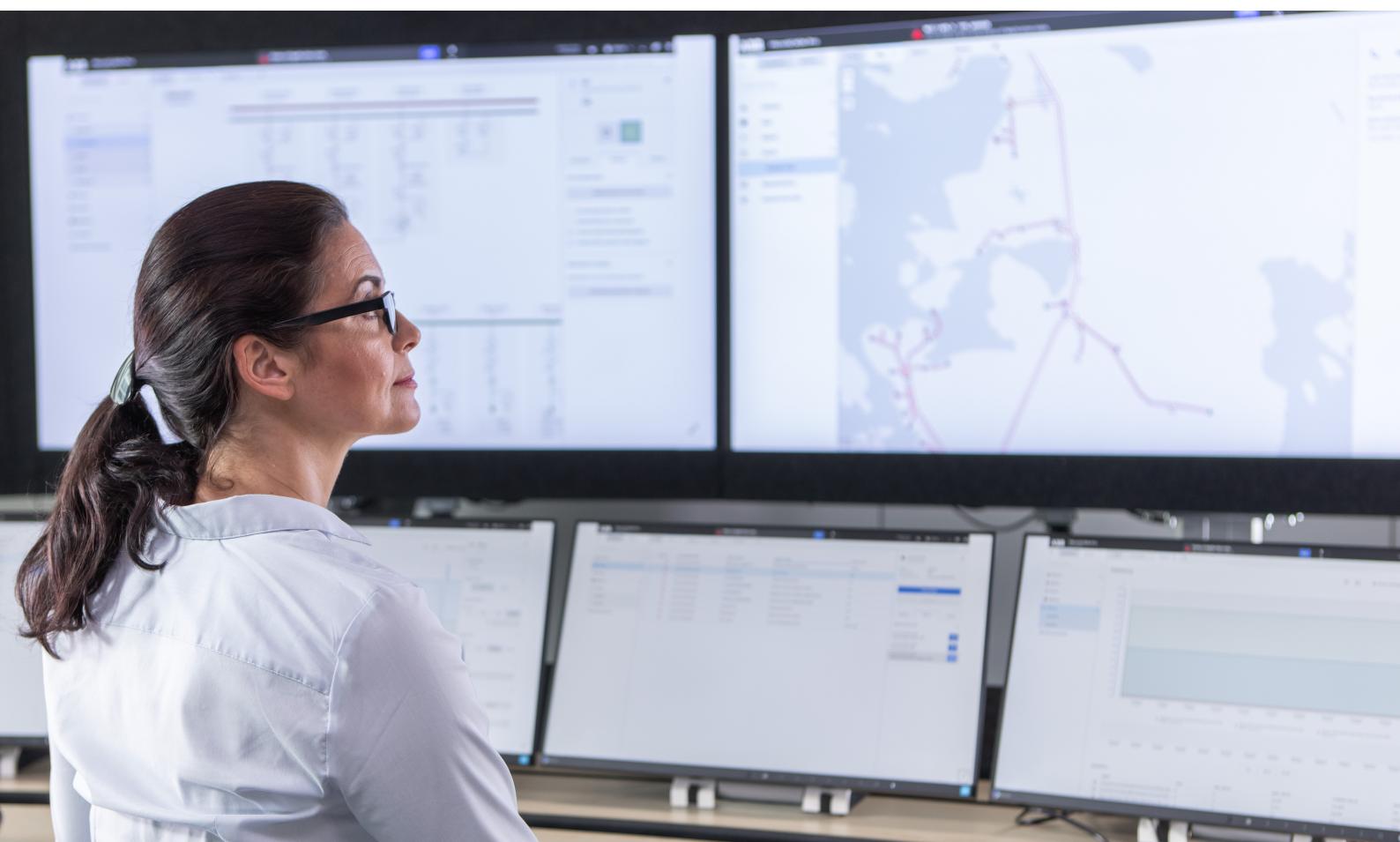

GRID AUTOMATION PRODUCTS

MicroSCADA X SYS600 10.2

IEC 60870-6 (ICCP) Protocol



Trace back information:
Workspace: Main
Version: a3
Checked in 2021-05-25

Contents

1	Copyrights	5
2	Introduction	7
2.1	This manual	7
2.2	Use of symbols	7
2.3	Document conventions	7
2.4	Related documents	8
2.5	Document revisions	8
3	Safety information	9
3.1	Backup copies	9
3.2	Fatal errors	9
4	Instructions	11
4.1	General	11
4.2	Configuration	12
4.2.1	Base system configuration	13
4.2.2	Communication system configuration	15
4.2.2.1	ICCP Configurator	15
4.2.2.2	ICCP Configuration Utility	31
4.2.2.3	Starting and stopping the communication	39
4.2.2.4	Troubleshooting	40
4.2.2.5	Attribute descriptions	43
5	Technical description	51
5.1	General	51
5.2	Communication	52
5.2.1	Data flow	52
5.2.2	Communication attributes, client STAs	53
5.2.3	Communication attributes, server STAs	54
6	Abbreviations and terms	57
7	Updating pilot installations	59
Index		61

1 Copyrights

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

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

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

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

© 2021 Hitachi Power Grids. All rights reserved.

Trademarks

ABB is a registered trademark of ABB Asea Brown Boveri Ltd. Manufactured by/for a Hitachi Power Grids company.

All other brand or product names mentioned in this document may be trademarks or registered trademarks of their respective holders.

Guarantee

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

Third Party Copyright Notices

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

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

2 Introduction

2.1 This manual

This manual provides thorough information on the use of IEC 60870-6 (ICCP) Protocol and information related to it. It describes how to configure the base system and the communication system to establish communication to remote control systems.

2.2 Use of symbols

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



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

WARNING!



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

CAUTION!



Information icon alerts the reader to relevant factors and conditions.

Note!



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

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

2.3 Document conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a dialog, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for file names.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the CTRL key. Although the Enter and Shift keys are not labeled they are written in capital letters, e.g. press ENTER.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the space bar, comma key and so on.

- Press **CTRL+C** indicates that the user must hold down the CTRL key while pressing the C key (in this case, to copy a selected object).
- Press **ALT E C** indicates that the user presses and releases each key in sequence (in this case, to copy a selected object).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
 - The following convention is used for menu operations: **Menu Name > Menu Item > Cascaded Menu Item**. For example: select **File > Open > New Project**.
 - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.
- System prompts/messages and user responses/input are shown in the Courier font. For example, if the user enters a value that is out of range, the following message is displayed: **Entered value is not valid**. The user may be told to enter the string **MIF349** in a field. The string is shown as follows in the procedure: **MIF349**
- Variables are shown using lowercase letters: sequence name

2.4 Related documents

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

Name of the manual	Document ID
SYS600 10.2 Communication Gateway, COM500 <i>i</i>	1MRK 511 468-UEN
SYS600 10.2 System Configuration	1MRK 511 481-UEN
SYS600 10.2 System Objects	1MRK 511 482-UEN
SYS600 10.2 Application Objects	1MRK 511 467-UEN
SYS600 10.2 Status Codes	1MRK 511 480-UEN
SYS600 10.2 Cyber Security Deployment Guideline	1MRK 511 485-UEN

Other referenced manuals

This manual may contain references to or information from following documents:

MMS Lite Reference	Reference manual for MMS-Lite from SISCO, Inc
--------------------	---

2.5 Document revisions

Version	Revision number	Date	History
A	10.2	31.3.2021	New document for SYS600

3 Safety information

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

3.1 Backup copies

Taking backup copies

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

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

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

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

System backup

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

Application backup

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

3.2 Fatal errors

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

Handling

In case of a fatal error:

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



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

CAUTION!

3. The data kept in the main memory at the moment of a fatal error is placed in the drwtsn32.log file with Windows 2003 Server, Windows XP and earlier. By default it is placed under %SYSTEMDRIVE%\Documents And Settings\All Users\Application Data\Microsoft\Dr Watson. Log and dump file paths can be checked with the drwtsn32 application. (Start -> run -> drwtsn32.exe). Analyze and copy the data in these files.

Starting with Windows Server 2008 and Windows 7 the crash handling has changed. The location of the dump files can be read from the registry under the key

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps. The DumpFolder value tells the location of the dump files. Collect the data from this location.

4. Restart the system.

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

Status codes

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

4 Instructions

4.1 General

Protocol implementation

IEC60870-6 protocol, also known as ICCP (Intercontrol Center Communication protocol) implementation of SYS600 follows IEC60870-6 TASE.2 standards. It is used to share information between control centers. Both client and server modes are supported and blocks 1, 2, 4 and 5 can be configured in both directions. Protocol is called later as ICCP in this document.

Block 1	Periodic System Data
Block 2	Extended Data Set Condition Monitoring
Block 4	Information Messages
Block 5	Device Control

See chapter 'Technical Description' for a full set blocks used in ICCP. When ICCP acts as server in SYS600, it is operating together with COM500*i* application. COM500*i* signal routing forwards the process data and incoming control commands according to the cross-reference information configured for COM500*i*.

When ICCP acts as client in SYS600, it receives process data and sends control common similarly to any other master protocol in SYS600. The STA object type and data base type both in client and server is IEC (STY 29). The protocol uses string-based data identification similarly to IEC61850.

ICCP implementation is a separate executable which connects itself to the LAN Link of base system using CPI Interface. The executable utilizes ICCP and MMS libraries from SISCO Inc. (Systems Integration Specialists Company, Inc) and the configuration is partially made using 'ICCP Configuration Utility' application from SISCO Inc.. The usage of ICCP requires the installation of the ICCP Configuration utility from SISCO Inc.. Installed package includes also tools for diagnostics. ICCP Configurator is the default configuration tool for making ICCP configuration for client and server. It is started from SYS600 Tool Launcher, Application Configuration tab.

Chapter 'Technical Description' depicts the protocol background, software layers and used components in more detailed level. Furthermore, data communication attributes used in runtime are described.

Requirements

The usage of ICCP in SYS600 requires:

- SYS600 10.2 or newer
- SYS600 license for ICCP
- ICCP Configuration utility from SISCO Inc., delivered with the SYS600 package
- COM500*i* license if ICCP server functionality is used

Installation

ICCP for SYS600 is installed as part of SYS600 10.2 or newer. Install SYS600 package according to the instructions in SYS600 Installation and Administration manual.

After SYS600 Installation the ICCP Configuration utility package from SISCO Inc. need to be installed by executing the setup.exe file from directory \sc\prog\iccp\utility. All components should be selected and installed. Note that some of the component may require rebooting of the computer and setup.exe must be started again after reboot, otherwise the installation is not complete.

When the installation directory asked, it is recommended to use the default directory proposed by the setup.



Do not install the ICCP Configuration utility package under the directory \sc which is reserved for SYS600.

Note!

After a successful installation, program group SISCO is created under 'Start' icon of Windows. Create a short-cut for ICCP Configuration Utility to the desktop. When tool is started, it may give a pop-up indicating inadequate privileges, but this does not prevent the tool usage. If ICCP Configuration Utility is run as administrator, the pop-up is not shown.

Chapter ICCP Configuration Utility describes the needed steps to complete the configuration.



ICCP Configuration cannot be completed if the package containing ICCP Configuration utility is not installed.

Note!

4.2 Configuration

General

The configuration can be divided into following parts:

- Base system configuration
- Communication configuration including COM500*i* configuration (only with server)

In communication configuration of ICCP, the process object names are visible since they mapped to the ICCP data value names. This is different from other protocols supported by SYS600. When ICCP is operating as client, addressing with process objects is string-based which means that the process object attribute IN (Item Name) must match with ICCP Name in the configuration. String address is applied to both incoming process data and outgoing commands. When ICCP is operating as server, numerical identifiers are used between ICCP_Link.exe and COM500*i* application.



Mirroring or routing the process objects updated by ICCP Client cannot be made to versions older than 10.2 using string based addressing. A workaround solution is to use manually created xxx_C.ini mapping file. See attribute station attribute RN for more information

Note!

The overall configuration sequence is as follows:

1. Basesystem configuration, NOD:B and STA:B objects. This is the basic configuration before ICCP_Link.exe may function. In ICCP Configurator, one project for each NOD:B object.
2. NCC creation to COM500*i*. Type ICCP, enter also the STA:B object number created for the server functionality.
3. Process object and COM500*i* configuration. All indications and commands which should be received or transmitted, are defined in this phase. In ICCP Configurator, project saving and opening is possible and it does not affect COM500*i* nor process object configuration.
4. Writing Configuration. COM500*i* and process object configuration is updated as well as the XML file for the executable.
5. Configuration finalization using ICCP Configuration Utility. This phase is necessary because the configuration tool is not supporting the full chain of configuration. Before the first import, localAR and all configured remoteAR names must be defined in the 'Addressing' section of the SISCO Configuration Utility. IP-addresses and other parameters of the related systems are listed here. In case of the client configuration, data set definitions are made in Configuration Utility and the XML for executable need to be exported before communication can be started.

Steps 1-2 are described in chapter 'Base system configuration' and steps 3 and 4 in detail in the configuration tool specific chapter. Detailed information for step 5 is described in chapter 'ICCP Configuration Utility'. Following picture shows the components used during configuration:

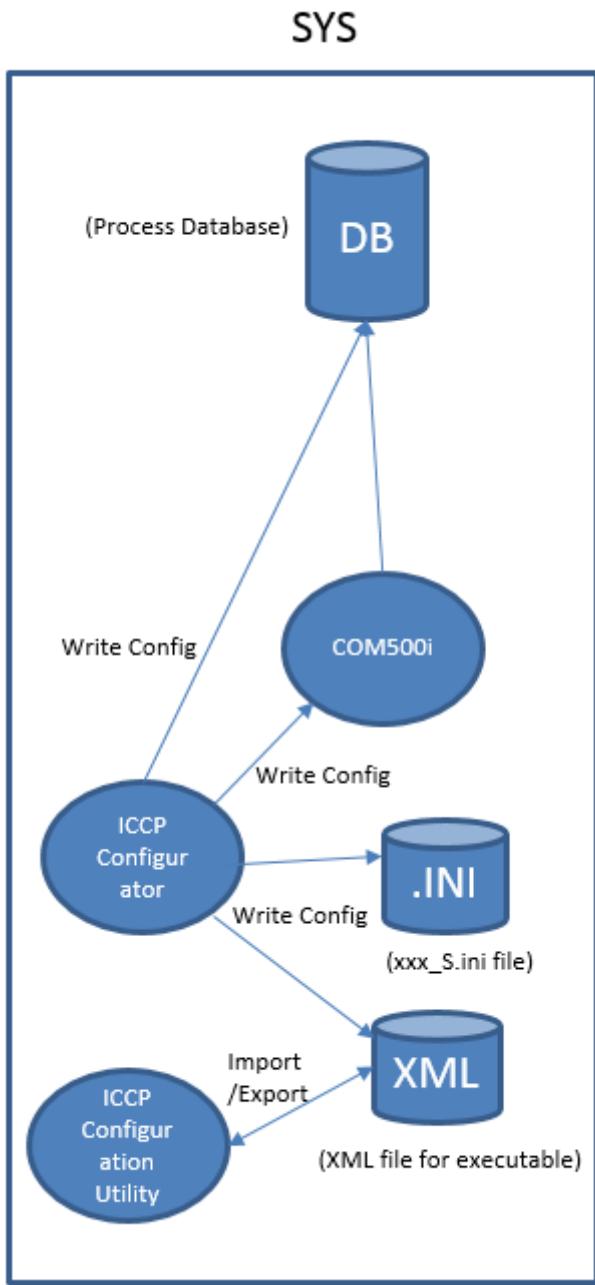


Figure 4.1: Component in configuration

Data flow in runtime is described in chapter 'Technical Description, Data flow'.

4.2.1 Base system configuration

ICCP configuration in base system requires:

- Node definition for each ICCP instance
- STA:B definitions, type IEC
- Process database

Node definitions and STA definition or NCCs to COM500i are **not** created in configuration import from ICCP Configurator.



The base system objects need to be created before protocol and process object configuration. For the node and STA:B creation, it is recommended to use System Configuration Tool, see SYS600 System Configuration manual for details.

Currently the instance count of ICCP_Link.exe in one system is limited to 1. This means that one node should be created and necessary amount of STA:B objects under the node. The amount of STA:B object depends on used ICCP configuration but the basic rules are as follows:

- If operating as server, one STA:B object as an interface to COM500*i*. This STA object will be the interface to RCCs operating as client.
- If operating as client, one STA:B object for each RCC the data is received from

When configured using System Configuration Tool, SSS (System Self Supervision) process objects are created when enabled. This functionality is like in other protocols, see SYS600 System Configuration manual for details.

See SYS600 Communication Gateway, COM500*i* manual for details how to configure NCCs of type ICCP. NCC creation is needed only when ICCP is operating as server.

Node definition

It is recommended to create the permanent node definition for ICCP using system configuration tool. Maximum value for the ICCP Node number is 55.

The node created using this definition uses LAN Link of the system. The usage of ICCP requires hardening setting REQUIRE_ENCRYPTED_ACP = "NETWORK" in SYS_BASCON.COM, setting "ALL" cannot be used. If used SYS_BASCON.COM is originated from older SYS600 version and setting REQUIRE_ENCRYPTED_ACP is not present for LAN Link, it need not to be set because of ICCP.

In Hot-Standby systems it is recommended to use same node definitions in both systems. This simplifies the used command procedures and system configuration in general.

In case communication modules are accessed by adjacent systems in run-time or there are other reasons to use different node numbers in communication modules in a Hot-Standby pair, it is possible. In this case, all node number must be explicitly listed in SYS_BASCON.COM or created in run-time using SCIL scripts. This approach is used e.g. in SAS reference application. See STA:B definition chapter for more details of this configuration.

If node is created using SYS_BASCON.COM template, it needs to be added to GW_NOD_Numbers list :

```
;  
;Gateway Nodes or nodes of the remote communication units  
;  
#local GW_NOD_Numbers = vector(15),- ;for example, vector(20,22),-  
GW_NOD_Addresses = vector(215)
```

In case nodes are defined using SCIL scripts instead of using SYS_BASCON.COM template, node creation can be made followingly:

```
;create node  
#CREATE NOD15:B = list(SA = 215, LI = 1, NN = "NCC1", CX = "ICCP Server 1")
```

The number of instances are limited by license.

STA:B definitions

If System Configuration Tool is used to create node object for ICCP, the STA definitions are added under it.

Recommendations for STA object creation:

- Max. one STA:B operating as server in each node. If instance operates only as client, no STA object for server is needed
- In client mode, one STA for each RCC, also when the remote RCC is redundant

The STA number assigned for server functionality need to be entered to the COM500*i* configuration as a station object for NCC. Please note that in ICCP, one NCC definition may provide the same data to several RCCs. Station type must always be "IEC". Station numbers must match with 'Client STA'/'Server STA' numbers in ICCP Configurator.

UN attributes of the process objects must follow the STA numbers operating as client.

In case system configuration tool is not used, permanent STA definitions for ICCP may be by adding the STA numbers to the SYS_BASCON.COM template. Note that the node number must be equal to nodes created in previous chapter:

```
;
;Host stations are typically defined with the PC-NET system configuration tool.
;Image stations, gateway stations etc. can be defined here.
;
; Server ClientRCC1
#local Stations = ( 101, 102 )
#local Sta_ST = ( "IEC", "IEC" )
#local Sta_Nodes = ( 15, 15 )
#local Sta_MR = ( "NONE", "NONE" )
#local Sta_H_Apl = ( 0, 0 )
#local Sta_H_UN = ( 0, 0 )
#local Sta_I_Apl = ( 0, 0 )
#local Sta_I_UN = ( 0, 0 )
```

Process Objects

Process objects for data received from ICCP (client mode) can be created using tools used in IEC60870-5-101/104 configuration, e.g. Object Navigator with SA-LIB standard functions or Excel-based LBETool.

ICCP Configurator is used import the created process objects and assign the string base ICCP data value names for each of them. In SYS600 10.2 or newer, the addressing in client mode may be based on ICCP data value name. In this case, the IN attribute of the process object should equal with the ICCP data value name in question.

Process objects for data provided using ICCP (server mode) are created using Object Navigator with SA-LIB standard functions or using protocol specific tools such as IET data loader, SCL importer etc.

In ICCP, the negotiation of the ICCP data value names and their meaning with the remote partner is the most essential part of the configuration process. The amount of configuration work is affected by the order of the necessary tasks. Following rules are worth to consider in order to minimize the the workload:

- When operating as server, propose the namespace for ICCP based on existing process database in SYS600. A filtered database export is needed to form a base for data provided from ICCP.
- When operating as client, expect the namespace for ICCP that is based on the data provided from the remote partner. Database creation is based on functionality in RCC.

If process database creation in client mode has been made but ICCP namespace has not been defined, a namespace proposal can be given based on existing process object names, similarly to server mode. A filtered database export is needed.

ICCP name may contain letters 'a'-'z', 'A'-'Z', numbers '0'-'9' and only special characters '\$' and '_'. The maximum length of the name is 32 characters.

4.2.2 Communication system configuration

General

The main purpose of ICCP Configurator is used to define data mapping between Data Value and Device names of ICCP and process objects and define how the configured items are used in RCCs both in client and server configurations. See following chapters for detailed instructions.

4.2.2.1 ICCP Configurator

When ICCP Configurator is started for the first time, it requires a creation of a new project. Each project created with ICCP Configurator may configure max. one NCC of type ICCP and a client connection to several RCCs. Furthermore, each project configures the functionality of exactly one instance of ICCP_Link.exe. If the system needs not to provide data using ICCP, server configuration and NCCs of type ICCP need not be created. In addition to sent

and received data, ICCP Configurator is used to define the connection between SYS600 base system and ICCP_Link.exe instance.

After the configuration, the generated XML file must be imported to ICCP Configuration utility and update with other necessary information such as data sets (client). The tree structure in ICCP Configurator is similar to ICCP Configuration utility presented in chapter ICCP Configuration Utility, the prerequisites for successful importing are described in the same chapter.

Based on given configuration parameters, ICCP Configurator generates a command procedure for starting the instance with correct configuration. This procedure is written to a separate .SCL file and should be called from APL_INIT_1:C of the main application. See more information from chapter Starting and stopping the communication. Following picture and table describes the main functions within the tool.

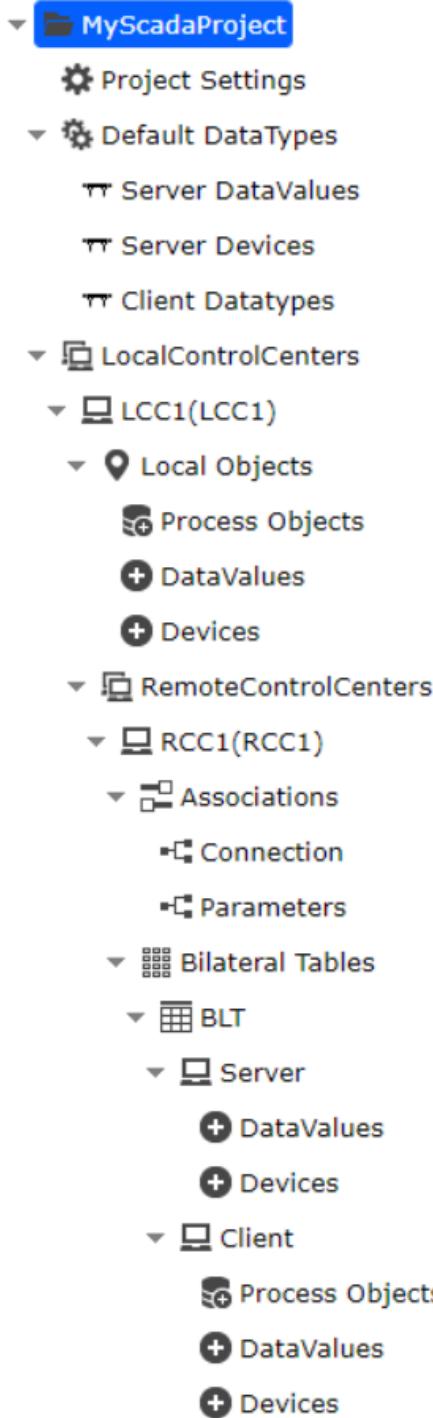


Figure 4.2: Tree structure of ICCP Configurator

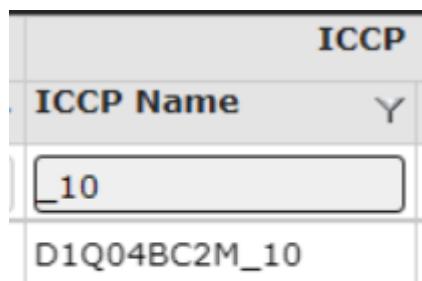
Project	
- Project settings	
Default DataTypes	
- Server DataValues	Default datatypes for DataValues defined for ICCP server.
- Server Devices	Default datatypes for Devices defined for ICCP server.
- Client Datatypes	Default datatypes for defined for ICCP client.
LocalControlCenters	Tables below this selection defines the local functionality of the ICCP.
- LCC	Parameters for LCC, i.e. ICCP operating as a server. Server STA object number 'server_sta' is defined here
-- Process objects	Table is used to filter and select the indication and command process process objects which are provided to remote control centers. If process object is enabled for ICCP communication, it will be added DataValues (outgoing indications) or Devices (incoming commands) for detailed definitions, such as COM500 <i>i</i> .
-- DataValues	Table defines the data items provided to RCC(s) and their mapping to SYS600 process objects. COM500 <i>i</i> definitions given here control how the data will be sent using COM500 <i>i</i> .
-- Devices	Table defines the controllable devices provided to RCC(s) and their mapping to SYS600 process objects. COM500 <i>i</i> definitions given here control how the received command will be processed using COM500 <i>i</i> .
- Remote control centers	Tables below this selection defines the functionality of the ICCP operating as a server or client. A list of created RCCs is shown. Server DataValues and Server Devices of each RCC are subsets of data items defined in LCC.
-- RCCx	Table defines the name and version the created RCC (x). Each remote control center is seen as separate RCC. With client, each RCC is mapped to specific STA object.
--- Associations	Association parameters for RCC in question. Parameters below this item can be set using SISCO Configuration Utility, too. Only one association / RCC is supported now.
---- Connection	Defines whether role ICCP with RCC is client, server or both. Defines also the association names.
---- Parameters	Defines the configuration parameters of the association.

- - - Bilateral tables	Table of bilateral table used with the RCC in question. Both client and server data belong to same BLT and may be present in the same configuration. In case client or server functionality is not used, corresponding client DataValues and Devices tables are empty. Only one bilateral table / RCC is supported now.
- - - BLT	Parameters of one (x) Bilateral table.
- - - Server	Tables below this selection shows the cross-referenced indications and command process objects cross referenced for this RCC. Server STA object which is common for all (server) RCCs is displayed here as a reference. In case server functionality is not used, tables below should be empty.
- - - DataValues	Table shows the indication data items selected for the RCC in question. Editing in this level is not possible.
- - - Devices	Table shows the controllable data items selected for the RCC in question. Editign in this level is not possible.
- - - Client	Tables below this selection need not to be filled if server functionality is not used. Client STA number 'client_sta' given in this dialog defines the UN number of the process object to be used with this RCC.
- - - Process objects	Table is used to filter and select the indication and command process process objects which are updated and used with the remote control center in question. If process object is enabled for ICCP communication, it will be added DataValues (incoming indications) or Devices (outgoing commands) for detailed definition.
- - - DataValues	Table defines the data items received from RCC and their mapping to SYS600 process objects. Furthermore, ICCP settings for data reception is defined here. Indications selected in 'Process objects' table should be visible here.
- - - Devices	Tables in this sheet defines the controllable devices available in RCC and their mapping to SYS600 process objects. ICCP setting for command transmission is defined here. Command process objects selected in 'Process objects' table should be visible here.

A more detailed description of each table is described in following chapters.



In each dialog having columns, it is possible enter a filter which decreases the number of visible lines.



Short-cuts in the upper edge have following functions:

Short-cut	Function
	Project parameters and datavalue/device definitions are saved to a project specific file. Configuration for ICCP_Link.exe nor for COM500i is not generated. Application changes are not made either.
	Opens a selection dialog for a project.
	Removes the selected item from LCC (server) or from specific RCC (client). Short-cut is available only in datavalue/device specific tables.
	Adds a new RemoteControlCenter.
	Triggers reading of the CSV-files which contains list of process objects and their ICCP Names
	Configuration XML file is updated for ICCP_Link.exe and cross-references to defined NCC in COM500i are rewritten. This also updates the instance starting/stopping procedures for ICCP_Link.exe.
	Starts ICCP_Link.exe instance, but does not start communication. Communication starting requires execution of SCIL sentence #SET NETxx:SIU1=1
	Stops ICCP_Link.exe instance and communication configured to it.



'Write Configuration' rewrites the cross-references to defined NCC in COM500*i*. This means that the cross-references entered to the same NCC using e.g. COM500*i* will be deleted.

Note!

Project saving means that all parameters and selected process objects are stored to project specific files for later use. Saving will not update the configuration for ICCP_Link.exe nor for COM500*i*. Button 'Write Configuration' writes configuration for use, i.e. XML file for the ICCP_Link.exe is created and the COM500*i* configuration for the defined NCC is updated.

In case the configuration write is made for the first time or there is a need to change IP-addressing, associations or, data sets, reimporting the XML file to ICCP Configuration Utility is needed.

Project Settings

The basic configuration for ICCP is made using Project table in ICCP Configurator. Following table contains explanations for each parameter. If not otherwise specified, filling of all entries is needed to complete basic setup for ICCP in SYS600.

Parameter Name	Explanation
client_import_file	An excel CSV file which can be used to import definitions for client data values and devices.
node_number	Number of the base system node, i.e. NOD:B object for the ICCP_Link.exe
project_directory	Directory where project is saved.
project_name	Name of the project.
runtime_directory	Target directory containing the XML-file for the ICCP_Link.exe. This directory is given as parameter when the instance of ICCP_Link.exe is started. XML-file for the ICCP_Link.exe is updated in this directory when 'Write Configuration' button is pressed.
server_import_file	An excel CSV file which can be used to import definitions for server data values and devices.

Default Datatypes

Tables below this selection defines the default ICCP data types and signal handling values for Server Data Values, Server Devices and Client as a whole.

It is recommended to fill these tables before item definitions to mentioned tables are done. In most cases, making these definitions require knowledge of the intended configuration in the remote system. If a signal list is available and it contains definitions for e.g. ICCP indication types, it is worth to add the most common setting of each data type to Server DataValues table and save the project. When signals are added after this, less engineering work is needed and possibility of a human error is smaller. The same approach is applicable with Server Devices and Client Data Types.

Common parameters in the Default Datatypes dialog are:

Parameter Name	Explanation
client_device_sbo	Defines if the ICCP client uses Select-Operate or Direct Operate (column SBO) in its commands by default. Y = Select is used, N = Direct is used.
normal_source	Defines the default data source (column Normal Source) for the selected items in ICCP server. Normally 'Telemetered'
read_only	Defines the default Read-Only definition for the selected Data Value items in ICCP server. Normally 'Y' (Only Read is allowed)
scope	Possible value are VCC/ICC. This value must match with the definitions in the remote system.
select_time	Defines the default maximum time for pending select (also called armed in ICCP) in server devices.

Parameter Name	Explanation
tag_enabled	Defines if tagging is enabled by default in server devices. Y=Tagging is enabled, N= Tagging is not enabled
version	Defines the default ICCP version for the created RCC. This value must match with the definitions in the remote system. Possible values are "1996-08" and "2000-08"

Server Data Values

Table defines the default ICCP Data Type and COM500*i* signal handling of each data item selected to be provided to RCC in Local Data Values / Process Objects, based on process object type of the selected item.

SYS600 DataType	Indication Type	Signal Handling
AI	RealQTimeTagExtended	None
BI	StateQTimeTagExtended	Send as Inverse Value
DB	StateQTimeTagExtended	None
DI	DiscreteQTimeTagExtended	None
PC	DiscreteQTimeTagExtended	None

When e.g a SYS600 Process object of type Binary Input (BI) is selected to be proveded for RCCs, it will get default ICCP Type 'StateQTimeTagExtended' and COM500*i* will get default signal handling 'Send as Inverse value'. These settings can still be changed separately for each selected item.

Server Devices

This table is used to define which defaults are given to selected control commands and setpoint commands received from RCC. Selection is based on Control Type, process object type and/or its index based on rules described below. Following default values are editable:

- ICCP Type : this is the default data type used in ICCP Communication
- Sbo : Defines if Select-before-operate is used or not in ICCP Communication
- Signal Handling : Defines the default signal handling

Column 'Control Type' is compared with the Control Type created by SA-LIB and stored to CD Attribute of the process object group. If e.g. Control Type "DIR-2BO" is defined for selected output process object and process objects with indices 11 and 12 exist, ICCP Configurator will bind these together and creates one ICCP Item which is seen as a controllable device in ICCP client. An empty field in ICCP Type columns of the second DIR-2BO line (index 12) is an indication that no separate ICCP Item should be created for it.

By default, all control types (except SEC-2DO) which has more than 1 output process objects in the process side, the ICCP Type of second, third, fourth and fifth is empty. This means that by default, commands are received using single ICCP item and used control model is not visible for the ICCP client.

Control Type	IX	Iccp Type	Sbo	Command Type	Command Purpose	Command Purpose Text	Signal Handling
Filter	Fl	Filter	Filter	Filter	Filter	Filter	Filter
AO	13	Real	N	Direct cmd with 1 AO			None
AO	0	Real	N	Direct cmd with 1 AO			Project Specific
BO	0	Command	Y	Secured cmd with 1 BO			None
DIR-2BO	11	Command	N	Direct cmd with 2 BOs	1	Open Command	None
DIR-2BO	12		N	Direct cmd with 2 BOs	2	Close Command	None

In case the empty field is filled with ICCP Type, e.g. like this,

DIR-2BO	11	Command	N	Direct cmd with 2 BOs	1	Open Command
DIR-2BO	12	Comm... ▾	N	Direct cmd with 2 BOs	2	Close Command

ICCP Configurator will create a separate ICCP Item for the output point with index 12, when a output process object having a 'DIR-2BO' control model is selected for the RCC.

In general, it is recommended to fill the ICCP Type fields of "not first" items only in very special cases. Handling of the incoming command may require project specific modifications to COM500*i* signal routing procedures, depending on the item usage logic in the ICCP client.

In case the selected process object does not have control model, or it is not found from 'Control Type' column, ICCP Configurator will select the default ICCP Type and COM500*i* signal handling according to process object type and index. In the picture above:

- If the selected process object is of type Analog Output and its index is 13, default ICCP Type will be 'Real' and default Signal handling is 'None'
- If the selected process object is of type Analog Output and its index is **not** 13, default ICCP Type will be 'Real' and default Signal handling is 'Project Specific'

This means that the definitions in line containing IX=0 are applied, if the only matching property is the SYS600 Process Object Type: AO = Analog Output, BO = Binary Output, DO = Digital Output.



IX=0 definition defines the default properties of selected process objects having IX values not explicitly listed

Note!

Supported control types are:

Control Type	Command Type
DIR-2BO	Direct command with 2 BOs
DIR-BO	Direct command with 1 BO
SEC-2BO	Secured command with 2 BOs
SEC-2DO (Selection in SA-LIB is 'Direct cmd with 2DOs')	Secured command with 2 DOs (is mapped to two ICCP Data Items by default)
SEC-4BO	Secured command with 4 BOs
SEC-5AO	Secured command with 5 AOs
SEC-5BO	Secured command with 5 BOs
SEC-BO	Secured command with 1 BO

Defining the ICCP Type, Select-before-operate and COM500*i* signal handling is possible specifically for each item selected to be sent to RCCs.

Client Data Types

Table defines the default ICCP Data Type each data item selected to be received/transmitted from/to RCC in Client / Process Objects, based on process object type of the selected item. Same table defines both input and output objects.

SYS600 DataType	Iccp Type
AI	RealQTimeTagExtended
AO	Real
BI	StateQTimeTagExtended
BO	Command
DB	StateQTimeTagExtended
DI	DiscreteQTimeTagExtended
DO	Discrete
PC	DiscreteQTimeTagExtended

When e.g a SYS600 Process object of type Binary Input (BI) is selected to be received from RCCs, it will get default ICCP Type 'StateQTimeTagExtended'. ICCP type can still be changed separately for each selected item.

Local Control Center

Process Objects

This table is the main dialog for selecting the process objects provided to any remote control center when ICCP is operating as server. At upper edge of the dialog, a temporary filter is given to list potential process objects to be cross-referenced to any of the RCCs. At right end of each line, in column 'RCC', selecting the cell and clicking left button of the mouse, an RCC selection menu is opened and RCC can be selected. In the picture below, process objects matching to filter OI=="East*" is listed and selected for RCC1. Furthermore, the ICCP name of the data item has been changed from its default to 'EastwickBC_Q0_State'.

Thus, the basic sequence for data item configuration for server is:

1. Enter filter which covers the undefined items as well as possible. The syntax of the filter is similar to one used in Object Navigator tool. In case items which already have RCC definition is displayed, it means that the selection has already been done using some other filter.

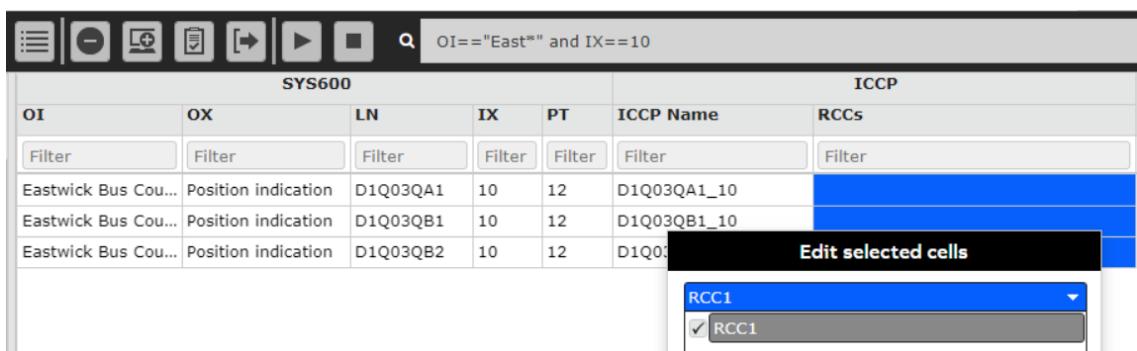


Figure 4.3: Server data selection

2. Select items to RCCs as needed using the selection in 'RCCs' column. If the selected process object is an indication, it is added to DataValues table. Similarly, if the selected process object is an output process object or group of those, those are added to Devices table. If displayed item is not needed in any of the RCCs, it should not be selected and it will not be added.
3. Enter the agreed ICCP Name to column 'ICCP Name' and enter other definitions such as Data Type in ICCP and signal handling for COM500i. The default name is derived from process object LN and IX. Note that if multiple RCCs is used, same name and settings are used with all RCCs.

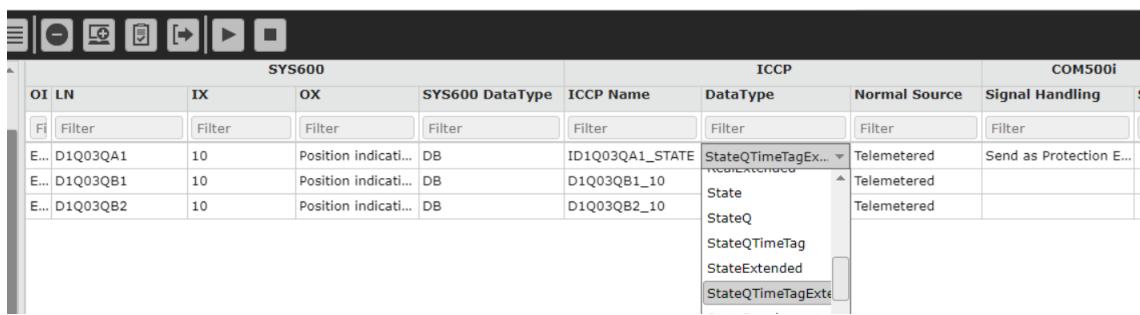


Figure 4.4: Server data definition

Repeat steps 1-3 until all necessary indications and command has RCC definitions and are listed in DataValues and Devices tables. Saving of the project is recommended time to time. If the amount of selected items in step 2 is big or the configuration itself is big, tool does not react to the pressing of 'Ok' button quickly. The processing is on-going despite of this.



The maximum amount of cross-referenced items in COM500*i* is 50000, this is also the maximum amount of items configured for ICCP operating as server

Note!



When signal list is defined, proposing an ICCP name which is directly derived from process object name and index may speed up engineering and reduce typos which prevent communication establishment with the remote RCC

If given filter is not matching to any process objects, displayed list is empty. If given filter matches to 10000 or more process object, a warning is displayed that 10000 first ones are displayed. In this case it is recommended to enter another filter which has less matches.

Detailed definitions, such as COM500*i* and data type in ICCP transmission can be made in tables DataValues and Devices not until the RCC selection is made. These definitions for one item are the same for all RCCs receiving the data.

DataValues

Sheet **DataValues** defines the data values given to remote RCCs. Each cross-referenced data point from process database is listed here and corresponding ICCP name should be given here, with the naming rules explained at beginning of this chapter. Additionally, the data type for each data item need to be defined and possible signal handling properties for Com500*i*.

The properties of the data items given to RCC are (**bolded** values can be edited by selecting the cell and entering a value or making a selection using left mouse button):

- **OI Object Identifier** (in process database)
- **LN Logical name** (in process database)
- **IX Index** (in process database)
- **OX Object Text** (in process database)
- **SYS600 Datatype** (in process database)
- **Name (ICCP Name)**, name used in transmission (consider naming rules)
- **DataType**, data type in ICCP communication level
- **Normal Source**, data source in ICCP communication level
- **COM500 signal handling** (in COM500*i* configuration)
- **COM500 scale** (in COM500*i* configuration)

When process object is selected for a RCC, DataType and COM500 signal handling is set according to the default data types defined in table under tree selection Default DataType -> Indications (DataValue) stored for the project. Correct values in this table speed up the definition work significantly.

Devices

Sheet **Devices** defines the controllable devices provided to the remote system. Commands from RCC are received using ICCP names defined here and the redirected to local process devices using the cross reference and signal handling information defined for each point.

The properties of the controllable devices given to RCC are (**bolded** values can be edited by selecting the cell and entering a value or making a selection using left mouse button):

- OI Object Identifier (in process database)
- LN Logical name (in process database)
- IX Index (in process database)
- OX Object Text (in process database)
- SYS600 Datatype (in process database)
- **Name (ICCP Name)**, name used in transmission (consider naming rules)
- **Device Type**, data type in ICCP communication level
- **SBO**, indicates if the device uses 'select before operate' in ICCP communication level. Should match with remote configuration.
- **Chk Back Id**, used with SBO devices only, identifier used in ICCP communication level to assure that correct device is selected. Should match with remote configuration.
- **Sel Time**, maximum time for keeping the device selected in server.
- **Tag En**, indicates that the device can be tagged, i.e its operation can be prevented or limited from the remote system.
- **Command purpose** (in COM500*i* configuration)
- **Command type** (in COM500*i* configuration)
- **Command signal handling** (in COM500*i* configuration)

When process object is selected for RCC, DataType and COM500 signal handling is set according to the default data types defined in table under tree selection Default DataType -> Indications (DataValue) stored for the project. Correct values in this table speed up the definition work significantly.

Remote Control Centers - RCC

RCCs are listed as RCC1, RCC2..., but in each dialog, RCC name can be given. Given name defines the RCC name in the ICCP Configuration utility and it must match with corresponding configuration in the remote end.

Version string given is stored as such to the XML. It is recommended to keep this value in its default.

New RCC is added using button 'Add New RemoteControlCenter'.

Remote Control Centers - Associations

Associations dialog is divided to two tables of parameters, Connections and Parameters.

Connections table lists the local AR (Application Reference Name) and remote ARs. Local AR is the name of the local system, usually containing a company name, too. Remote ARs are basically similar, any string which identifies the remote system in local configuration is applicable. These values need not to match with remote configuration.

In case the local system is HSB, same local AR name will be used in both systems. In case the remote system is HSB, 'remote_ar' and 'remote_ar_2' need to be filled.

Table 4.1: Connection

Parameter Name	Explanation
local_ar	Name of the local system, e.g. 'SYS600_SCADA' or 'SiteA'
remote_ar	Name of the remote system 1, e.g. 'ICCO_Srv1' or 'SiteB1'
remote_ar_2	Name of the remote system 2, e.g. 'ICCO_Srv2' or 'SiteB2'
remote_ar_3	Name of the remote system 3, e.g. 'ICCO_BackupSrv1'
remote_ar_4	Name of the remote system 4, e.g. 'ICCO_BackupSrv2'

Parameter Name	Explanation
role_description	Role of the system. Use 'Client & Server : Calling' if it has been agreed that this system opens the connection. Use 'Client & Server : Called' if it has been agreed that remote system opens the connection. Note that if the local system is "client only" in terms of ICCP, it may yet be a server in terms of TCP/IP. And vice versa. The usage of selections 'Client : Calling' and 'Server : Called' are not recommended as primary setting since they have caused compatibility problems with some vendors.

'Parameters' dialog defines common parameters to be used with this RCC.

Table 4.2: Parameter

Parameter Name	Explanation
assoc_heartbeat_time	Interval of the MMS Identify message to verify the TCP/IP connection
assoc_retry_time	Waiting time before retrying a timeout Association request.
conclude_timeout	Response timeout for Conclude request sent to remote AR.
initiate_timeout	Response timeout for Initiate request sent to remote AR.
max_ind_pend	Maximum number of remote requests the server will allow
max_mms_msg_size	Maximum MMS Message size in bytes

Remote Control Centers - Bilateral Tables

Dialogs under 'Bilateral tables' defines the properties of bilateral tables between local control and RCC in question. Currently not more than one BLT is supported per RCC.

Table 4.3: BLT

Parameter Name	Explanation
blocks	Supported ICCP block. Usage of default values is recommended
id	Identifier of the bilateral table. This field must match with the corresponding id name configured to RCC.
local_domain	Local domain name. Default name is derived from LCC and RCC names. This field must match with the remote_domain name configured to RCC.
name	Name of the bilateral table
remote_domain	Remote domain name. Default name is derived from RCC and LCC names. This field must match with the local_domain name configured to RCC.
shortest_interval	This is the shortest periodic interval (in seconds) allowed for transfer sets.

Remote Control Centers - BLT - Server

Selections for the data items and devices provided to each RCC are defined in dialogs Local Control Center / DataValues and Local Control Center / Devices. Thus, dialogs under 'Remote Control Centers - BLT - Server' are for verification only.

Remote Control Centers - BLT- Client

Process Objects

This table is the main dialog for selecting the process objects provided to any remote control center when ICCP is operating as client. Similarly to corresponding server dialog, a temporary filter is given to the field at the upper edge of the dialog. This is used to list all potential process objects to be bound to certain RCC. At the right end of each line there is column 'Enable ICCP' which defines whether the data item in question should be updated or

controlled via ICCP. Setting the value of this field to 'true' enables it for ICCP reception or transmission. After the selection, the name of the data item should be updated to match with the agreed ICCP Name.

Process objects should be created with some other tool before the data item definitions can be done.

Basic sequence for data item configuration for client is:

- Enter filter which covers the undefined items as well as possible. The syntax of the filter is similar to one used in Object Navigator tool. If the process objects are fully configured, a good filter for selecting is "UN==x", where x is the STA objects reserved for the RCC in question, i.e. the same which has been entered to parameter client_sta in table 'BLT/Client' in the same RCC. In case items which already have ICCP Enabled = true is displayed, it means that the selection has already been done using some other filter.

SYS600							ICCP	
OI	OX	LN	IX	PT	ICCP Name	Enable ICCP		
ICCO South 1L1	Current 1	ICCO_STH_1L1_ME1	27	9	SOUTH_1L1_ME1	false	Edit selected cells	
ICCO South 1L1	Current 2	ICCO_STH_1L1_ME1	28	9	SOUTH_1L1_ME2	false		
ICCO South 1L2	Current 1	ICCO_STH_1L2_ME1	27	9	SOUTH_1L2_ME1	false		
ICCO South 1L2	Current 2	ICCO_STH_1L2_ME1	28	9	SOUTH_1L2_ME2	true		

Figure 4.5: Client data selection

- Select items to RCCs as needed by setting ICCP Enabled -> true for items needed. If the selected process object is an indication, it is added to DataValues table. Similarly, if the selected process object is an output process object or group of those, those are added to Devices table. If displayed item is not needed in any of the RCCs, it should not be selected, and it will not be added.
- Enter the agreed ICCP Name and DataType to corresponding columns. The default name is derived from process object LN and IX.

SYS600						ICCP		
OI	LN	IX	OX	SYS600 D...	ICCP Name	ICCP DataType	ReadOnly	
ICCO South 1L1	ICCO_STH_1L1_ME1	27		Current 1	AI	SOUTH_1L1_ME1	RealQTimeTagExten...	Y
ICCO South 1L1	ICCO_STH_1L1_ME1	28		Current 2	AI	SOUTH_1L1_ME2	Real	Y
ICCO South 1L2	ICCO_STH_1L2_ME1	27		Current 1	AI	SOUTH_1L2_ME1	RealQ	Y
ICCO South 1L2	ICCO_STH_1L2_ME1	28		Current 2	AI	SOUTH_1L2_ME2	RealQTimeTag	Y
							RealQTimeTagExtended	Y

Figure 4.6: Client data definition

Repeat steps 1-3 until all necessary indications and command has definition ICCP Enabled = true and are listed in DataValues and Devices tables. Saving of the project is recommended time to time.



Proposing an ICCP name which is directly derived from process object name and index may speed up engineering and reduce typos which prevent communication establishment with the remote RCC

If given filter is not matching to any process objects, displayed list is empty. If given filter matches to 10000 or more process object, a warning is displayed that 10000 first ones are displayed. In this case it is recommended to enter another filter which has less matches.

Detailed definitions, such as data type in ICCP transmission or properties of the controlled device of the remote system can be made in tables DataValues and Devices not until ICCP enabling is made.

DataValues

Sheet **DataValues** defines the data values received from remote systems. Process objects are of type IEC and they can be created with any tool. Addressing is based on process object attributes unit number (UN) and item name (IN), these should match with the STA object in ICCP configuration and the ICCP name.

The properties of the data items received from RCC are (**bolded** values can be edited by selecting the cell and entering a value or making a selection using left mouse button):

- OI Object Identifier (in process database)
- LN Logical name (in process database)
- IX Index (in process database)
- OX Object Text (in process database)
- SYS600 Datatype (in process database)
- **Name (ICCP Name)**, name used in transmission
- **ICCP DataType**, data type in ICCP communication level
- **ReadOnly**, data source in ICCP communication level

Devices

Sheet **Devices** defines the controllable devices available in the remote system. Commands to RCC are sent using ICCP names defined here. Similarly to data value for client, the attributes unit number (UN) and item name (IN) of the output process object should match with the STA object in ICCP configuration and the ICCP name.

The properties of the controllable devices available in RCC are (**bolded** values can be edited by selecting the cell and entering a value or making a selection using left mouse button):

- OI Object Identifier (in process database)
- LN Logical name (in process database)
- IX Index (in process database)
- OX Object Text (in process database)
- SYS600 Datatype (in process database)
- **ICCP Name**, name used in transmission
- **ICCP DataType**, data type in ICCP communication level
- **SBO**, indicates if the device uses 'select before operate' in ICCP communication level. Should match with remote configuration.
- **Chk Back Id**, used with SBO devices only, identifier used in ICCP communication level to assure that correct device is selected. Should match with remote configuration.
- **Tag En**, indicates that the device can be tagged, i.e its operation can be prevented or limited from the remote system.

Import from Excel

Importing the item lists from .csv files for Data Values (indications) and Devices (Command) can be made for both client and server. This is useful especially when the ICCP Names of the items are defined by the remote partner or if Excel has been used to define the properties of exchanged items.

Importing takes place when user presses button 'Read ICCP Names from CSV file'.



The minimum format of the .csv file is as follows.

Server:

```
LN;IX;ICCPName;RCCs
D1Q03QA1;10;EastwickBC_Q0_State;RCC1,RCC2
.
.
```

Client:

```
LN;IX;UN;ICCPName
D1Q03QA1;102;EastwickBC_Q0_State
```

Excel import is made at the same time for both client and server files. Therefore, it is recommended to:

- keep client CSV file empty when server process objects are imported
- keep server CSV file empty when client process objects are imported

Import operation can be made in blocks using filtering and/or replacing the CSV between import. Open dialog Local Control Center / Process Objects (server import) or Client / Process Objects (client import). In case not all objects listed in the CSV should not be imported at once, enter a filter which will define the set of imported objects. With an empty filter, all objects are imported. Furthermore, between imports it is possible to replace / redefine a new CSV file to be imported and / or redefine the default datatypes as needed. This may lead to e.g. following import sequence:

1. Define a valid server CSV file and empty client CSV file
2. Go to Local Control Center / Process Objects
3. Enter filter 'PT==9' (only Analog Input process objects are imported) and press import button. After this, Server DataValues list should be filled analog input process objects.
4. Go to 'Default Datatypes' and define default datatype for Binary Input be 'StateQTimeTagExtended' and Signal Handling 'None' (next imported block will use these defaults)
5. Go to Local Control Center / Process Objects and enter filter 'PT==3 and IX==10', press import button -> Binary inputs with process object IX==10 from CSV are imported and they will get ICCP Type 'StateQTimeTagExtended' and Signal Handling 'None' by default
6. Go to 'Default Datatypes' and define default datatype for Binary Input be 'StateQ' and Signal Handling 'Project Specific' (next imported block will use these defaults)
7. Go to Local Control Center / Process Objects and enter filter 'PT==3 and IX<>10', press import button -> Binary inputs with process object IX==10 from CSV are imported and they will get ICCP Type 'StateQ' and Signal Handling 'Project Specific' by default
8. Repeat steps with different filters and CSV files until all process objects are defined. Repeat the same with client using dialog Client / Process Object
9. Save project and/or make 'Write Configuration' as needed

Continue according to chapter 'Configuration completion ICCP Configurator'.

Configuration completion ICCP Configurator

Configuration in ICCP Configurator is completed when following steps are performed:

1. Base system configuration, i.e. nodes and STA objects are created (chapter 'Base System Configuration')
2. COM500i NCC is created (chapter 'Base System Configuration')
3. All RCCs are created and required names have been given (chapter 'Remote Control Center - RCC')
4. Local and remote AR names have been defined Associations of each RCC (chapter "Remote Control Center - RCC")
5. Remote and local domain names have been defined (chapter "Remote Control Center - Bilateral Tables")
6. Data Values and Devices provided to other RCCs has been defined, ICCP Names and Types given, signal handling defined (chapter 'Local Control Center/'Remote Control Center - BLT - Server')
7. Data Values and Devices in all RCCs has been defined, ICCP Names and Types given (chapter 'Remote Control Center - BLT - Client')
8. 'Project settings / runtime_directory' has been redefined if default directory is not used

When these steps are completed, configuration can be written using 'Write Configuration' button



Close COM500i Signal X-reference tool before pressing the button.

When configuration writing is successfully completed, following modification are made to the system:

- Project is saved
- COM500i cross-reference information is updated

- `xxx_S.ini` file is updated. This file contains the mapping of ICCP Names to identifiers seen COM500*i* Signal X-reference tool
- IN-attributes of the process objects listed in client Data Value and Devices are updated with the given ICCP Name
- XML-file for the `ICCP_Link.exe` is created to the directory defined in 'Project settings / runtime_directory'

The contents of the COM500*i* cross-reference information should be visible in COM500*i* cross-reference tool.

Note! Configuration writing time is directly proportional to the item count. With a maximum configuration having 50000 items configured for server, 'Write Configuration' may take 10-15 minutes, depending on computer resources. See manual SYS600 Communication Gateway, COM500*i* for signal count limit details for server. Configuring client items is about four times faster compared with server items.

XML-file for the `ICCP_Link.exe` requires further processing using SISCO ICCP Configuration Utility before communication be started. See chapter 'ICCP Configuration Utility' for details of this. The name of the XML-file is equal to `lcc_name` parameter under LocalControlCenter:

Parameter	Value
Filter	Filter
<code>lcc_name</code>	LCC1

In this case, `LCC1.xml` is created.

4.2.2.2 ICCP Configuration Utility

ICCP Configuration utility is used to define addresses for LCCs and RCCs, associations and data set definitions. The settings are described below. When configuration has been created using ICCP Configurator, the generated XML file for `ICCP_Link.exe` must be imported and parameter settings described in chapters Addressing, IP Listen settings and data set (client only) must be done. If configured local and remote ARs are not defined in 'Addressing' part, XML import will fail.

Note! XML Importing to ICCP Configuration utility must be made from the directory specified as 'runtime_directory'

Note!

In order to prepare the `ICCP_Link.exe` Instance for access from SYS600, the configuration of Addressing, Local Control Center, at least one Remote Control with one bilateral table is needed.

In case of client functionality, data sets must be created in ICCP Configuration utility using e.g. Auto DSTS or manually and the XML must be exported back for the executable. Auto DSTS is activated by clicking right mouse button above the tree item 'Client'.

In case of server-only functionality, it is enough to import the generated XML file to ICCP Configuration utility to make the configured LCC visible in the tree. Importing of the configuration will update the XML files used by the tool itself. Exporting XML for `ICCP_Link.exe` is not needed (as mentioned, this applies only to server-only functionality).

Note! Do not change the ICCP Names or types of the data items in the ICCP Configuration Utility. Furthermore, do not remove or add the items, otherwise the connection between the ICCP Name and corresponding process object configuration is lost.

Note!

Addressing

In 'Addressing' part of the configuration tree, the IP-addresses of all partners need to be entered, including the SYS600 computer itself. Following entry must be added

- an address entry containing a locally used IP-address and a AR name equal to Associations/Connection/local_ar in ICCP Configurator.
- an address entry containing a remote IP-address and a AR name equal to Associations/Connection/remote_ar in ICCP Configurator.
- if remote system is redundant, an address entry containing a secondary remote IP-address and a AR name equal to Associations/Connection/remote_ar2 in ICCP Configurator.

Example:

Connection	Value
Filter	Filter
local_ar	SiteA
remote_ar	SiteB1
remote_ar_2	SiteB2

Connections above will require definitions for "SiteA", "SiteB1", "SiteB2":

AR Names									
Basic	Selectors	All							
Name	IP Address	AP Title	AE Qual	A...	AE...	PSEL	SSEL	TSEL	D
SiteA	10.1.231.1	1 3 9999 2	2			00 00 00 05	00 01	00 01	
SiteB1	10.1.241.1	1 3 9999 3	3			00 00 00 06	00 01	00 01	
SiteB2	10.1.242.1	1 3 9999 3	3			00 00 00 06	00 01	00 01	
SiteC1	10.1.251.1	1 3 9999 4	4			00 00 00 07	00 01	00 01	
SiteC2	10.1.252.1	1 3 9999 4	4			00 00 00 07	00 01	00 01	
SiteD	10.1.202.1	1 3 9999 4	4			00 00 00 07	00 01	00 01	

All addresses need to be added to ICCP Configuration utility before importing the XML file generated when 'Write Configuration' is made. "SiteA" definition contain locally used IP-address and other MMS parameters. Furthermore, "SiteB1" and "SiteB2" entries refer to one remote pair of systems operating as ICCP server or client. In case the local system is redundant, a different IP-address is given to AR Name matching to local_ar in the parallel system but MMS-parameters may be the same. The given local AR name should be applicable for both SYS600 computers.

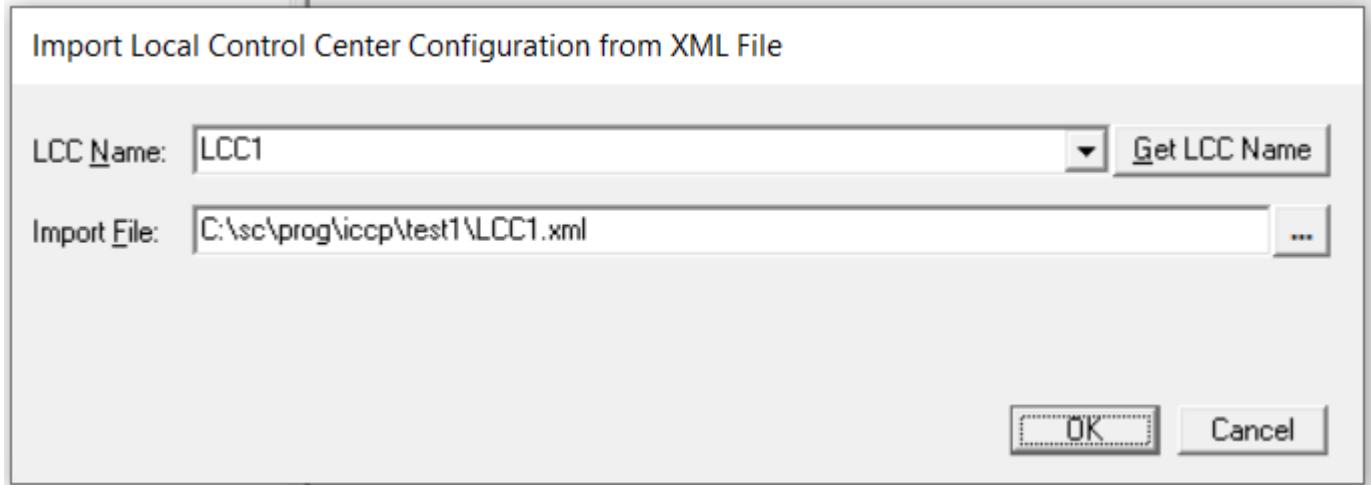
It is strongly recommended to configure all listed MMS parameters to be equal with the remote system system defined by AR Name and IP-Address. In case the remote system is redundant, same parameters are easiest to handle (see SiteB1/SiteB2 in the picture above). Using that approach, the MMS parameters of the local system can also be the same and the only difference between systems is the IP-address.

If the configuration contains multiple RCCs, all remote servers and their IP-addresses are listed to the 'Addressing' dialog of ICCP Configuration Utility. Same selector values can be used with multiple RCCs. The created AR Names for remotes are referenced in the association definition of each RCC in configuration.

ICCP_Link.exe does not utilize SNAP-Lite, no setting under IP-Listen setting is needed. In case of role_descritcion 'called' Listening address will be the address defined for local_ar.

Local Control Center

XML file importing is required in all cases, it is done using the File->Import from XML file selection from the menu bar. Note that the import must be done from the directory specified with Project Settings/runtime_directory of the project in ICCP Configurator. Import operation in the tool updates the internal configuration file \ProgramData\ISCO\Network\Config\siscostackcfg.xml. ICCP_Link.exe does utilizes this file directly automatically, if it is not updated with necessary information, communication does not start.



When XML file is imported, a tree having an equal name to imported XML is created, see figure below.

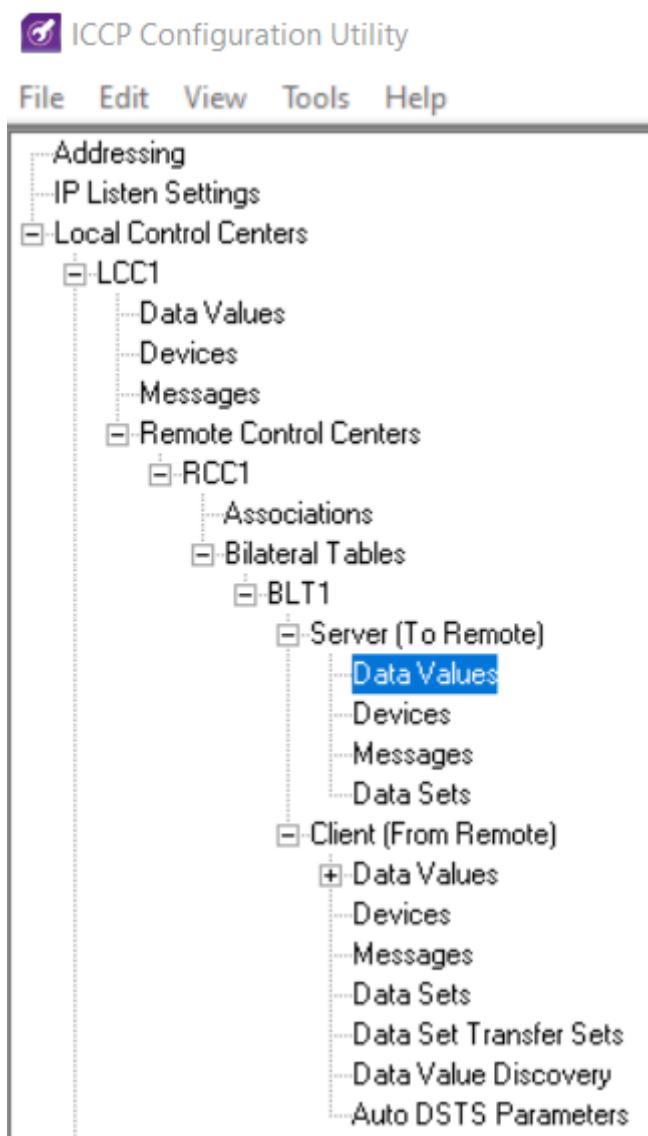


Figure 4.7: Tree structure after importing



When XML is re-imported, it is recommended to delete the previous tree structure having a same name before importing. Otherwise the import may fail, or the result is a mixture of new and old configuration.

All local control centers (LCCs) are listed here as tree structure.

Definitions under the created Local Control Center should be as configured earlier.

- Data Values :** A union of all data items, their types and sources provided from this LCC. Entries are added only if SYS600 ICCP functions as server. Table is updated by the ICCP Configuration tool in SYS600
- Device:** A union of controllable devices , their types and other properties provided in this LCC. Remote control send control commands to these and the received commands are forwarded to process units according to COM500i configuration. Entries are added only if SYS600 ICCP functions as server. Table is updated by the ICCP Configuration tool in SYS600.
- Messages :** A union of Info Message references. One reference can be used send different messages or files to remote control centers. If needed, creation of two info reference for each RCC can be considered, one for information message between control centers and one for file transfer.
- Remote Control Center :** the definition of the remote control centers which communicate with the LCC in question. Table is updated by the ICCP Configuration tool in SYS600.



Do not add or delete data items or redefine ICCP Names or type of those in ICCP Configuration utility, otherwise, the configuration is not consistent with the COM500i anymore

Note!

Remote Control Center

Each RCC created to project should visible as Remote Control Center under the created Local Control Center. If the RCC is redundant, one entry is still enough. The dialogs mentioned below for verification only, do not the change these setting ICCP Configuration Utility.



If redefinition for association or bilateral table is needed, make the changes in ICCP Configurator instead, not in ICCP Configuration Utility!

Note!

Definitions under the created Remote Control Center:

- Associations:** Association binds the LCC and RCC to each other and defines the used local and remote AR Names. In case the RCC functions as redundant, 'Remote AR' and 'Remote AR 2' need to be defined with names entered in 'Addressing' part. 'Local AR' should be the name of the local computer. At least one association is needed to transfer data between LCC and RCC. In association level it is also configured if the association is opened by the RCC (called) or locally (Calling). An agreement for this is needed with the remote partner. When AR names are correctly defined in ICCP Configurator, association dialog may look like this (SiteB1/SiteB2 functions as HSB):

Associations for RCC 'RCC1'				
Basic	Remote ARs	Timers	Parameters	All
Name	Service and Connection Roles	Remote AR	Remote AR 2	Re AR
PRIMARY	Client & Server: Calli...	SiteB1	SiteB2	<N

- Bilateral Tables:** Data Value, Devices and Information messages in both directions are defined under a bilateral table. The names of the tables are free but the 'Bilateral Table Id' need to be agreed with the remote partner. Same string can be used as name and ID. Furthermore, the contents of 'Local Domain' and 'Remote Domain' need to be agreed with the remote partner. A configured BLT with defined domain names may look like this:

Bilateral Tables for RCC 'RCC1'						
Basic	Blocks	All				
Name	Bilateral Table ID	Active	Local Domain	Remote Domain	Shortest Interval (sec)	Description
BLT1	BLT1	Yes	RCC1_LCC1	LCC1_RCC1	1	

Bilateral Table

The definitions under each bilateral table defines the Data Values, Devices and Info messages received and transmitted from/to this specific RCC using the bilateral table in question. At the moment, one bilateral table for a RCC may be active at a time.

ICCP Configurator binds the data definitions to SYS600 process objects. In case of the client functionality, the definitions of Auto DSTS or alternatively, Data Sets and Data Set Transfer Set, is needed. The usage of Auto DSTS is recommended.

The definitions in part 'Server (to Remote)' are similar to LCC level, the difference is that here the Data Value, Device and Info Message are defined for this RCC only. If not more than one RCC exists, the definitions are similar to LCC level. If the instance does not function as server, modifications under 'Server (To Remote)' is not needed.

The definitions in part 'Client (From Remote)' define the data coming from the remote, Devices accessible in the remote and Information Messages from remote. If the instance does not function as server, modifications under 'Server (To Remote)' is not needed. In addition to data definitions (Data Values, Device and Info Messages), the configuration of Data Sets and Data Set Transfer Set is necessary:

- Auto DSTS : If usage of is agreed with the remote, enable AutoDSTS creation and define the association on which it is used. The usage of 'Assign data values by data type' is recommended, too. This defines the data transfer properties separately for each data type and also defines that one data set (automatically created) contains only one data type. It is also possible to define if the data is sent spontaneously (RBE) and what is the interval of the integrity check. Setting 'Critical' means that a dataset update from remote is explicitly confirmed by the client. Auto DSTS is activated by clicking right mouse button above the tree item 'Client'.
- Data Sets : If data sets are created manually, a suitable naming convention and rules for Data Value division to data sets is needed. Selecting the Data Values for each data set is made using button 'Change'. The data set creation logic can be based on data types, transmission priority etc. If scope is set to 'ICC', this must be defined in ICCP Configurator as well LCC.
- Data Set Transfer Sets : These parameters define the functionality of each data set in detail.

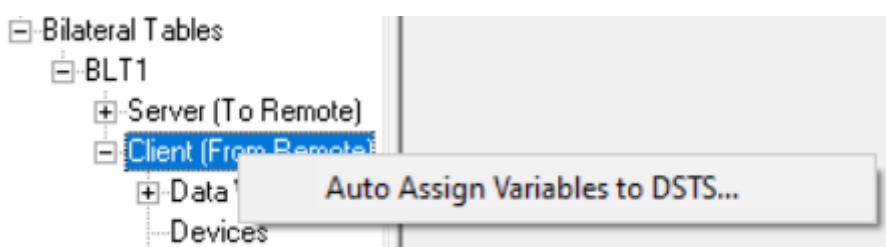


Figure 4.8: Auto DSTS activation

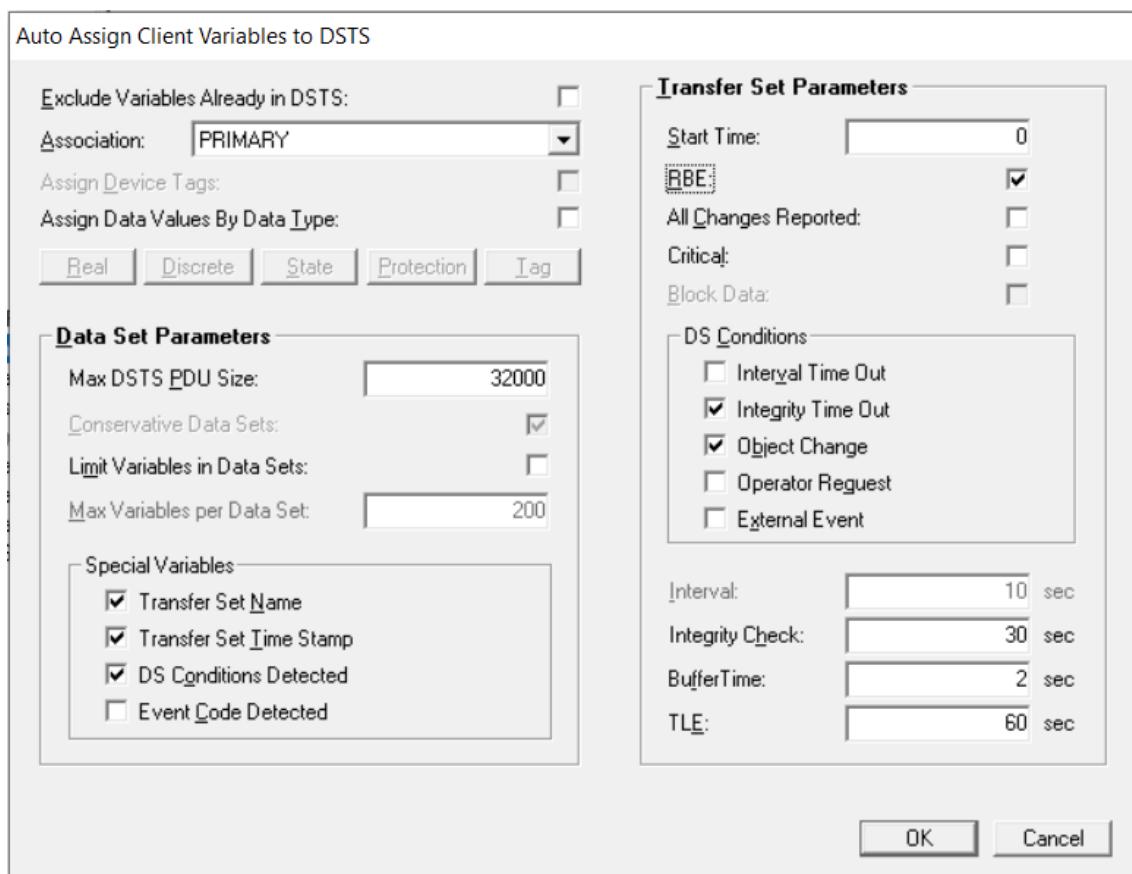
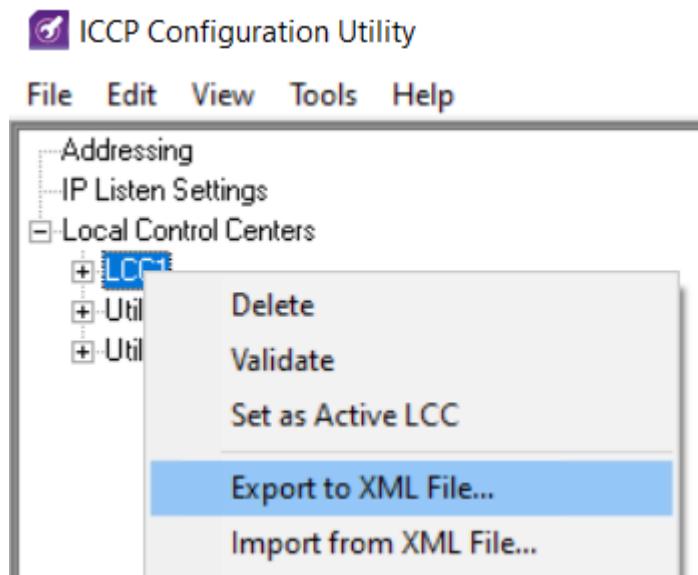
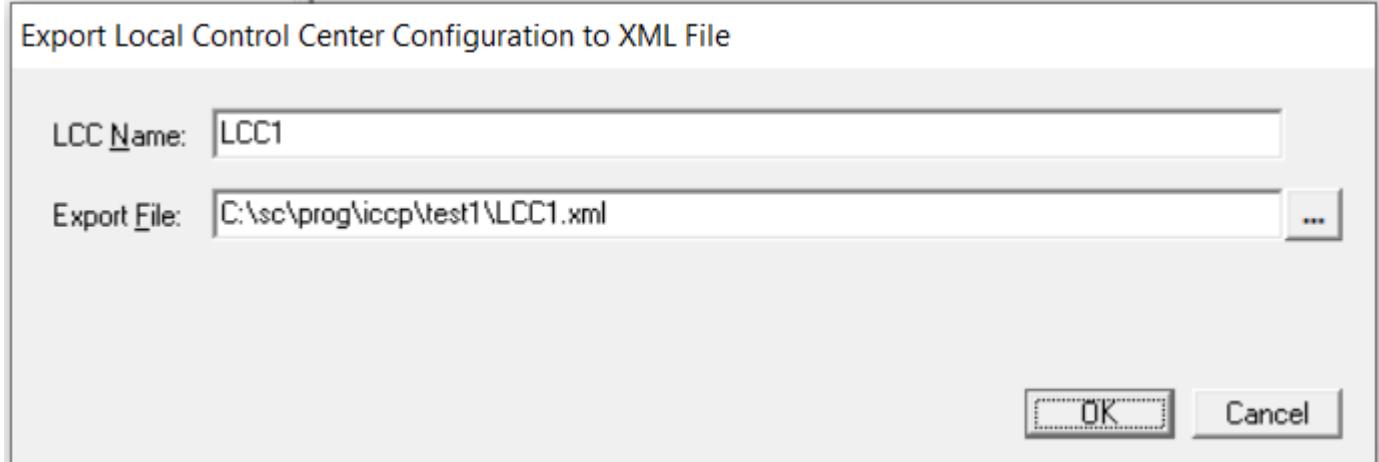


Figure 4.9: Data Set Definition

When all (Client) items has been defined to Data Sets in all RCCs, configuration can be exported XML file which is used by the executable. Export is activated by clicking the right mouse button above the used local control center:



The file to be exported is the same which was imported, i.e. the file will be overwritten with an additional info related to data sets.



See chapter 'Configuration Completion' for verification and next steps.

Configuration completion ICCP Configuration Utility

Configuration in ICCP Configuration Utility is completed when following steps are performed:

1. XML-file created during 'Write Configuration' in ICCP Configurator has been successfully imported (chapter ICCP Configuration Utility / Local Control Center)
2. In case of 'Server only' configuration, skip to step 5
3. All (client) data items are defined to data sets and the properties the data set are defined (chapter ICCP Configuration Utility / Bilateral Table)
4. XML-file is exported to directory defined in 'Project settings / runtime_directory'
5. System is ready for communication testing

If case configuration changes are needed, make the changes to the project in ICCP Configurator and repeat the steps as instructed above.

See chapter 'Starting and stopping the communication' for the next steps.

Secure Stack Configuration Utility

Secure Stack Configuration Utility is a separate configuration tool which is started from 'Tools/Secure Stack configuration Utility'. This tool can be used to set the conditions for MMS parameter verification of the remote system.

Tree selection Stack Configuration->Common Application Settings->AR Name Matching Settings allows user to disable checking of certain MMS parameters. If e.g. the settings of the remote system is not known and connections is tested, it is possible to uncheck PSEL, SSEL and TSEL selector matching and retest connection.

'Match IP Address' is not set by default, this means that the connection may come from any IP-address.



For cyber security, redefining the AR Name Matching Setting may be necessary

Note!

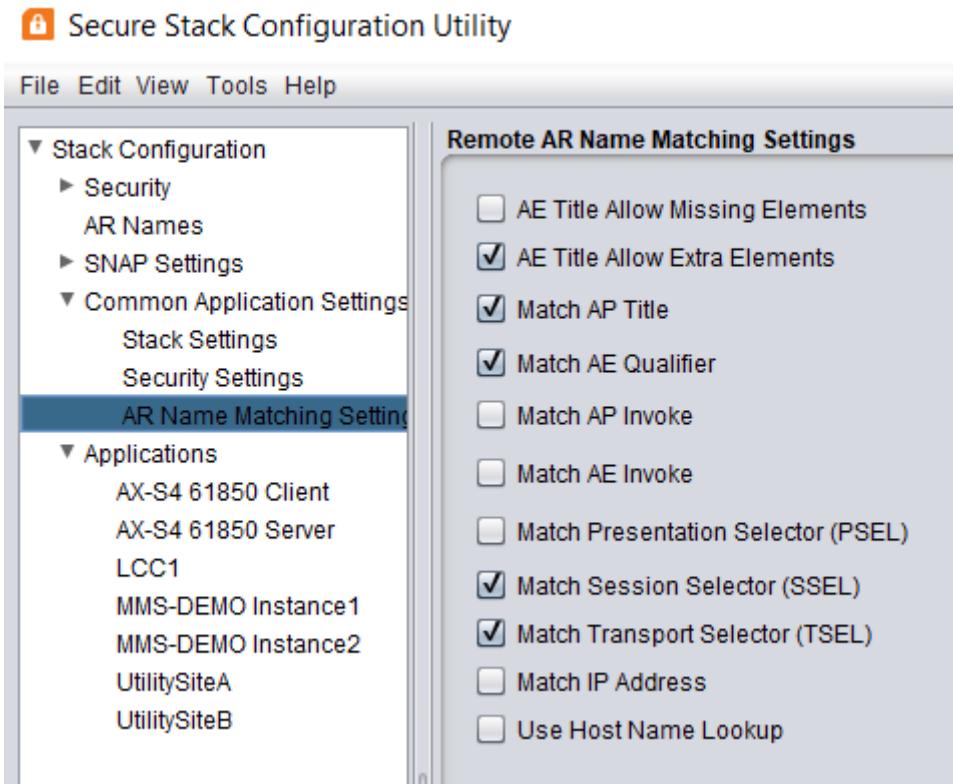


Figure 4.10: AR Name Matching setting

MMS Parameters of all remote systems are verified according to settings here.

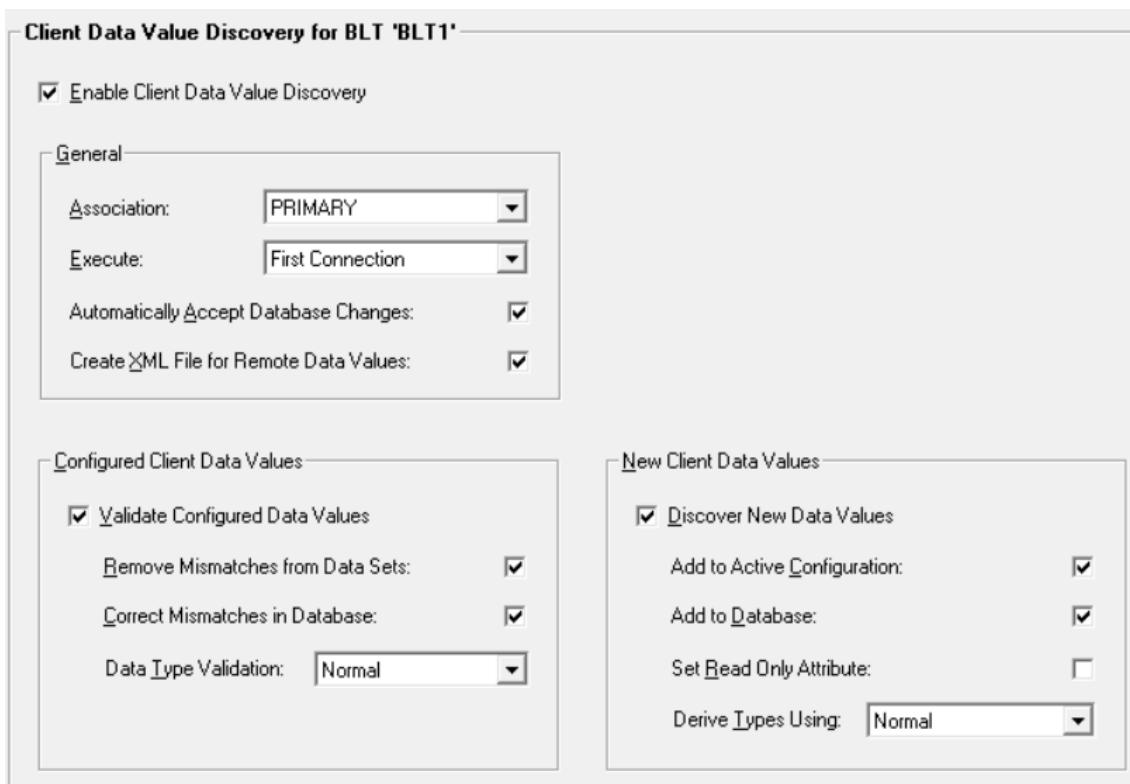
Note! ! The changes made to AR Name Matching settings are **not** stored to the XML-file loaded by the executable. This means that these configuration changes can be made as a last step in testing phase. Restarting of the communication is necessary for the changes to be effective.

Data Value Discovery

Data Value Discovery is a function which can be used to read available items in remote server and find mismatches in ICCP Names and types. It is available only for the client. The properties of the Data Discovery are defined separately for each RCC in ICCP Configuration Utility, definition is activated separately for each RCC, under 'Client':



With the settings below, in communication start-up, ICCP_Link.exe will create a file \sc\prog\iccp\rem_xxx.xml (xxx is the name of the RCC) for each RCC where Data Value Discovery is enabled. This XML contains a description of all available data items in RCC and reports possible mismatches of ICCP names and types. This is an efficient method to resolve configuration errors.



Data Value Discovery can also be triggered manually using line attribute DD Data Discovery request, see detailed information from attribute description.



The usage of Data Value Discovery is not affecting to the current configuration nor to XML-file loaded by ICCP_Link.exe.

Note!

If Client Data Value Discovery parameters are changed in some of the RCCs, the XML-file used by ICCP_Link.exe need to be exported again from ICCP Configuration Utility, see chapter Configuration Completion ICCP Configuration utility.



Remember the to disable Data Value Discovery and export the XML again when all mismatches have been resolved!

Note!

4.2.2.3 Starting and stopping the communication

When 'Write configuration' is made in ICCP Configurator, command procedures START_ICCP.SCL and STOP_ICCP.SCL are created to the same directory where the XML file for the executable is generated (Project Settings / runtime_directory).

Starting an instance requires modifications to procedure APL_INIT_1:C of the main application. In case the system is HSB, APL_INIT_H:C of the main application need to be modified, too. 'Project settings/runtime_directory' and project name defines the location in detail.

```
; Starting of NOD15 instance - directory is the default \sc\prog\ICCP\ and project name
; for this instance is 'TEST1'
@res = DO(READ_TEXT( "\sc\prog\ICCP\TEST1\START_ICCP.scl" ))

#set NET15:SIU1=1 ;Configuration is client only and started from main
;#set NET15:SIU1=0 ;Configuration contains server
```

If server configuration is used, communication is enabled by the COM500i application.

Stopping an instance requires modifications to procedure APL_CLOSE:C of the main application.

Similarly to starting, directory given in 'Project settings / runtime_directory' and project name defines the location in detail.

```
; Starting of NOD15 instance - directory is the default \sc\sys\ICCP\ and project name for
; this instance is 'TEST1'
@res = DO(READ_TEXT( "\sc\prog\ICCP\TEST1\STOP_ICCP.scl" ))
```

If system is HSB, copying the files from \sc\prog\ICCP\project needs to be made manually from System A to System B after 'Write Configuration' has been made. No modification is needed. Following changes to the project must still be done in the same system where it has been originally made.



In case of HSB, it is recommended to use the same system for configuration changes.

Note!

It is also possible to define the runtime_directory to point to shadowed area, e.g. '\sc\apl\main\apl_PICT' but a creation of a separate procedure which shadows the files using SCIL statement #SHADOW_FILE is needed. This procedure need to be executed after the configuration has been written.

If ICCP_Link.exe starting is needed for test purposes only, it can be done using 'Start ICCP_LINK'



button in ICCP Configurator. If 'Enable Communication' checkbox is checked, communication is enabled immediately (same as #SET NETxx:SIU1=1).



Note!

If configuration contains server configuration, COM500*i* application will activate the communication after a initialization period though 'Enable Communication' check-box is not set.



Note!

If setting remote control center/association/connection/role_description of some RCC is of type 'Calling', an enabled communication will activate connection attempts to configured addresses immediately. Since the final addresses most probably are from another organization, it may make sense to define temporary addresses in ICCP Configuration utility while system is internally tested.

4.2.2.4 Troubleshooting

Following table gives some hints if problems are faced during configuration or communication testing. If word is written in *italic*, more information of the topic can be found via index of this manual.

For all symptoms related to the functionality of ICCP_Link.exe, following actions can considered

- Consider patching the last parameter of 'ops_process' call in START_ICCP.SCL from "SERVER" to "CLIENT". This will open a console for the ICCP_Link.exe.
- Check file \sc\prog\iccp\iccpTrace.log.
- Consider enabling the logging by modifying the START_ICCP.SCL file with setting of logging mask:
; Add NOD attributes

#SET net'ICCP_NODE':slm(101)=67

See line attribute LM for more information.

Table 4.4: Troubleshooting

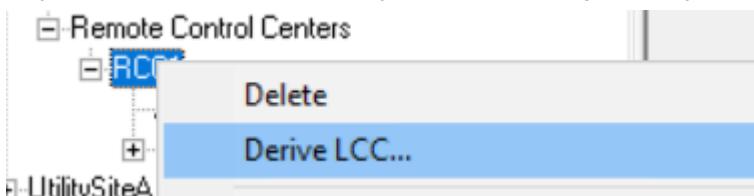
Symptom	Possible reason
Configuration writing in ICCP Configurator fails	<ul style="list-style-type: none"> • <i>server_stn</i> number is not given to Local Control Center and/or <i>client_stn</i> number is not given to Remote Control Center • Nodes and station objects are not created, or the system is not restarted after the information has been entered to the system configuration tool • COM500iNCC is not created or it is not of type ICCP or the given station object is not matching with <i>server_stn</i> number given to Local Control Center • 'Project Settings/<i>runtime_directory</i>' refers to location which cannot be accessed, is write-protected or drive is not existing
XML file importing to ICCP Configuration Utility fails	<ul style="list-style-type: none"> • Defined ICCP Names are not following the <i>naming rules</i>. • Addressing in ICCP Configuration Utility is not containing <i>local_ar</i> and <i>remote_ar(s)</i> • No items have been defined to DataValues in either Client or Server • Check \ProgramData\SISCO\ICCP Configuration\IccpXML.log for other possible reasons
ICCP_Link.exe starting is reported to be failed and ICCP_Link.exe is not staying in the Windows task list	<ul style="list-style-type: none"> • XML file is not in the location specified in START_ICCP.SCL (Variable Config_dir) • XML file name is not matching with the third parameter given to ops_process(.. in START_ICCP.SCL (ops_process("EXE_File' 'CONFIG_Dir' LCC1 ..)) • XML file has not been successfully imported to ICCP Configuration Utility with the specified name, check the tree structure in ICCP Configuration Utility • Virus scanner has quarantined ICCP_Link.exe
ICCP_Link.exe starting is reported to be failed and ICCP_Link.exe is staying in the Windows task list. Attribute access to defined node is not possible from test dialog	<ul style="list-style-type: none"> • NODxx:BLI refers to some other link but of type "LAN" (xx is node_number) • Basesystem <i>hardening setting</i> "REQUIRE_ACP_ENCRYPTION" is other than "NONE" or "NETWORK" • LAN Link to basesystem is not created • Node number and its address is bigger than 55 / 255 • There is no license for ICCP, all attribute access return SCIL error 30014, NODE_LICENSE_FAILURE
Listening socket for port 102 is not visible though ICCP_Link.exe is running and system is configured as role_description = called and communication is enabled.	<ul style="list-style-type: none"> • firewall is preventing the listening. Open port 102 • IP-address defined for <i>local_ar</i> in ICCP Configuration Utility is not matching with any of the local IP-addresses
ICCP_Link.exe is not making TCP connection attempts though it is running and system is configured as role_description = calling and communication is enabled.	<ul style="list-style-type: none"> • IP-addresses defined to <i>remote_ar(s)</i> cannot be resolved. Try pinging. • IP-address defined for <i>local_ar</i> in ICCP Configuration Utility is not matching with any of the local IP-addresses

Symptom	Possible reason
Communication in TCP-level is opened once shortly but quickly closed. This sequence happens only once after restart. Wireshark indicates "Release Response (not finished) as last message before TCP session close.	<ul style="list-style-type: none"> TCP Connection attempt is not coming from a configured IP-address and 'Match IP Address' is checked <i>Secure Stack Configuration Utility</i> <i>local_domain</i> or <i>remote_domain</i> is not matching with the corresponding values in RCC BLT <i>id</i> is not matching with the BLT id defined in RCC
Communication in TCP-level is opened shortly but quickly closed. This happens cyclically . Wireshark indicates "MMS Initiate error".	<ul style="list-style-type: none"> MMS parameters, e.g. PSEL, SSEL or TSEL of RCC is not matching with the local configuration. Check parameters of in RCC. Matching conditions defined in Secure Stack Configuration Utility may be worth trying. Line is not in use (server), i.e. NETxx:SIU1 is 0.
Communication in TCP-level is stable but no data is transferred	<ul style="list-style-type: none"> Verify that line and station objects are in use (evaluate NETyy:SIU1 and STAxx:SIU from test dialog, should be 1) Data set creation has failed, check the amount and state of data sets using attribute DS (Client STA) or SD (Server STA). Item count in data set should be bigger than 1 and status should be OK/Enabled. See attribute description for details. Index 0 has special meaning in attribute reading. Note that data transfer may be occurring in one direction but not in opposite direction. Verify signal lists or consider using Data Discovery (client). If problem is with the data transmitted from SYS600 operating as ICCP server, the usage of Data Discovery may be asked from the contact persons of the remote system. Verify that '#LOCAL MAINAPL' setting in the used START_ICCP.SCL procedure refers to expected application Enable UNDEF_PROC event channel and forward snapshot variables to console_output, see more information from SYS600 System Objects manual
Communication in TCP-level is stable but only part of the configured data is transferred	<ul style="list-style-type: none"> Some of the configured datasets has mismatching ICCP Names or types. Verify signal lists or consider using Data Discovery (client). If problem is with the data transmitted from SYS600 operating as ICCP server, the usage of Data Discovery may be asked from the contact persons of the remote system. Verify the consistency of ICCP type and the process object type in process database. E.g. ICCP Type 'Discrete' requires analog input point. Verify that source of the data is cross-referenced correctly in COM500i (Server). Verify also from the <i>ini-file</i> xxx_S.ini file that the identifier in question is matching with expected ICCP item. Verify that IN and UN attributes of the process object are matching with configuration and the process objects are in AUTO state (client)

Below some hints which may be useful in configuration and testing phases:

- When process object are created, it is worth to consider setting the agreed ICCP Name to the IN-attribute of the process object. Re-entering of the ICCP names to ICCP Configurator is thus avoided.
- Setting the most common required ICCP Datatypes to default data types speeds up item definitions and results less errors.
- Entering temporary IP Addresses to ICCP Configuration Utility allows free start & stop operations of ICCP_Link.exe.

- If system to be built is 'Server only', a corresponding Client system XML can be created using selection



This XML can be imported to another SYS600 system which operates as a test client. An empty client configuration can be made using ICCP Configurator.

or the Starting procedures need to be created manually. Data set creation must be made similarly to client configuration described earlier in this manual.

Basically, the same derived LCC can be created in 'Client only' setups, too, but in this case the xxx_S.ini file need to be created manually to have mapping to ICCP Item Names from identifiers entered to COM500 or for direct access using attribute DV.

- If system contains client, Data Discovery can be used to obtain Data items configured to RCC. See chapter 'Data Discovery' for more information.

4.2.2.5 Attribute descriptions

General

These chapters describe the attributes of the node, line and station objects. If not explicitly mentioned otherwise, attribute is supported both in client and server.

Node attributes

This chapter describes the attributes of the ICCP node. These attributes are accessed using notation NETn:Sat or NODn:Sat where

'n'	The object number (=node number) of the unit, 1...250
'at'	An attribute name

Node attributes define the common properties of the all functions within one node.

Base system attributes for a node are accessed using notation NODn:Bat, see SYS600 System Objects manual for more information. These attributes are same for all types of communication units.

SA Address of the node

ACP messaging address of the node, value is 200 + net node number given as argument using definition 'c'.

Data type:	Integer
Value:	200 + node number
Index range:	No indexing
Access:	Read Only

DV Device create

By writing a value to this attribute, ICCP node creates a device which is accessed using STA:S notation. Given arguments define the number of the STA object and the mode of the operation (server or client). If the ICCP node instance should operate both as server and client, two write operation is needed. Same created STA object may serve multiple Remote Control Center. See chapter Starting and Stopping the communication for an example.

Data type:	Integer
Value:	Element 1: STA object number, 1...512 Element 2: Mode of operation 1=Server, 2=Client Element 3: Line number, always 1.
Index range:	No

DV Device create

Access: Write only

Example:

```
#SET NET15:SDV=(101, 2, 1) ;create STA101 operating as client
```

EX Node exit

By writing a value to this attribute, ICCP node closes all connections and exits itself. Usage of this attribute is recommended instead of killing the process. The written value is meaningless.

Data type: Integer

Value: 1

Index range: No

Access: Write only

Line attributes

Line attributes are accessed like node attributes, but index must be given. At the moment, the given index must always be 1.

IU In Use

IU Attribute controls if the RCC connection in use or not. All communication is stopped, and STA objects are in suspended state if line IU is 0. Setting IU to 1 enables all associations.

Data type: Integer

Value:
0 = Line is out of use
1 = Line is in use

Index range: Always 1

Access: Read/Write

DD Data Discovery request

ICCP Protocol has special 'data discovery' functionality which means that the client part of the link negotiates with the server part of the remote control center that both sides have same object list. For example, client DataValue list matches to Data Value list of the RCC server. Differences with lists are stored to xml-file if it is enabled in ICCP Configuration Utility configuration. This xml-file is also possible to read (merge function) to original xml configuration file for later use. Since Data Discovery function is executed at the initialization phase of communication and for performance reasons, recommended configuration is executed only once, the user can request discovery also for later starts by DD attribute. Discovery is executed when line IU is set from 0 to 1. When read, value indicates the current status of the Data Discovery. When written, Data Discovery is initiated.

Data type: Integer

Value:
0 = Data Discovery has not been done
1 = Data Discovery has been requested
2 = Data Discovery has been done
3 = Data Discovery has been done by user's request

Index range: Always 1

Access: Read/Write

GS General Data Sending

Writing to this attribute triggers the sending of all local data (whole local database using all associations) to the remote clients. The value written should always be 1. Attribute is meaningful only for server and is most useful for test purposes.

Data type: Integer

GS General Data Sending

Value: 1 = Sending of local data

Index range: Always 1

Access: Write

Example :

#SET NET15:SGS1= 1 ; Send all data

OM Operating Mode

Operating mode is a bitmask which defines certain modes for operation. Please see the description of each bit below.

Data type: Integer

Value: 0...1 (Bits 1...15 not used)

Default: 0

Index range: Always 1

Access: Read/Write

Bit 0 When bit 0 is set, client bypasses the CheckBackId validation of the remote device during Device Select operation. When bit 0 is not set, received CheckBackId of the selected device is validated against the configured CheckBackId of the device. This is the default functionality. If validation fails, Device operate is not made. Bit 0 is meaningless if link operates only in server mode.

Example :

#SET NET15:SOM1= 1 ; Bypass CheckBackID validation

AC Association Status

Attribute returns the status of specific association. If requested association is not configured, returned association name is "-"

Data type: String

Value: "STATUS, ASSOCIATION_NAME, AR_NAME" where

STATUS is integer

0 : Idle

1 : Active

2 : Connecting

3 : Abort

4 : Backoff

5 : Invalid

ASSOCIATION_NAME

name of association

AR_NAME

name of remote AR

Index range: 100+association number

Access: Read

Example :

```
@AC = NET15:SAC101 ; Status of association 1
; %AC= "1,PRIMARY, SiteB1" meaning 'Active', name = "PRIMARY", current remote_ar "SiteB1"
@AC = NET15:SAC102 ; Status of association 2
```

```
; %AC= "0,-,-" meaning 'Idle', not configured, no remote AR
```

MD Info message directory

Attribute MD defines the directory where incoming info messages using file transfer method is stored. By default, used directory is c:\temp\|. The directory defined with MD can be freely selected, but the application must have write access to the specified directory. Last character of defined string should be "\|. Function is supported only with STA objects operating as clients. See LINE attribute IC for the options of the processing of the incoming messages.

Data type:	String
Value:	Directory name
Default:	"c:\temp\ "
Index range:	Always 1
Access:	Read/Write

Example :

```
#SET NET15:SMD1="C:\sc\apl\main\PICT\|"; Received files are in a shadowed area.
```

IC Info Message Configuration

Attribute IC defines how the received info messages are processed in client. The referred (IEC) bitstream object should have OA equal to 'Info Reference' field defined for the Message (in ICCP Configuration Utility) and UN of the process object equals to STA object in question. If message is stored to a file, its extension is ".txt" and the name of the file is a combination of 'Info Reference', 'Local Reference' and 'MessageId' values. The directory is specified with LINE attribute MD Message Info Directory. Transmission of info messages takes place only in direction server -> client. Configuration of Info Messages is not supported in ICCP Configurator.

Data type:	Integer
Value:	-1 : All info messages are stored to a file and the header info is stored to a bitstream object. 0 : Info message is stored to the bitstream object as a whole if its length is not more than 200 characters. If length is more than 200 characters, message is stored to a file similarly to configuration value -1. Values >0 : if given value equals to 'Info Reference' value of the received message, message is stored to a file similarly to configuration value -1. Other messages are stored to bitstream object similarly to configuration value 0.
Default:	0
Index range:	Always 1
Access:	Read/Write

LM Logging mask

This attribute is used to verify and control the state of ICCP logging. Logging can be configured separately for each layer or for each group of functions. Received events are written to SISCO logging subsystem and on-line view can be seen from SISCO Log viewer (menu selection 'Tools->Start Log Viewer'). Using attribute LM, it is possible to modify the set of logged events and data while system is running. Mask definitions are from SISCO documentation. In startup of Iccp_Link.exe, the logging configuration is read from logcfg.xml which is located in the same directory with the configuration XML for the ICCP_Link.exe executable. Manual editing of the logcfg.xml is possible, entry

```
<LogFileName>\sc\prog\iccp\IccpTrace.log</LogFileName>
```

defines the name of the log. The location and name of the file is freely selectable.

Data type:	Integer
Value:	Logging mask for the group - bitmask and enabled option for each group is described below
Default:	0

LM**Logging mask**

Index range:

- 100+logging group
- 01 = ICCP Lite Plus library logging
- 02 = MMS library logging
- 03 = MVL library logging
- 04 = SISCO security stack library logging
- 05 = MACE handling functions logging
- 06 = ACSE, COPP (OSI Presentation) and COSP (OSI Session) layer logging
- 07 = Transport layer messages logging
- 08 = Network layer logging
- 09 = ASN.1 logging
- 11 = ICCP server logging
- 12 = ICCP client logging
- 13 = Line SDL transitions logging
- 14 = Device SDL transitions logging (all devices online)

Access:

Read/Write

Group 01

- MILOG_ERR 0x00000001
- MILOG_NERR 0x00000002
- MILOG_FLOW 0x00000004
- MILOG_DEBUG 0x00000008
- MILOG_ADDR 0x00000010
- MILOG_CFG 0x00000020
- MILOG_DISCOVERY 0x00000040

Default: MILOG_ERR, MILOG_DISCOVERY, MILOG_ERR and MILOG_NERR enabled

Group 02

- MMS_LOG_DEC 0x00000001L MMS decoding process
 - MMS_LOG_ENC 0x00000002L MMS encoding process
 - MMS_LOG_ERR 0x00010000L Abnormal errors
 - MMS_LOG_NERR 0x00020000L Normal errors
 - MMS_LOG_RT 0x00100000L All Runtime Type transactions
 - MMS_LOG_RTA 0x00200000L All Runtime Type AlternateAccess transactions
 - MMS_LOG_AA 0x00400000L All Alternate Access transactions
- Default : MMS_LOG_ERR enabled

Group 03

- MVLLOG_ERR 0x00000001 MVL Critical Errors
 - MVLLOG_NERR 0x00000002 MVL Normal Errors
 - MVLLOG_ACSE 0x00000040 MVL ACSE Encoding/Decoding
 - MVLLOG_ACSEDATA 0x00000080 MVL ACSE Encoding/Decoding HEX data
 - MVLULOG_FLOW 0x00000200 MVL program Flow
- Default: MVLLOG_ERR and MVLLOG_NERR enabled

Group 04

not supported at the moment - reserved for secure stack logging

Group 05

not supported at the moment - reserved for secure stack logging

Group 06

- ACSE_LOG_ERR 0x00000001 ACSE Errors
 - ACSE_LOG_ENC 0x00000002 ACSE Encoding
 - ACSE_LOG_DEC 0x00000004 ACSE Decoding
 - ACSE_LOG_DIB 0x00000008 ACSE DIB (i.e., network addressing)
 - COPP_LOG_ERR 0x00001000L COPP Errors
 - COPP_LOG_DEC 0x00001000L COPP Decoding
 - COPP_LOG_DEC_HEX 0x00002000L COPP Decoding Hex
 - COPP_LOG_ENC 0x00004000L COPP Encoding
 - COPP_LOG_ENC_HEX 0x00008000L COPP Encoding Hex
 - COSP_LOG_ERR 0x00010000L COSP Errors
 - COSP_LOG_DEC 0x00100000L COSP Decoding
 - COSP_LOG_DEC_HEX 0x00200000L COSP Decoding Hex
 - COSP_LOG_ENC 0x00400000L COSP Encoding
 - COSP_LOG_ENC_HEX 0x00800000L COSP Encoding Hex
- Default: ACSE_LOG_ERR and COPP_LOG_ERR and COSP_LOG_ERR enabled with decoding

LM	Logging mask
Group 07	TP4_LOG_ERR 0x00000001 Transport Errors TP4_LOG_FLOWUP 0x00000002 Transport Decode (incoming TPDUs) TP4_LOG_FLOWDOWN 0x00000004 Transport Encode (outgoing TPDUs) Default: TP4_LOG_ERR enabled
Group 08	SOCK_LOG_ERR 0x0001 Socket errors SOCK_LOG_NERR 0x0002 Recoverable errors SOCK_LOG_FLOW 0x0004 Program flow SOCK_LOG_RX 0x0008 Socket transmit SOCK_LOG_TX 0x0010 Socket receive Default: SOCK_LOG_ERR and SOCK_LOG_NERR enabled
Group 09	ASN1_LOG_DEC 0x0001 ASN.1 decode process ASN1_LOG_ENC 0x0002 ASN.1 encode process ASN1_LOG_ERR 0x0004 Abnormal ASN.1 errors ASN1_LOG_NERR 0x0008 Normal ASN.1 errors Default: ASN1_LOG_ERR and ASN1_LOG_NERR enabled
Group 11	LOG_ERR 0x0001 Log server errors LOG_NERR 0x0002 Log server recoverable errors LOG_FLOW 0x0004 Log server flow LOG_DATA 0x0008 Log server info message data Default : none
Group 12	LOG_ERR 0x0001 Log client errors LOG_NERR 0x0002 Log client recoverable errors LOG_FLOW 0x0004 Log client flow LOG_DATA 0x0008 Log client info message data Default : none
Group 13	LOG SDL 0x0001 Log SDL Transitions of Line layer process Default : none
Group 14	LOG SDL 0x0001 Log SDL Transitions of STA layer process Default : none

Example :

```
#SET NET15:SLM112=1+2+4 ; Log client errors, recoverable errors and execution flow
#SET NET15:SLM114=1 ; Log transitions of STA processes
```

LS **Log current State**

Attribute LS is used to trigger the writing the current configuration or state to the logging device. Possible value are listed below. This attribute is useful e.g. when there is a need to verify that intended configuration is actually loaded.

Data type:	Integer
Value:	1..4 1 = Log configuration 2 = Log ICCP state 3 = Log association state 4 = Log local xref state by SYS600 address view
Index:	Always 1
Access:	Write

Example :

```
#SET NET15:SLS1=4 ; Log current datapoints and their SYS600 addresses to logging device
```

Station attributes**IU In Use**

IU Attribute controls if the RCC connection in use or not. Communication is disabled and STA object is in suspended state if station IU is 0.

Data type:	Integer
Value:	0 = STA object is out of use 1 = STA object is in use
Index:	No index
Access:	Read/Write

DI Database Initialization

When ICCP is operating as server, local data need to be initialized before communication to remote may begin. Setting DI to 1 indicates that current data value has been updated and communication can be started. Attribute is meaningful only in STA objects operating as server.

Data type:	Integer
Value:	0 = Database for local data is not initialized 1 = Database for local data is initialized
Index:	No index
Access:	Read/Write

DS Dsts control and status

Data Set Transfer Set (Dsts) controls a data structures for data sets. Configuration of data set is managed by ICCP Configuration utility. Using the DS attribute is possible enable/disable the functionality of dsts and check also the status of specified dsts.

This attribute is valid only for manually configured dsts structures, not structures which are created using auto dsts function in runtime. Attribute is supported only in STA objects operating as client. See attribute SD for corresponding server function.

Data type:	Integer
Value:	When written: 0 = Disable communication of Dsts 1 = Enable communication of Dsts 10 = GI for Dsts (Refresh request) When Read: if index = 0 vector = (DSCOUNT, DSCOUNT_OK) if index > 0 vector = (STATUS, ITEMCOUNT, STATE)
Index:	Data Set num or 0 when read
Access:	Read/Write
DSCOUNT	Total number of Data sets, numbering 1..DSCOUNT
DSCOUNT_OK	Number of Data sets communicating in a association
STATUS	0 = Failed 1 = OK
ITEMCOUNT	Total amount of data items (including 3 predefined special variables)

DS Dsts control and status

STATE 0 = Communication of Dsts is disabled
1 = Communication of Dsts is enabled

SD Dsts status

Data Set Transfer Set (Dsts) controls a data structures for data sets. Data sets are created by client when communication is established. Using attribute SD it is possible to check the status of specified dsts.

Attribute is supported only in STA objects operating as server. See attribute DS for corresponding client function.

Data type:	Integer
Value:	if index = 0 scalar integer = DSNUM if index > 0 vector = (STATUS, ITEMCOUNT, FLAGS)
Index:	Data Set num or 0 when read
Access:	Read
DSNUM	Total number of Data sets, numbering 1..DSNUM
STATUS	0 = Data set is disabled (transfer stopped) 1 = Data set is enabled (transfer started)
ITEMCOUNT	Total amount of data items (including 3 predefined special variables), 0 if inactive
FLAGS	Internal status bit mask Bit 0 : In use Bit 1 : First change Bit 2 : Report in progress

DN Data set Name

Attribute DN returns the name of the specified Data Set. Total number of data sets can be read using attributes DS and SD, index 0 and numbering starts from 1. In server, returned data set name is an empty string if specified data set is disabled or no data has been configured to it. In client, returned data set name is an empty string only if no data has been configured to it. See attributes DS and SD for more information.

Data type:	String
Value:	name of the Data Set
Index:	Data Set number
Access:	Read

RN Remote control center Name

The content of this attribute binds STA object and RCC object. RCC name is defined in configuration and same value should be given to RN attribute of the corresponding STA object. Attribute is meaningful only in client. When empty string is given, STA object uses numerical process object addressing method and reads the cross reference info between iccp object names, process object addresses (OA) and process object unit numbers (UN) from xxx_C.ini files, the format of this file is similar to xxx_S.ini file. With ICCP Configurator it is not possible to use numerical addressing since it does not create xxx_C.ini file. If xxx_C.ini is used, OAs of the process objects must be defined manually. In addition to matching and not empty RN, the process object addresses should have matching UN = STA number and IN = ICCP Name of the process object. Currently only client has string addressing method.

Data type:	String
Value:	Remote control center (RCC) name
Index:	No
Access:	Read/write

5 Technical description

5.1 General

IEC60870-6 protocol has originally been developed under the guidelines of the Utility Communication Architecture targeting to create a standardized protocol to meet the needs for sharing of real-time data and other functions between control centers, powerplants and substations.

Protocol widely called as ICCP or TASE.2 is later standardized by International Electrotechnical Commission (IEC) as IEC60870-6 group of standards which specify the functionality.

ICCP uses OSI-7 model and from TASE.2 version, MMS (Manufacturing Message Specification), similarly to IEC61850 protocol used within substations.

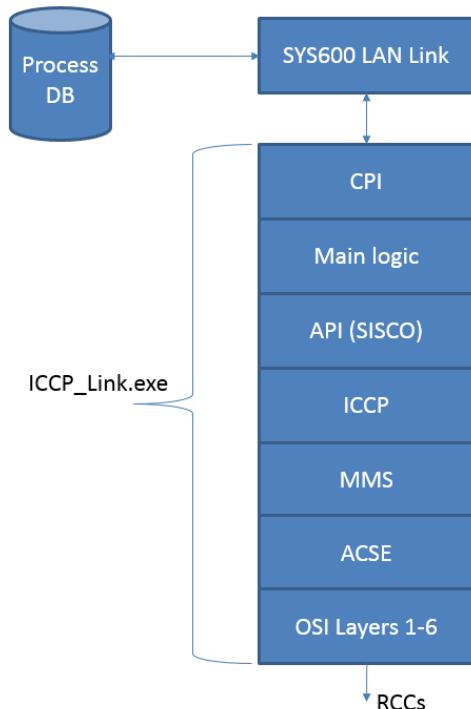


Figure 5.1: Layers in ICCP_Link.exe

When ICCP is configured as server, TCP port 102 is opened in defined local IP-address and a firewall rule must be defined to open the port. Encryption and certificate validation using TLS (Transport Layer Security) is not supported. Chapter 'Secure Stack Configuration Utility' defines setting which can be used to improve security



TCP port 102 is used also by the IEC61850 Server functionality of SYS600 and same IP-address cannot be used with ICCP

Note!

Blocks used in ICCP (bolded ones supported in SYS600):

Block 1	Periodic System Data
Block 2	Extended Data Set Condition Monitoring
Block 3	Blocked Transfers

Block 4	Information Messages
Block 5	Device Control
Block 6	Execution of programs in remote ICCP server
Block 7	Extended reporting of error conditions and device state changes at ICCP server
Block 8	User objects for plant information, scheduling, accounting and outage
Block 9	Transmission of historical data based on requests from client

5.2 Communication

5.2.1 Data flow

Data flow from/to process database to/from RCCs occurs through client and server communication attributes and direct process object writes:

- Write to output process object for a client activates an internal message which generates a control command
- Write to client STA communication attribute generates a control command or a tagging command
- Received data update in server generates an internal message which updates an input process object in process database
- Received control command in server generates an internal message which updates COM500i BNCC process object in process database
- Write to server STA communication attribute generates a data update from server

Data updates from other protocols and received commands from other NCC types but ICCP are processed in COM500i signal routing procedures and may activate data updates in ICCP server or control command transmission ICCP client. Following picture describes data flow between different components and layers:

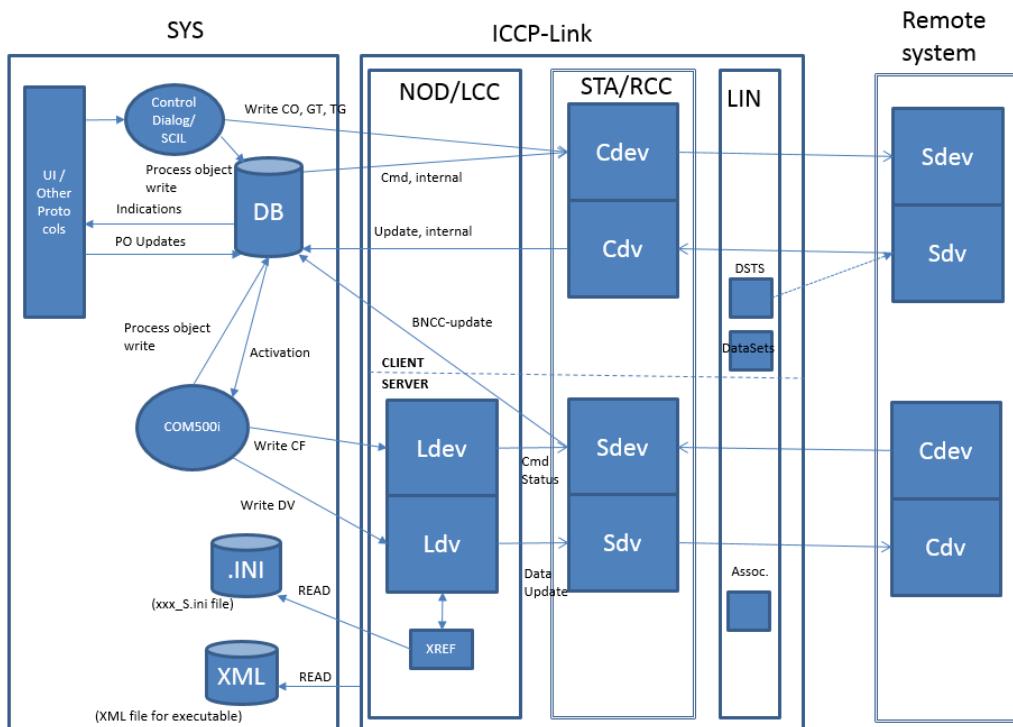


Figure 5.2: Data flow

5.2.2 Communication attributes, client STAs

Following attributes can be used to control the runtime functionality of the STA objects operating as client. In addition to these, control commands and analog setpoints can be sent using process objects which are in automatic state (SS=2).

CO Command Out

Client can also send control commands or setpoint values to device in a remote center device by writing directly to attribute CO, instead updating an output type process object. Attribute CO should be used only in special cases, e.g testing or maybe in sequences. First parameter of the given vector is a string containing ICCP name of the device. Second parameter is the command and third parameter defined whether the command is select or operate.

Data type: Integer

Value: Vector of (DEVNAME, VALUE, OPERATION) where
 DEVNAME (String of max 32 bytes equal to iccpname)
 VALUE (depends on Datatype defined for the device)
 Control command : 0 or 1
 Discrete : Integer
 Real : Floating point number
 OPERATION
 0 = Operate
 1 = Select

Index: No

Access: Write

```
#SET STA101:SCO = ("TPC_XDL_10L5_CB_13",1,1) ; Send a 'select' control command to device
TPC_XDL_10L5_CB_13
```

TG Tag of Remote device

Attribute TG is used to perform the operation 'Set Tag' on a device in a remote control center. Tagging in ICCP means defining an control inhibition and a reason for it to a controllable device in the remote system. First parameter in the given vector defines the device to be tagged, second parameter defines the tag and third parameter defines the freely set reason string for the tagging. The accessed device should have tagging enabled, otherwise an error is retruned to SCIL if tag setting is attempted. See attribute GT for tag reading.

Data type: Integer

Value: When written, vector of (DEVNAME, TAG, REASON) where
 DEVNAME (String of max 32 bytes equal to iccpname)
 TAG (Integer):
 0 = NO-TAG
 1 = OPEN-AND-CLOSE-INHIBIT
 2 = CLOSE-ONLY-INHIBIT
 Real : Floating point number
 REASON (String of max 128 bytes)

Access: Write

```
#SET STA101:STG = ( "TPC_XDL_10L5_CB_13", 1, "In service") ; Set tag OPEN-AND-CLOSE-INHIBIT
with reason text for a device TPC_XDL_10L5_CB_13
```

GT Get Tag

Writing to attribute GT triggers the reading of the tagging of the control device in the remote control system. Attribute is supported only in client. When written, two strings are given. First one defines the ICCP name of the control device to be inspected, second one defines the string address (IN attribute) of the process object to be updated with the tag value. Process object type of the response object is IEC Bitstream and it should also have a matching UN (Unit number). The content of the update response process object is a string in format "F, text" where possible value for 'F' are

- '0' = Not tagged
- '1' = Open and Close inhibit
- '2' = Close inhibit
- '4' = Armed, not tagged
- '5' = Armed, open and close inhibit
- '6' = Armed, close inhibit
- 'N' = Tagging not enabled
- 'F' = State Unknown

"Armed" in ICCP means that device is reserved for operate. Value of 'text' is free text given to device when tagged. Tag setting of the remote device is done using attribute TG. Conversion from bitstream object to visible string is needed.

Data type: String

Value: Vector of strings (DEVNAME, RESPOBJ) where
 DEVNAME is the ICCP Name of the remote device in question (max 32 characters)
 RESPOBJ is the IN attribute value of the process object receiving the tag value (max 40 characters)

Index: No

Access: Write

```
#SET STA101:SGT=( "TPC_XDL_10L5_CB_13", "TPC_XDL_10L5_CB_13_TG") ; Get tag of
TPC_XDL_10L5_CB_13 to process object
```

5.2.3 Communication attributes, server STAs

Following attributes can be used to control the runtime functionality of the STA objects operating as server. Normally, these attributes are used only by the COM500i signal routing application.

DV Data Value

Data Values sent to RCCs are written using DV attribute of the station object operating as server. Numerical value given as an index in attribute write is mapped to ICCP Data item using the information loaded from xxx_S.ini in the startup of the ICCP_Link.exe. Used index is visible as an identifier in COM500i cross reference tool.

The meaning of each bit in state supplement byte is defined in detail in IEC60870-6 TASE.2 standard.

DV**Data Value**

Value:

Vector of (VALUE, QUALIFIER, SOURCE, STATUS, [RT], [RM]) where
 VALUE (dependent on type defined for item in ICCP Configurator)
 Type STATE : Byte (0 = Between, 1 = ON, 2 = OFF, 3 = Invalid)
 Type STATE SUPPLEMENT : Word
 If High Byte > 0, Low byte is used as such as state supplement byte [bit]:
 StateHi[0],StateLo[1],TagHi[2],TagLo[3],ExpectedStateHi[4],ExpectedStateLo[5],Res-
 served[6],Reserved[7]
 If High Byte = 0, Low byte is handled as STATE
 Type DISCRETE : Integer (32 Bit unsigned)
 Type REAL : Float
 QUALIFIER
 0 = Local Value update (update does not send value to RCCs)
 1 = Valid
 2 = Held
 3 = Suspected
 4 = Not Valid
 0 = Local Value update (update does not send value to RCCs)
 1 = Valid
 2 = Held
 3 = Suspected
 4 = Not Valid
 SOURCE
 0 = Telemetered
 1 = Calculated
 2 = Entered
 3 = Estimated
 STATUS
 0 = Normal
 1 = Invalid Time
 2 = Abnormal
 RT (optional)
 Time, Registration time without milliseconds
 RM (optional)
 Registration time milliseconds

Index:

Data item identifier (1..65535)

Access:

Write

6 Abbreviations and terms

Following table describes the terminology and abbreviations used in ICCP protocol and tools.

ACSE	Association Control Service Element. System is identified with a set of ACSE parameters (AP Title, AE Qualifier, AP Invoke and AE Invoke) in the application layer of OSI Addressing
ASN.1	Abstract Syntax Notation One. Transfer syntax used in MMS which is defined in ISO/IEC 8825 Standard
Armed	Reserved for execute operation, equal to 'Selected' in other protocols
AR Name	Application reference name. A readable identifier for a system
Association	Connection between two Control Centers
Auto DSTS	Automatic data set creation. This function creates the data set and places the local data values to data set without manual work
Bilateral table	Table containing the exchanged data items
Data Value	Data item provided from local system to at least one remote system
Data Set	A collection of Data Values transmitted together
Data Set Transfer Set	A definition how Data Set is transferred
Data Value Discovery	A mechanism to retrieve the details of the configuration of the remote system
Device	A controllable device, e.g. a switch
Dsts	Data Set Transfer Set
Local Control Center (LCC)	A collection of data and devices provided to the remote system
MMS	Manufacturing Message Specification
MVL	MMS Virtual Light interface
RBE	Report-by-exception, spontaneous transmission.
Remote Control Center (RCC)	A collection of data and devices available from the remote system
SDL	State Diagram Language
SBO	Select-before-operate
Tagging	Defining an control inhibition and a reason for it to a controllable device in remote system. Not related to tagging for e.g. SYS600 Historian.

7 Updating pilot installations

Installations made on top of SYS600 version 10.1.1 or older are configured using Excel-based tool. Existing and working ICCP connections can be used as such after the SYS600 update, too, but configuration changes using ICCP Configurator cannot be made until existing configuration is transferred to a ICCP Configurator project. It is recommended to make the changes in a separate computer. The principal sequence of updating of these systems to SYS600 version 10.2 or newer is as follows:

1. Take a backup of the system to be updated, including the ICCP Excel Tool which contains the latest configuration of process object cross-referenced for ICCP (both server and client)
2. Export the cross-reference information of the process objects to client and server specific CSV-files which structured as specified in chapter Import from Excel
3. Verify the contents of the CSV files to the existing configuration in COM500*i* cross-reference tool and files xxx_S.ini and xxx_C.ini
4. Update SYS600 to a new version and install/reinstall ICCP Configuration Utility
5. Add ICCP specific nodes and STA object to the system configuration using System Configuration Tool. Remove corresponding settings from SYS_BASCON.COM or from procedures. Restart SYS600.
6. Start ICCP Configurator and create a new project for the system to be updated.
7. Define project setting according to requirements, define local_ar/remote_ar fields, domain name fields, BLT id values according to existing system. Values seen in ICCP Configurator installed to the existing system are helpful
8. Save project and take a backup of it
9. Copy the created CSV-files to locations defined in 'Project Settings' and import the contents as described in chapter 'Import from Excel'. Depending on configuration, it may be feasible to make the import in smaller blocks, using filtering and redefinition of default datatypes.
10. When all process objects have been imported and defined to be similar with the existing system, continue configuration as instructed in chapters 'Configuration completion ICCP Configurator' and 'ICCP Configuration Utility'. Values entered to ICCP Configuration Utility are worth to copy from the existing system



Note! When 'Write Configuration' is made for the first time, COM500*i* cross-reference information will be updated with new identifiers and IN-attribute of the process objects for client will be updated.

A special Excel tool needed in step 2 is delivered separately when requested. Contact nearest Hitachi ABB Powergrids representative for delivery.

Deleting the existing xxx_C.ini file (mapping information for client) is worth to do when configuration tasks have been completed. A start procedure created by ICCP Configurator will define a RCC specific value to RN attribute of the client STA objects, this instructs ICCP_Link.exe to use string based addressing and xxx_C.ini is not used anymore. In versions older than 10.2, the content of the RN attribute has been empty.

Index

A

abbreviations, 57
 AC, 45
 ACSE, 47, 57
 Add New RemoteControlCenter, 20
 Addressing, 31
 APL_INIT_1:C, 39
 APL_INIT_H:C, 39
 Armed, 21, 54, 57
 AR Name, 57
 ASN.1, 47, 57
 assoc_heartbeat_time, 27
 assoc_retry_time, 27
 Association, 57
 Associations, 18
 Association Status, 45
 Auto DSTS, 57
 Auto DSTS Activation, 35

B

Bilateral Table, 57
 Bilateral tables, 19
 Bitstream object, 46
 Blocks, 27
 BLT id, 27
 BLT Id, 42

C

CD Attribute, 22
 Certificate, 51
 Check Back Id, 26, 29, 45
 Client, 43
 client_device_sbo, 21
 client_import_file, 21
 client_sta, 19, 41
 CO, 53
 COM500*i* command purpose, 26
 COM500*i* command type, 26
 COM500*i* cross-reference information, 30
 COM500*i* scale, 25
 COM500*i* signal handling, command, 22, 26
 COM500*i* signal handling, indications, 22, 25
 Command Out, 53
 Commands Type, 22
 Communication starting, 39
 conclude_timeout, 27
 Configuration completion ICCP Configuration Utility, 37
 Configuration completion ICCP Configurator, 30

Control Command, 53
 Control Type, 22
 CSV file, 29
 Cyber security, 37, 51

D

Database Initialization, 49
 Data Discovery, 42, 44
 Data Set, 57
 Data set Name, 50
 Data Set Transfer Set, 57
 Data Type, client devices, 29
 Data Type, client indications, 29
 Data Type, indications, 25
 Data Value, 57
 Data Value (attribute), 55
 Data Value Discovery, 38, 57
 DD, 44
 Default DataTypes, 18, 21, 30
 Delete Rows, 20
 Device, 57
 Device Type, commands, 26
 DI, 49
 Diagnostics, 11
 DN, 50
 Domain names, 30
 DS, 50
 Dsts, 57
 Dsts control
 Dsts status (client), 50
 Dsts status (server), 50
 DV, 44, 55

E

Enable Communication check-box
 Check-box Enable Communication, 40
 Encryption, 51
 EX, 44
 Excel import, 29

F

Filtering, columns, 20
 Filtering, process objects, 30
 firewall, 41, 51

G

General Data Sending, 45
 Get tag (Remote), 54
 GS, 45
 GT, 54

H

hardening setting, 14, 41

I

IC, 46
 ICCP Configuration Utility, 31
 ICCP Data Type, 21
 ICCP name , 15
 ICCP Name, Data Values, 25, 29
 ICCP Name, Devices, 26, 29
 ICCP name mismatch, 38
 Id, 27, 42
 IEC61850, 51
 IEC 60870-5-101 master protocol, 7
 Import from Excel, 29
 IN-attribute, 31
 Info message, 46
 Info Message, 46
 Info Message Configuration, 46
 ini-file, 31, 42
 ini-file (C), 12, 50, 59
 IN Item Name, 12
 initiate_timeout, 27
 In Use, 44, 49
 IP Address Matching, 42
 IP-Listen setting, 32
 IU, 44, 49

L

LCC, 57
 lcc_name, 31
 license, 41
 LM, 48
 local_ar, 26, 41
 local_domain, 42
 domain names, 27
 LocalControlCenters, 18
 Log current State, 48
 Log file, 46
 Logging mask, 48
 LS, 48

M

Match IP Address, 37, 42

max_ind_pend, 27
 max_mms_msg_size, 27
 Maximum selection time, 26
 Max name length, 15
 MD, 46
 Message Directory
 Info message directory, 46
 MMS, 47, 57
 MMS Parameters, 32
 MVL, 47, 57

N

name
 bilateral table name, 27
 Naming rule, 15, 41
 naming rules, 25–26
 NODE_LICENSE_FAILURE, 41
 node_number, 21, 41
 normal_source, 21
 Normal Source, 25

O

OM, 45
 Operating Mode, 45

P

Process Object (client), 19
 process object group, 22
 Process Objects (server), 18
 project_directory, 21
 project_name, 21
 Projects, 20

R

RBE, 57
 RCC, 57
 read_only, 21
 Read ICCP Names from CSV, 20
 Read Only, 29
 redundant remote system, 32
 rem_xxx.xml, 38
 remote_ar, 26, 41
 remote_ar_2, 26
 remote_ar_3, 26
 remote_ar_4, 26
 remote_domain, 42
 domain names, 27
 Remote control center Name, 50
 RN, 12, 50, 59
 role_description, 27, 40–41
 runtime_directory, 21, 31, 39, 41

S

SA, 43
SA-Lib Control Type, 22
Save, 20
SBO, 21–22, 26, 29, 57
SCIL error 30014, 41
scope, 21
SD, 50
SDL, 47, 57
Secure Stack Configuration Utility, 37, 42
Security, 37, 51
select_time, 21
Select Before Operate, 22
Select-Before-Operate, 26, 29
Selection time, 26
Server, 43
server_import_file, 21
server_sta, 18, 41
Setpoint, 53
shadowing, 40
shortest_interval, 27
SNAP-Lite, 32
Special characters, 15
STA object type, 11
Start ICCP_LINK, 20, 40
Stop ICCP_LINK, 20
string based addressing, 12

Supported control types, 23
System Configuration Tool, 14

T

Tag
 Device Tag (Remote), 53
tag_enabled, 22
Tagging, 26, 57
Tagging enabled, 22, 26, 29
TCP port, 51
Terminology, 57
TG, 53
TLS, 51
Trace file, 46
Transport Layer Security, 51

V

version, 22
Virus scanner, 41

W

Write Configuration, 20, 30



Hitachi ABB Power Grids
Grid Automation Products
P.O. Box 688
FI-65101 Vaasa
Finland
hitachiabb-powergrids.com/microscadax

—

1MRK 511 508-UEN A/2021 © Copyright Hitachi Power Grids. All rights reserved.