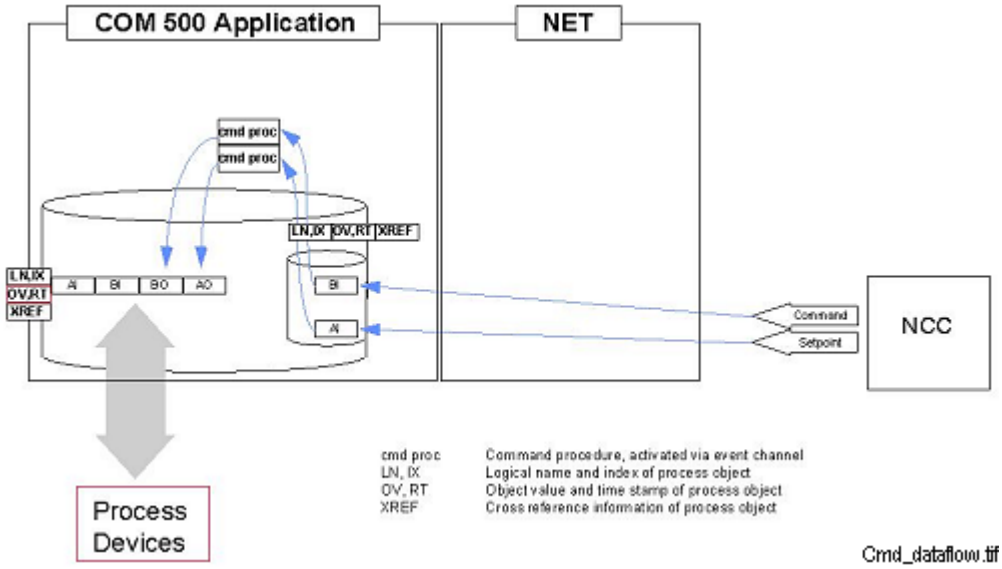

Figure 35. A command from NCC passes through COM 500 application and a process object.

The input process objects that receive the commands from NCCs are created automatically by the Signal Cross-reference Tool. The following output data types are supported in COM 500 revision 2.0A:

- Binary output.
- Digital output.
- Analog output.

4.3.2.2 Telegrams and Parameters

The following table presents the different IEC 870-5-101 ASDU types that are expected in COM 500 when receiving commands from a NCC. The ASDU used depends on the signal handling attributes selected for the signal in the Signal Cross-reference Tool. Only those signal handling attributes that affect the ASDU used are mentioned.

Table 8. IEC 870-5-101 ASDU types in COM 500

Data type	Signal Handling Attr.	ASDU Type
Binary output	Receive as double command Otherwise	C_DC_NA_1(46) C_SC_NA_1(45)
Analog output	Any	C_SE_NA_1(48) C_SE_NB_1(49) C_SE_NC_1(50)
Digital output	Any	C_RC_NA_1(47)

The following table describes the output data objects and variations used with DNP V3.00 slave protocol.

Table 9. DNP V3.00 output data objects and variations in COM 500

Data type	Signal Handling Attr.	Obj.	Var.
Binary output	Any	12	1
Analog output	Any	41	2

Analog output values are scaled and limited using a reverse algorithm that is used when analog input values are sent up.

4.3.2.3 Command Confirmations

IEC 870-5-101 protocol includes the concept of command confirmation and termination. In COM 500 commands sent from a NCC using the IEC 870-5-101 protocol are confirmed and terminated as follows:

- System commands are always confirmed by the COM 500 application software.
- All application commands except reading of user data (ASDU 102) are confirmed and terminated by the COM 500 application software.
- Reset process commands (ASDU 105) are only confirmed, not terminated.

Data commands are confirmed and terminated as follows:

- Command is confirmed when the handling of the command in the corresponding command procedure begins.
- If the command is not authorised or fails a negative termination is sent.
- If the indication related to the command has not been connected to the command in the Signal Cross-reference Tool, the command is terminated when the handling of the command in the corresponding command procedure is finished.

If an indication is connected to a data command, the following rules apply:

- If the output objects are of IEC type, the command is terminated when the termination is received from the device.
- In case of some other output object type the command is terminated when the connected indication is received.
- If the termination (IEC) or indication (other types) is not received within the time-out defined in the Signal Cross-reference Tool, a negative termination is sent.

The value of the PC attribute of each IEC slave station should be set to 0 when using COM 500 software. Otherwise commands may not be properly confirmed or terminated.

4.3.3 System Messages, System and Application Commands

NET unit generates protocol-specific system messages as status codes to inform about some special conditions, for example about loss of communication. COM 500 provides means to use system messages and make actions accordingly.

NCC protocols provide specific application and system commands that are used tasks such as time synchronisation or interrogation of data. These commands are also handled by a COM 500 application.

The system and application commands supported by COM 500 can be found in the interoperability lists and device profiles of the NCC protocols or in the MicroSCADA technology manuals describing the implementation of these protocols.

The NET unit handles some of the application and system commands and some are received to process objects that are created by the Signal Cross-reference Tool and executed by the COM 500 command procedures. To ensure that these commands are received and executed properly, the values of the MI and CA attributes of IEC and DNP slave stations should be checked as stated in Chapter 3 of this document.

4.3.4 Time Synchronisation

By default COM 500 should be synchronised from each NCC. Until a synchronisation message is received, the time stamps of the messages set to the NCC are marked as invalid. After a synchronisation message has been received, time stamps are marked as valid until the system is restarted for the next time.

If the system is synchronised by other means, e.g. by using a local GPS receiver, there is no need to mark the time stamps as invalid. This can be done in different NCC protocols as follows:

- In IEC 670870-5-101 by setting the RM attribute of the IEC slave station so that bit 1 of the value is set (RM = 2 if no other bits are set).
- In RP-570 slave by setting the TI attribute of the SPI station to 1.
- In DNP V3.00 by setting the TC attribute of the DNP slave station to 1.

4.4 Cross-reference and Parameter Files

Cross-reference information and various parameters are are stored in free type process objects and ASCII text files. The files described in this chapter should never be edited manually because it may lead to a severe application malfunction.

4.4.1 Cross-reference Files

The Signal Cross-reference Tool writes the cross-reference information to ASCII text files from where they are loaded automatically at start-up. The following text files are found in the directory /APL/’name’/com500:

- COM_XRNCC1.TXT contains the NCC definitions.
- COM_XRCMD*.TXT contain the cross-reference information of the commands (output process objects).

Cross-reference information of indications is stored in free-type process objects.

4.4.2 System and Application Parameter File

COM 500 uses a set of system and application parameters to control its operation. These parameters are used by the Signal Cross-reference Tool and the COM 500 command procedures. Some of the parameters can be edited in the **Parameters** page of the Signal Cross-reference Tool and some are internal parameters of COM 500. The parameters are saved in the file com500.ini.

4.4.3 Parameter Files of the Signal Cross-reference Tool

These parameter files are used for defining user-interface, attributes and view definitions in the Signal Cross-reference Tool. There are three parameter files, which have the following names, locations and purposes:

- Attr_com contains a list of valid attributes concerning process objects, that can be attached as columns to the indications and commands pages of the Signal Cross-reference Tool. Location: /sc/Stool/SysConf.
- ComTool.ini is located in the directory: /sc/apl/'name'/par/'user' and contains:
    - The definitions for the Signal Cross-reference Tool co-ordinates on screen during the last session, number of columns attached to indications and commands pages and number of selected view definitions.
    - The definitions for ruler positions located between adjacent columns to define the width of the column.
    - Definitions for attribute names and their titles in columns.
- ComView.ini contains the definitions for the number of views assigned to the Signal Cross-reference Tool. The names of views defined using View Definitions. The assigned indication and command signal conditions for every view defined by using View Definitions. Location: /sc/apl/'name'/par/'user'.

4.5 Application Objects

4.5.1 General

Several application objects are created by COM 500. Some of these objects are created automatically at start-up, and some are created when definitions are made in the Signal Cross-reference Tool. Also existing application objects are modified.

In COM 500 the following naming convention is used:

- Event channels, time channels, free-type objects and command procedures are named COM_*:*.
- Process objects and free-type process objects are named BNCC*:P.

COM 500 application objects should not be removed or modified.

4.5.2 Application Objects Created by COM 500

COM 500 creates new application objects for the following purposes:

- Event channels and command procedures for sending data from the input process objects to the NCCs.
- Process objects for receiving commands from NCCs.
- Event channels and command procedures for interpreting commands coming from the NCCs and sending them to the correct output process objects.
- Event channels, process objects and command procedures for group alarm functionality.
- Event channels and command procedures for COM application start-up and initialisation.
- Event channels and command procedures application and system commands.
- Time channels, event channels and command procedures for command termination.
- Free-type objects and process objects for storing cross-reference data.

In addition to the application objects described above, also some other objects may be needed for COM functionality. The Signal Cross-reference Tool creates these objects.

If an address overlap occurs when creating a process object with a predefined address, the execution of the COM 500 command procedure is attached to the secondary objects of the event channel connected to the existing process object.

4.5.3 Application Objects Modified by COM 500

When a cross-reference is attached to a process object, the following modifications are made:

- The Table Index (TI) attribute of the process object is set to point to the due position of the cross-reference table.
- The due event channel (e.g. COM_USAI for analog input process object) is attached to the process object and the event channel activation is set on. If the event channel (AN attribute) of the process object is already reserved, the execution of the COM 500 command procedure is attached to the secondary objects of the event channel.

Other modifications made by COM 500 include:

- The execution of the command procedure COM_COMINI is attached to the predefined command procedure APL_INIT_1.
- The execution of the command procedure COM_SUSSTA is attached to the predefined command procedure APL_EVENT.

4.6 COM 500 Command Procedures

For each MicroSCADA datatype supported in COM 500 revision 2.0A there is a command procedure to deliver this type of data. Each of these command procedures is executed in a parallel queue of its own. The datatype and function of these command procedures can be identified based on the following naming convention:

- The command procedures that send data to the upper level system are named COM_US + data type, for example COM_USDB for double binary data.
- The command procedures that forward commands from the upper level system to the process devices are named COM_DS + data type, for example COM_USAO for analog output data.

In addition to the command procedures designed for the actual data delivery, there are also command procedures for other purposes such as parameter reading, interpreting cross-reference information, executing the system and application commands etc. All the command procedures included in COM 500 revision 3.0 are shortly described in this chapter.

In some cases the command procedures are executed spontaneously, i.e. driven by event channel execution caused by the update of a process object, and in some cases the procedures are executed by other command procedures in a forced way.

The command procedures are created automatically at start-up when a monitor is opened for the first time to an application that has been prepared to COM 500. The source code of the procedures is read from the text files located in the directory /com/active/com_. For performance reasons most of the command procedures are compiled when they are created.

4.6.1 Description of the Command Procedures

COM_101SCR:C

Creates an IEC 870-5-101 slave line and station(s). This command procedure can be used for communication system configuration and executed for example from the predefined command procedure APL_INIT_1:C.

COM_AUTHCH:C

Executes the COM 500 authorisation check mechanism. This command procedure is executed by each of the command procedures COM_DS**:C.

COM_BOCMD:C

Executes a direct or secured object command depending on the command type, the protocol of the upper level system and the protocol of the process device. This command procedure is executed by the command procedure COM_DSBO:C.

COM_CNTINT:C

Executes a counter interrogation command sent from an IEC 870-5-101 NCC. This command procedure is executed by the command procedure COM_IESA:C.

COM_COMINI:C

Executes the start-up mechanism of a COM 500 application by executing a number of sub-procedures. The execution of this command procedure is automatically attached to the predefined command procedure APL_INIT_1:C.

COM_CPISS:C

Interprets the system messages coming from a CPI-connected NCC and executes a number of sub-procedures according to the system message.

COM_DNPSCR:C

Creates an DNP V3.00 slave line and station(s). This command procedure can be used for communication system configuration and executed for example from the predefined command procedure APL_INIT_1:C.

COM_DNPSS:C

Interprets and confirms application commands sent from a DNP V3.00 NCC. Application commands (e.g. general interrogation) are executed by a number of sub procedures.

COM_DSAO:C

Forwards an analog output-type command coming from a NCC to the process device. The value of the command is handled according to the signal handling attributes selected in the Signal Cross-reference Tool.

COM_DSBO:C

Forwards a binary output-type command coming from a NCC to the process device. The value of the command is handled according to the signal handling attributes selected in the Signal Cross-reference Tool. Executes the sub procedure COM_BOCMD:C.

COM_DSDO:C

Forwards a digital output-type command coming from a NCC to the process device. The value of the command is handled according to the signal handling attributes selected in the Signal Cross-reference Tool.

COM_DSXREF:C

Interprets and handles the data stored in the command cross-reference tables. This command procedure is executed by each of the command procedures COM_DS**:C.

COM_GENINT:C

Executes a general interrogation command sent from a NCC depending on the protocol of the NCC. This command procedure is executed by the command procedures COM_IESA:C, COM_CPISS:C and COM_RPSS:C.

COM_GRPAL:C

This command procedure takes care of the group alarm handling.

COM_GRPSND:C

Performs a forced sending of group alarms e.g. at start-up or when re-initializing a NET database.

COM_IECTRM:C

Forwards a command termination sent from an IEC 870-5-101/103 process device to a IEC 870-5-101 NCC.

COM_IESA:C

Interprets and confirms the application commands sent from a IEC 870-5-101 NCC. Application commands (e.g. general interrogation) are executed and terminated by a number of sub procedures.

COM_IESEI:C

Sends an end of initialisation message (ASDU 70) to an IEC 870-5-101 NCC after the COM 500 process database has been updated after start-up (after the time set in the parameter Database Initialisation Time has expired). This command procedure is executed by the command procedure COM_NETINI:C.

COM_IESS:C

Interprets and confirms the system commands sent from an IEC 870-5-101 NCC.

COM_NETINI:C

Initialises the internal databases of the RP-570 and CPI devices in NET. Executes the command procedures COM_GENITNT:C, COM_IESEI:C and COM_RPSDI:C (after the time set in the parameter Database Initialisation Time has expired).

COM_PNDIND:C

Checks whether there are pending indications (indications connected to a command that have not yet been updated) older than the parameter Pending Indication Time-out, and if such indications are found, terminates the corresponding commands (negative termination).

COM_RDDATA:C

Executes a read of user data command sent from an IEC 870-5-101 NCC. This command procedure is executed by the command procedure COM_IESA:C.

COM_RDGEN:C

Initialises and reads the system and application parameters either from the application itself or from the parameter file com500.ini. This command procedure is executed either by the command procedure COM_COMINI:C at start-up or by the Signal Cross-reference Tool when the parameters have been edited.

COM_RDXREF:C

Loads the cross-reference information from the cross-reference files to RAM at start-up. This command procedure is executed by the command procedure COM_COMINI:C.

COM_RESPRC:C

Executes a reset process command sent from an IEC 870-5-101 NCC. This command procedure is executed by the command procedure COM_IESA:C.

COM_RESSEL:C

Resets the internal selection flag of a command coming from a CPI-connected NCC after the time set by the parameter CPI Command Reset Time has expired, if no execute command is received to the same address.

COM_REVDTA:C

Contains the revision information of the current COM 500 version.

COM_RPSCR:C

Creates an RP-570 slave line and station(s). This command procedure can be used for communication system configuration and executed for example from the predefined command procedure APL_INIT_1:C.

COM_RPSDI:C

Sets the DI (database initialised) attribute of a RP-570 slave station after the COM 500 process database has been updated after start-up (after the time set in the parameter Database Initialisation Time has expired). This command procedure is executed by the command procedure COM_NETINI:C.

COM_RPSFT:C

Loads FTABS from a text file to an RP-570 slave station. The FTAB file must be created by using other software.

COM_RPSS:C

Interprets the system messages coming from an RP-570 slave NCC and executes a number of sub-procedures according to the system message.

COM_RPSY:C

Marks an RP-570 slave NCC as synchronized when a clock synchronization message has been received from the NCC.

COM_SUSSTA:C

Sends the indications connected to process devices up after the process device is suspended. This command procedure is connected to the predefined event channel APL_EVENT.

COM_USAI:C

Sends analog input values to the NCCs. The value of the signal is handled according to the signal handling attributes selected in the Signal Cross-reference Tool.

COM_USAO:C

Sends the status of analog output values to DNP V3.00 NCCs if the corresponding signal handling attribute has been selected in the Signal Cross-reference Tool.

COM_USBI:C

Sends binary input values to the NCCs. The value of the signal is handled according to the signal handling attributes selected in the Signal Cross-reference Tool.

COM_USBO:C

Sends the status of binary output values to DNP V3.00 NCCs if the corresponding signal handling attribute has been selected in the Signal Cross-reference Tool.

COM_USDB:C

Sends double binary input values to the NCCs. The value of the signal is handled according to the signal handling attributes selected in the Signal Cross-reference Tool.

COM_USDI:C

Sends digital input values to the NCCs. The value of the signal is handled according to the signal handling attributes selected in the Signal Cross-reference Tool.

COM_USOXR:C

Interprets and handles the data stored in the command cross-reference tables when the status of output objects is sent to DNP V3.00 NCCs.

COM_USPC:C

Sends pulse values to the NCCs. The value of the signal is handled according to the signal handling attributes selected in the Signal Cross-reference Tool.

COM_USXREF:C

Interprets and handles the data stored in the indication cross-reference tables. This command procedure is executed by each of the command procedures COM_US**:C.

4.6.2 Modifications to the Command Procedures

Without the exceptions mentioned in this chapter the COM 500 command procedures should not be modified.

Each of the command procedures COM_US**:C and COM_DS**:C are divided into three sections based on the NCC protocol; the first section is for RP-570 slave, the second for IEC 870-5-101 slave and the third for CPI. In each of these sections there is the following empty SCIL block:

```
;*************** project-specific modifications ***********************
#IF LENGTH(SELECT(%SIGNAL_HANDLING_ATTRS(%NCC_NR),"==1")) > 0 #THEN #BLOCK
   ;write the project-specific code in this block
#BLOCK_END
```

This project specific block is for the project specific modifications that are protocol and data type specific, and can be activated by selecting the “Project Specific” signal handling attribute in the Signal Cross-reference Tool for an individual signal. For example if an analog signal is connected to three NCCs and the “Project Specific” signal handling attribute is selected for the RP-570 NCC, the SCIL code in the project specific block in the RP-570 section of the command procedure COM_USAI:C is executed when this individual signal is sent to the NCCs. The project specific block can be used for modifying the value, time stamp, status information and other parameters that are sent to the NCCs, or the data of the command received from a NCC.

When editing the COM 500 command procedures it should be ensured that the command procedures that are compiled will also be compiled after editing, otherwise the performance of COM 500 may decrease.

4.7 File Summary

The following files are copied to the system when COM 500 revision 2.0 A is installed. All files are located in the directory /com/active/com_.

4.7.1 VSO Files

File	Description
ComTool.VSO	COM 500 Signal Cross-reference tool.
Com_Stand.VSO	COM 500 Diagnostics and Setting Tool.
Com_Start.VSO	COM 500 Start (login) picture.

4.7.2 INI Files

File	Description
Comstand.ini	Initialization file for Diagnostics and Setting Tool.
Comtool.ini	Initialization file for Signal Cross-reference tool.
Tools.ini	Initialization file for Tool Manager.
Toolsupd.ini	File used for updating Tool Manager.
Toolview.ini	Initialization file for Tool Manager.

4.7.3 Text Files for Command Procedure Source Code

File	Description of the Procedure
COM_101SCR.TXT	Creates line and station for IEC 870-5-101 slave.
COM_AUTHCH.TXT	Checks the authority of a command from a NCC.
COM_BOCMD.TXT	Performs a direct or secured command.
COM_CNTINT.TXT	Performs a counter interrogation.
COM_COMINI.TXT	Initialises a COM 500 application.
COM_CPISS.TXT	Receives and interprets CPI system messages.
COM_DNPSCR.TXT	Creates line and station for DNP V3.00 slave.
COM_DNPSS.TXT	Interprets DNP V3.00 application commands
COM_DSAO.TXT	Performs a direct command using an AO object.
COM_DSBO.TXT	Performs a command using BO objects.
COM_DSDO.TXT	Performs a direct command using a DO object.
COM_DSXREF.TXT	Decodes command direction cross-reference data.
COM_GENINT.TXT	Performs a general interrogation.

COM_GRPAL.TXT	Sends up a group alarm to a NCC.
COM_IECTERM.TXT	Forwards command termination to NCCs.
COM_IESA.TXT	Interprets IEC 101 application commands.
COM_IESEI.TXT	Sends an end-of-initialisation message.
COM_IESS.TXT	Interprets IEC 101 system commands.
COM_NETINI.TXT	Initialises the protocol converters in NET.
COM_PNDIND.TXT	Terminates pending indications after a time-out.
COM_RDDATA.TXT	Performs a read-of-user-data command.
COM_RDGEN.TXT	Reads application and system parameters.
COM_RDXREF.TXT	Reads cross-reference information.
COM_RESPRC.TXT	Performs a reset process command.
COM_RESSEL.TXT	Resets selection flags of CPI commands.
COM_REVDTA.TXT	COM 500 revision information.
COM_RPSCR.TXT	Creates line and station for RP-570 slave.
COM_RPSDI.TXT	Sets the DI attribute of an RP-570 slave station.
COM_RPSFT.TXT	Loads FTABS to an RP-570 slave.
COM_RPSS.TXT	Interprets RP-570 system messages.
COM_SUSSTA.TXT	Sends up the signals connected to a suspended station.
COM_USAI.TXT	Sends up analog data.
COM_USAO.TXT	Sends up analog output status.
COM_USBI.TXT	Sends up binary input data.
COM_USBO.TXT	Sends up binary output status.
COM_USBS.TXT	Sends up bit stream data.
COM_USDB.TXT	Sends up double binary data.
COM_USDI.TXT	Sends up digital input data.
COM_USXREF.TXT	Decodes control direction cross-reference data.
COM_USPC.TXT	Sends up pulse counter data.
COM_USXREF.TXT	Decodes monitoring direction cross-reference data.

4.7.4 Text Files for Object Creation and Other Purposes

File	Description
COM_COM.TXT	Source text file for the APL_COM.TXT definition file.
COM_START.TXT	Makes definitions at application start-up.
COM_UPDT.TXT	Updates the application from an older COM 500 rev.
COM_APLOBJ.TXT	Creates the COM 500 application objects.
APL_COM.TXT	Stores application start definitions for COM 500

ATTR_COM.TXtT	Stores process object attribute definitions.

INDEX

Page

A

B

C

D

T

U

Customer Feedback

About This Chapter

This chapter contains information on how to send customer feedback and how to get technical support from the SA Help Desk.

Customer Feedback Database

Customer Feedback is a Lotus Notes database, using which ABB companies can report errors, make improvement proposals and queries related to products manufactured by ABB Substation Automation Oy. Customer Feedback database is connected to the change management system of ABB Substation Automation Oy, which handles all error corrections and improvements made to the products.

Please note that the Customer Feedback database is primarily intended for writing reports about released products. If you are using for example a beta release in a pilot project, this should be clearly stated.

Writing A Customer Feedback Report

When writing a Customer Feedback report, the following general instructions should be taken in consideration:

- Write the report in English.
- Write only one error report, query or improvement proposal in a Customer Feedback report.
- If you are reporting an error, try to isolate the error as well as possible. Describe the sequence of events and actions that lead to the error. If any error messages or other debug information is provided by the system, please write it down. Include also information of the system, e.g. a system diagram, revision information and configuration data.
- If you are making an improvement proposal, try to describe how the improved function should work and avoid providing solutions. Information about the importance of the improvement, e.g. number of projects that require the improvement, helps us to make the decision whether and when the improvement should be implemented.

To make a Customer Feedback report, select Feedback Report from the Create menu. This opens an empty Customer Feedback document. Fill out the fields listed below. A question mark next to a field provides help for filling out the field.

1 Subject. This should contain a short description of the issue. A more detailed description can be given in the Description of Feedback field below.

2 Type of Feedback: Comment/Improvement, Query or Complaint/Error.

3 Customer Information.

4 Reporting Information. This should contain detailed information of the product the report is about.

5 The person who you want to send the feedback to and whether you want to get a reply from that person.

6 Information related to internal handling of the report (not obligatory).

7 Category.

8 You can issue the report by clicking the Issue Feedback button. This will send the report to the selected person and change its status to “in progress”.

Actions

When ABB Substation Automation Oy receives a Customer Feedback report, it is analysed by a sales person or a representative of the technical support. The analyser may ask for additional information in order to completed the analysis. After the report has been analysed, one of the following actions is taken:

- In case of a clear error, the report is moved to the change management system of ABB Substation Automation Oy. In this system, the error is analysed in detail and corrected in a future patch release or major release depending on the severity and impact of the error.
- In case of an improvement proposal, the report is also moved to the change management system, where it is taken as a requirement to future releases.
- In case of a query, an answer is provided.

When Customer Feedback reports are handled in the change management system, the outcome can be one of the following:

No Actions	It is decided that the report requires no further action. If, for example, the problem is caused by a configuration error, it belongs to this category.
Will be implemented in patch/current release	This result means that the correction or new feature will be available in the next official program release.
Moved to future release	This result means that the new feature will be available in some new program release in the near future.

SA Help Desk

ABB Substation Automation Oy provides a technical support service called SA Help Desk to support local engineering centres in their system projects. The purpose of SA Help Desk is to provide support for urgent issues such as:

- Year 2000 issues.
- High-priority issues concerning systems at customers’ sites.

For other kind of technical support, please use the Customer Feedback database. SA Help Desk is available every day from 06:00 to 21:00 Central European Time.

SA Help Desk can be contacted by telephone. The number is:

+358 50 334 1900