

## Ethernet & IEC 61850

## Start Up

## Manual

---

### Table of Contents

---

What Awaits You	1
Communication in Motion	2
New on the Shelf: IEC 61850	3
Our Solution for You	4
The Proof is in the Example	5
Creating a Project and Structuring It	6
Routing Information in Communicators	7
Creating a CFC Function	8
Routing Infos between Communicators	9
Up to the Devices	10
Unveiled: The Control Center Report	11
What Else You Should Know	12
Looking Back	13

---

Release: 07.05.09

E50417-F1176-C324-A3

## Information for Your Safety

This manual does not represent a complete listing of all the safety measures required to operate the equipment (module, device) since specific operating conditions may make further measures necessary. However, it contains information which you have to observe in order to ensure your personal safety and in order to avoid property damage. The information is highlighted by a warning triangle and, depending on the degree of danger, is shown as follows:



### Warning

indicates that death, severe personal injury or substantial property damage can result if proper precautions are not taken.

### Caution

indicates that minor personal injury or property damage can result if proper precautions are not taken.

---



### Qualified personnel

Commissioning and operation of equipment (module, device) described in this manual may only be carried out by qualified personnel. Qualified personnel in the sense of the safety instructions in this manual are persons who are entitled to commission, enable, earth and identify devices, systems and circuits in accordance with the standards of safety technology.

### Use as prescribed

The equipment (device, module) may only be used for the applications described in the catalogue and the technical specifications and only in combination with third party equipment recommended or approved by Siemens.

The successful and safe operation of this device is dependent on proper handling, storage, installation, operation, and maintenance.

Hazardous voltages are present in parts of this electrical equipment during operation. Severe personal injury or property damage can result if the device is not handled properly.

- The device is to be earthed to the protective-earth terminal before any other connections are made.
- Hazardous voltages can arise in all the circuit parts connected to the power supply.
- Hazardous voltages can be present in the equipment even after the power supply voltage has been removed, i.e. capacitors can still be charged.
- Equipment with current transformer circuits may not be operated openly.

The limits specified in this manual or in the operating instructions respectively may not be exceeded. This point must also be observed during testing and commissioning.

---

### Disclaimer of Liability

We have checked the text of this manual against the hardware and software described. However, since deviations cannot be ruled out entirely, we do not accept liability for complete conformity or for any errors or omissions.

The information given in this document is reviewed regularly and any necessary corrections will be included in subsequent editions. We appreciate any suggested improvements.

Subject to technical modifications without notice.

Document version: 1.02.01

### Copyright

Copyright © Siemens AG 2009. All rights reserved.

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

### Registered Trademarks

SIPROTEC, SINAUT, SICAM and DIGSI are registered trademarks of Siemens AG. Other designations in this manual may be trademarks that if used by third parties for their own purposes may violate the rights of the owner.

# Table of Content

1	<b>What Awaits You</b> .....	1
2	<b>Communication in Motion</b> .....	3
3	<b>New on the Shelf: IEC 61850</b> .....	7
4	<b>Our Solution for You</b> .....	13
5	<b>The Proof is in the Example</b> .....	19
6	<b>Creating a Project and Structuring It</b> .....	21
7	<b>Routing Information in Communicators</b> .....	27
8	<b>Creating a CFC Function</b> .....	35
9	<b>Routing Infos between Communicators</b> .....	39
10	<b>Up to the Devices</b> .....	49
11	<b>Unveiled: The Control Center Report</b> .....	53
12	<b>What Else You Should Know</b> .....	57
13	<b>Looking Back</b> .....	61



## What Awaits You

Hello again - this time with the new Start-Up manual on Ethernet and IEC 61850. After the striking success with the first guide of this kind (on DIGSI 4) we have decided to once again give you a helping hand.

### **Weighty Problems**

With the help of the most recent standard IEC 61850, a universal solution for protective relaying, substitution and remote control engineering with regard to information exchange and data management is finally no longer a problem. The true (mental) problem lies more clearly in the bulk of the documentation on IEC 61850: Many hundred pages could lead you to believe that the authors were paid according to the weight of the delivered opus. In fact they did a very thorough job and thought everything through very precisely.

To extensively utilize communication according to IEC 61850, it is almost absolutely necessary to examine this voluminous work more closely. (True powerusers will be in heaven). It comes in handy then if we can present several essential concepts of the IEC 61850 briefly and concisely in this slim book.

### **Low Fat**

That alone is not everything. We will show you which solutions Siemens can offer you on this topic. And we have thought out another practically-related task which we will use to help you become familiar with the world of IEC 61850 in a very easy manner. We have embedded this task in additional information which will give you more insight. And finally there is the odd piece of interesting information for all those eager beavers amongst you. All that, once again, in compact form on a mere 60 pages.

The structure of our Start-Up manual is once again set up in such a way that you don't have to read it completely from cover to cover. On the contrary, you could, if you want, only work through our task. The instructions that are necessary for this are highlighted in color in the individual chapters and are therefore easy to find.

### **Your Mission, if You Choose to Accept It**

This time around, the task is located in its own chapter. This is because it takes up a little more room. We can tell you this much, you will be configuring a simple but effective application with which you will be able to test the functionality of communication between devices.

- 
- Great Expectations** Before you get started, though, we will give you a brief overview of what you can expect from this book. In Chapters 2 to 4, we will report on important and essential facts on the change in communication, on IEC 61850 and Ethernet and on solutions that we, at Siemens, have developed for you. Chapters 5 to 10 are dedicated to our application for communication testing whereby the task itself is presented in Chapter 5. Chapters 6 to 9 describe the implementation and in Chapter 10 you will proceed to the task at hand - the test. To solve the task, the different components of DIGSI 4 will once again be used: Manager, Device Editor, Configuration Matrix, CFC and of course the brand new System Configurator. In other words, a prearranged game once again to demonstrate the versatility of DIGSI 4. However, we won't go into great detail on the general procedures that are not IEC 61850-specific. For that, there is the Start-Up manual for DIGSI 4.
- Da Capo** Chapter 11 is a bonus for all those who are interested in substitution control engineering and its integration in an IEC 61850-based system. And finally, Chapter 12 provides you with the information that you always wanted to know.
- Looking Back** In Chapter 13 we have put all of your work on one page in the form of a compact sequence graph.
- Call Me** Should you have additional questions on DIGSI 4, our Hotline would be more than happy to help you:  
Tel.: 01 80 - 5 24 70 00  
Fax: 01 80 - 5 24 24 71  
e-mail: [support.energy@siemens.com](mailto:support.energy@siemens.com)
- Training** You can obtain the individual course offerings from our Training Center. There, Siemens offers you extensive courses on configuring and commissioning IEC 61850 systems and on DIGSI 4.  
Siemens AG  
Energy Sector  
Power Distribution Division  
Humboldtstr. 59  
90459 Nuremberg, Germany  
Tel.: +49 9 11/4 33-70 05  
Fax: +49 9 11/4 33-79 29  
Internet: [www.siemens.com/power-academy-td](http://www.siemens.com/power-academy-td)

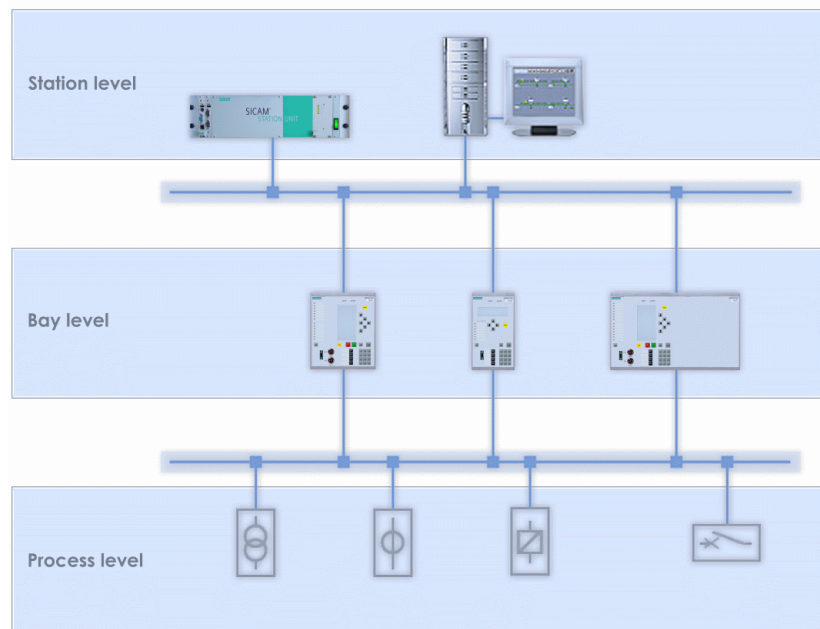
# Communication in Motion

# 2

For the selection of the transmission protocol, you always had many possibilities: IEC 60870-5-103, MODBUS, PROFIBUS FMS or PROFIBUS DP, only to name a few. SIPROTEC 4 and of course DIGSI 4 support all of these usual protocol standards. Everything's hunky-dorry then? Not necessarily, because the supposed advantage that you would assume with this variety very quickly proves to be a hindrance when you want to look beyond the edge of your plate.

## Home Game

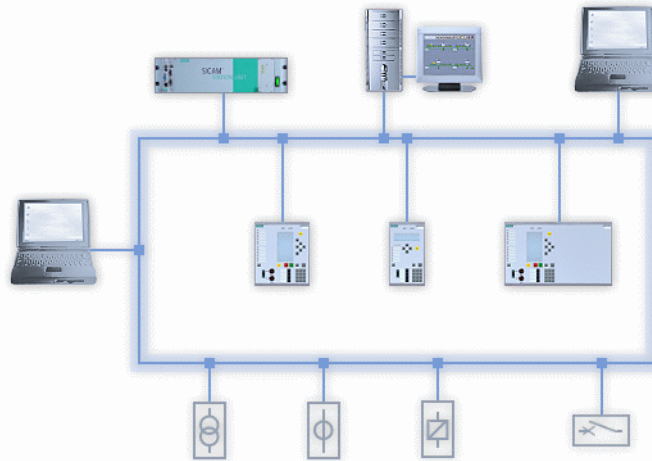
Many protocols are regionally limited and practically all are not compatible with one another. In view of an increasing concentration of the control of energy systems even beyond country borders (= the edge of your plate), one finds oneself confronted with a grave problem: protocols with respect to the contents transported with them must be conveyed to one another. This is not always possible without losses since information scope and processing are also different in the individual protocols. And it always has additional costs attached to it. The call for a cross-boundary, uniform standard was loud accordingly – and the call was heard. The new IEC 61850 standard is here and it promises a communication without boundaries.



Cross-level communication with IEC 61850

## Border Crossers

IEC 61850 crosses borders in different ways. It is the first and only standard that covers all levels of a switchgear system, that is, process, bay and station level. The individual devices communicate within a level or between any level - and that with one and the same protocol. The bus structure used for communication is flexible and can also be designed as redundant by means of a ring topology. By using TCP/IP, other services such as web services for remote maintenance can also be transmitted via the same communication network at the same time. Even operating relays with DIGSI 4 function on the same network.



The ring topology creates redundancy

## X Meets Y

The greatest advantage of IEC 61850, however, results from the standardization of services, descriptive language and interfaces to functions: the seamless interoperability of devices from different vendors! In this way, devices of Company X will in future be joined by those of Company Y and out of competitors will become friends that all use the same language - at least within switchgear systems.

## Flexible Term

To be sure, IEC 61850 is to be a future-oriented and above all future-proof standard. For that reason, the vertical communication up to the network control center is already being prepared today. As well, the expansion of the standard for devices in the fields of Power Quality and Power Metering are also being worked on. For it is obvious in any case: the fundamental principles according to which the communication is specified in IEC 61850 are not limited to electrical switchgear systems. They can just as well be applied to systems in the fields of wind energy, water or gas supply.



---

## **That's Ethernet**

Ethernet is used as the communication path, at least at this point in time. As you will discover in the next chapter, IEC 61850 is flexible enough to adapt itself to the trend in transmission technology without everything having to be thrown overboard at the slightest sign of change.

A look into the past explains the selection of Ethernet as the current communication medium. Already in 1976, Bob Metcalfe outlined the first designs of an Ethernet system. Back then, printer sharing by several users stood in the foreground. Today, Ethernet with a wide-spread popularity of more than 90 % is the most commonly used connection type between computers and their peripherals in a local network - and with that the mainstream communication medium of our time. Reason enough to focus on this transmission technology as far as IEC 61850 is concerned, especially since you can already fall back on a cost-effective product line and often also on an already existing infrastructure.

That Ethernet is highly backwards and forwards compatible, makes it particularly interesting with regards to present and future investments. In that way, a 10-MBit network can easily be integrated into the usual 100-MBit network of today - and this on the other hand in the 1-GBit network of tomorrow.

Next plus: the simple and standardized cabling in comparison to the use of parallel communication paths. A mix of different conductor technologies is possible: electric or optic, coax, twisted-pair or fiber-optic – all components that have long ago made their way into the office world, whose use is familiar and whose availability is secured. And last but not least, these components fulfill the EMC requirements that are demanded of protective ground and control system devices.



## New on the Shelf: IEC 61850

When you discover a new product of which there are already several of the same type, it's going to take more than the packaging go get you to buy it. You and we don't need an nth standard that may have wonderful properties in its own right. As soon as it needs to make a connection with the rest of the world, it starts to play cover-up. You also don't need a standard that can't keep up with the rest of the technical developments. And you need anything but a narrow-band niche solution.

### Wanted

Let us formulate several demands that a new standard must fulfill in order for it to be able to wave "I want to be an international standard" on its banners.

- We demand the independence of technological trends in system communication!
- We demand seamless interoperability between devices with different functionalities and from different vendors!
- We demand the feasibility for cross-vendor exchange of configuration data!

So, let's see what IEC 61850 can offer us!

### Together and Yet Apart

Let's start with the independency of technological trends in system communication. Currently, it makes sense to use Ethernet as the communications medium in connection with IEC 61850. Should a newer, better way for the transmission of data arise in the future, it will also be possible to use this one without having to say goodbye to IEC 61850. So that this can work, IEC 61850 strictly separates data and services from the required communication technology, which is practically unheard of in any other protocol.



A mapping whose structure and syntax is also specified by IEC 61850 makes the connection. Picture this mapping as a driver for a particular PC peripheral, say for a printer. When you replace this, you wouldn't come up with the idea of also replacing the PC. Instead you would merely replace the driver. The rest of the system would remain unchanged.

With reference to our case this means all applications, all databases as well as the general communications functions can continue to be used even if the communications technology should change. This ensures a long-term stability for the standard beyond many communications technologies and our first demand is fulfilled.

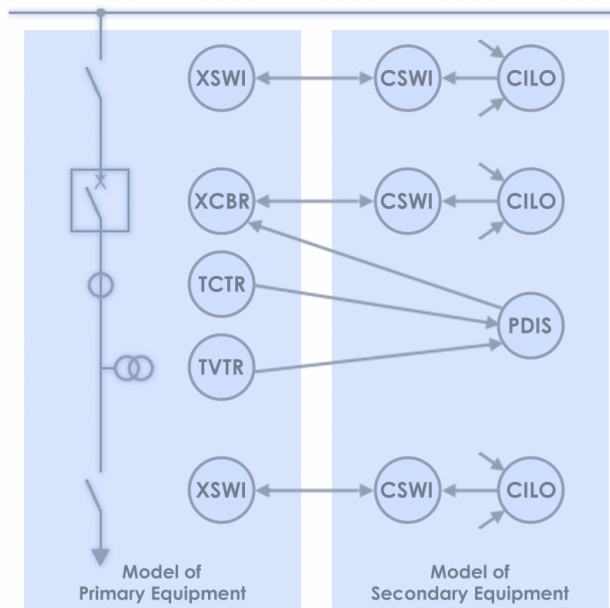
## International Cuisine

Now for interoperability. An interesting word! The approach that the standard takes is to standardize as much as is necessary and as little as possible. In the international sphere everyone eats what is on the table, but alone in the kitchen everyone can cook up their own specialities.

At this point, to avoid a possible misconception you must understand that IEC 61850 does not standardize any functions or their technological conversion respectively. It is up to each vendor to make his own selection of a particular software algorithm. No one has to forego any experiences they have accumulated.

On the contrary, IEC 61850 breaks down known entire functions into the smallest possible function units and standardizes the interfaces to these function units. These interfaces are interfaces for input data, for output data and for data that are required for configuration. A function unit including its interfaces is given the name **Logical Node**. Examples: PDIS for distance protection (DIS = Distance), XCBR for circuit breaker (CBR = Circuit Breaker) or CILO for interlock (ILO = Interlocking). From these logical nodes, the application-specific functions are then combined or modelled.

Today, as a rule, information from and to the process still passes through traditional I/O cards that are integrated in the field devices. In future, it will also be possible to separate the input / output interfaces from the devices. These will then be connected to the IEDs via a process bus as independent actuators or sensors respectively.



Application-specific functions are modelled from standardized nodes

---

## Pottery Class

More than ninety such nodes are uniquely described in IEC 61850 and thus already supply a wide base for almost all requirements. We would like to use a simple branch as shown in the figure above to demonstrate how application-specific functions are modelled from this repertoire of logical nodes. We have written the names of the nodes used in brackets, next to the respective equipment.

The data image of the two isolators (XSWI), the circuit breaker (XCBR) as well as a phase of the current or voltage transformer (TCTR or TVTR) are each represented by a logical node. Together with the equipment you have the model of primary equipment.

On the secondary equipment side, nodes of the **CILO** type check the interlock conditions and, if necessary, provide an enable for the nodes of the **CSWI** type. These then send switching commands to the isolators and the circuit breaker. The data of the transducers and voltages is passed to the **PDIS** node. This represents the distance protection and can initiate a trigger of the circuit breaker when there is a failure.

## Peer Pressure

The standard groups logical nodes under functional aspects in logical devices (LD), in our example - control and protection. The nodes of the primary equipment model are thereby assigned to the input or output interfaces. Other logical devices are, for example, measured-value acquisition or line fault recording. Logical devices are of course always embedded in tangible devices. A SIPROTEC 4 multifunction device, for example, also contains, as a rule, several different logical devices. The point, however, is that the logical devices required for a complete application can be wisely divided amongst different tangible devices, of course also amongst devices of different vendors.

## One-on-One Info

Which functions are grouped in which way within a device is once again left up to the individual vendor. However, IEC 61850 demands a profile from each standard-conforming device that gives an adequate description for others. This profile is also known as a device description or as an ICD (IED Configuration Description). The ICD contains all configuration data necessary to be able to integrate the device into a complete communication system. As a logical consequence of the demand for interoperability, this description must be written in a language that everyone understands and that is applicable for everyone.

The key to cross-vendor exchange of configuration data is then a standardized descriptive language. For this, SCL, the Substation Automation Configuration Description Language was developed. With this language, all relevant information can be written following a standardized structure based on XML into a text file.

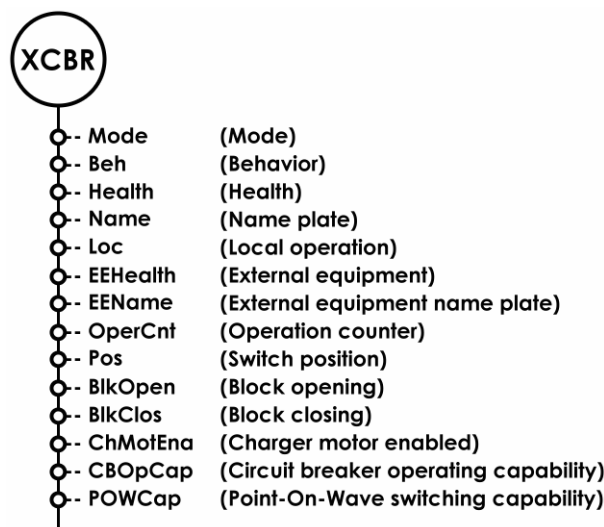
## Total Picture

The ICD describes the configuration of a device as well as its function and communication capabilities with regard to IEC 61850. Similarly, the so-called station description or SCD (Station Configuration Description) contains information on the network structure of a station (in other words, switchgear system), on the assignment of the devices to the primary equipment as well as on the internal station communication.

This SCD is generated with a configuration software that is known as the Systemkonfigurator . Of course, all information is documented in SCL at the same time and can for that reason be exchanged between different configuration tools without a problem.

## Objects and More

But how is the internal station communication implemented? In other words, the exchange of information between individual devices? Here, as well, the standard intercepts with its rules and divides all of the data that flows through a logical node into exactly defined data objects. Let's illustrate this with an example. We'll take the node **XCBR** to heart (exactly, the one for the circuit breaker). In the following figure you can see this node and its subordinate data objects. These are, amongst others, data objects for the operating mode (Mode) or the position (Pos).



To each Logical Node its Data Objects

So that data can now be exchanged between individual devices, you link two data objects of logical nodes from different devices with one another. You do this with the help of the DIGSI System Configurator, the already mentioned configuration software. More on this in the next chapters.

## Path Finder

According to IEC 61850, an information such as the data object is always represented based on a complete path with the following syntax:

**Logical Device (LD) / Logical Node (LN) / Data Object (DO).**

Since the logical node XCBR is assigned to the logical device CTRL, the following format results for the position of the circuit breaker:

CTRL / XCBR / POS.

---

**Good Points**

But there's more. Each data object also has, as a rule, so-called data attributes such as the actual value of the object or the unit of the value. CTRL / XCBR / POS therefore has data attributes that all refer to the position of the circuit breaker. An attribute is, for example, **ctlVal** (control value). By changing the value stored in this attribute, you can open and close the circuit breaker. Another example: the attribute **stVal** (status value) is transmitted to signal the status of the circuit breaker.

**At Your Service**

Even the way data is accessed follows a uniform standard. Services were defined, for example, for the reading and writing of a value or the switching of a device. Even the event-driven reporting, for example after changes or errors, is one of these services. You will discover more on this reporting in Chapter 11. There is one service each for the transmitting of fault line reporting and for time synchronization per SNTP. IEC 61850 consequently uses standard services as a base and maps these via Ethernet.

Up until now, we have kept secret the most important service for our sample application: GOOSE - Generic Object Oriented Substation Event. With this service, information on events can be quickly transmitted between different devices. And that is exactly what we need for our example.





## Our Solution for You

With Ethernet and IEC 61850, it's child's play to set up a direct connection between PC and protective equipment. In the device's Properties you define an IP address and a subnet mask and then select the usual path for the connection set up - and voilà, you're communicating via a direct Ethernet connection! (OK, there are a few things that you still have to take into consideration. You will find a few representative explanations in Chapter 12.)

### A Question of Conscience

But why should you procure a new device or upgrade your already existing one just for this? After all, you've been managing this type of communication quite well with the tried-and-true on-board systems such as the direct connection by means of serial cable or even a modem connection. Look at this opportunity as simply another option if the reason for your rethinking is another: you want an easy to expand network structure with devices that can communicate with one another, with devices from different vendors and functionality. But also the desire to have a much faster transmission of data between PC and SIPROTEC 4 device can also be a good reason, maybe even the most important. Be it as it may, we'll now show you what Siemens offers you as a solution so that you can professionally utilize IEC 61850 for yourself.

### A Question of Type

Right from the top you need a SIPROTEC 4 device. And so that this is also suitable for communication according to IEC 61850, its firmware must be Version 4.6 or, of course, higher. If you now decide (we are currently at the beginning of the year 2005) for a device of the type **7SA52**, **7SA6**, **7SJ61**, **7SJ62**, **7SJ63**, **7SJ64**, **6MD63**, **6MD663** or **6MD664**, then this prerequisite is already fulfilled. Further device types are, to be sure, in the pipeline. Thus, for example, **7UM62**, **7UT613**, **7UT633** and **7UT635** will make their appearance during the course of 2005 as well as **7SD52** and **7SD610** during 2006. Your personal Siemens sales partner can provide you with all the most current information.

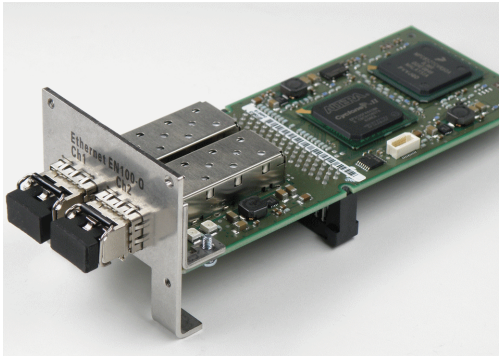
### A Question of Age

If you've already had a SIPROTEC 4 device in your possession for some time all is not necessarily lost. It needn't mean that every technical further development will bypass you and leave you out in the cold. With a firmware update, you can bring many a device up to par. Have a look on the Internet under [www.siprotec.com](http://www.siprotec.com) to find out whether your device is suitable for this rejuvenating cure and how you have to go about it. Behind the entry **Devices** you'll not only find all the necessary information but also the most current firmware updates and device drivers for DIGSI 4.

## Paving the Contact Way

What is a device without the appropriate interface to the outside? That's way there is now the new module EN100 that is designed for communication according to IEC 61850 using Ethernet.

If you order a current SIPROTEC 4 device with the option IEC 61850, the EN100 module is of course already installed in the device and is ready to use so to speak. You don't have to get your hands messy with screws or with bits or bytes.



Double the pleasure: The optical interfaces of the EN100 are there twice

Since we only just extolled the virtues of a firmware update a few short sentences ago, it is of course also possible as a consequence to purchase EN100 modules separately. A description on how you install this module and then use it is found in a separate manual about the module. You can download this manual from the [www.siprotec.com](http://www.siprotec.com) website. In addition, you will find a nicely animated instruction that shows you the fundamental procedures for swapping communication modules on the CD "SIPROTEC 4 You – Start Up".

Should you be one of those who ignores instructions on principle, then we request that you at least follow the following sequence:

1. Remove the prevailing module
2. Update the device firmware to Version 4.6
3. Install the EN100 module
4. Load the module firmware into the device

(That you don't work on the open device while it is live-wired is a foregone conclusion, or isn't it?)

Should you mix up this sequence, it could happen that after the module is installed the device can no longer be operated; removing the module with the subsequent downgrade is then the consequence and last but not least, the expense is considerably higher than necessary.

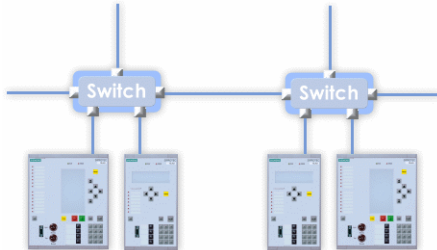
## Offshoots

The module consists of a single board that is mounted on Port B of the device, in other words, the system interface. There is an electrical connection version of the module for built-in and surface-mounted housings. For devices in the built-in housing, an optic version is also available.

One thing that is the same in all versions is that the physical interface is there twice. This can be clearly seen in the picture above. Only one of the two interfaces is active at any time, the other is passively monitored. If there is a fault in the active interface, a switch is automatically made to the passive one in just a few short milliseconds. This feature allows you to easily set up redundant structures.

## Don't Sweat It, Just Switch It

As soon as you want to connect more than two communicators with one another, you require a switch. This is an intelligent data distributor to which several devices can be connected, much like a star coupler. A star network structure results. It is always a point-to-point connection when the devices are connected to the star point (= switch).

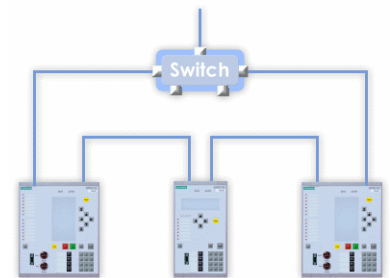


With good switches, you won't break out into a sweat even with larger structures

The switch handles the management of incoming and outgoing information. Unlike a hub that distributes the available bandwidth in equal parts to the available connections, a switch is much more clever. The switch takes the actual data traffic into consideration and dynamically adjusts the data transfer rate of each connection. That way there is no data 'stau' (also known as traffic jam).

If you have decided on EN100 modules in the optical version, then you are killing two birds with one stone. With this version, a switch is integrated directly on the module. That way, at least, you save on external switches when connecting SIPROTEC 4 devices with one another.

Insofar as you still need the one or the other switch, we don't know what the friendly fellow at the IT discounter will recommend. For communication problems, we recommend switches and routers by RUGGEDCOM. We have made many tests with these and have gathered positive know-how and they are sufficient for the EMC requirements in the switchgear system.



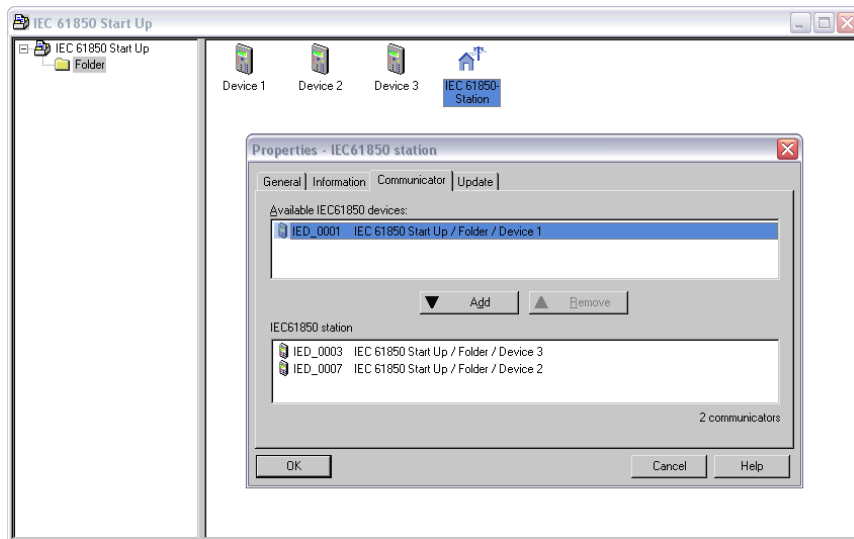
Time saver: The optical EN100 module has an integrated switch

## Next Station: IEC 61850

New hardware, new software! This is true even for DIGSI 4: It has to be at least Version 4.6! As of this version DIGSI 4 supports IEC 61850-conforming SIPROTEC 4 devices and everything that goes along with it, for example, the IEC 61850 station. This new element in the DIGSI Manager represents a switchgear system in which all devices communicate with one another according to IEC 61850 using Ethernet. You conveniently select, as you will find out in detail later on, which devices these specifically are from a list and enter them as so-called communicators in the IEC 61850 station.

The communicators are not limited to SIPROTEC 4 devices. DIGSI 4 of course fulfills the requirements of the standard and gives you the opportunity to integrate IEC 61850-conforming devices with different functionality and from different vendors in a project.

For this, you import the device description of the respective device and then add this device as a communicator of the station - in the same way as you do for SIPROTEC 4 devices.



You use a selection list to define the communicators of an IEC 61850 station

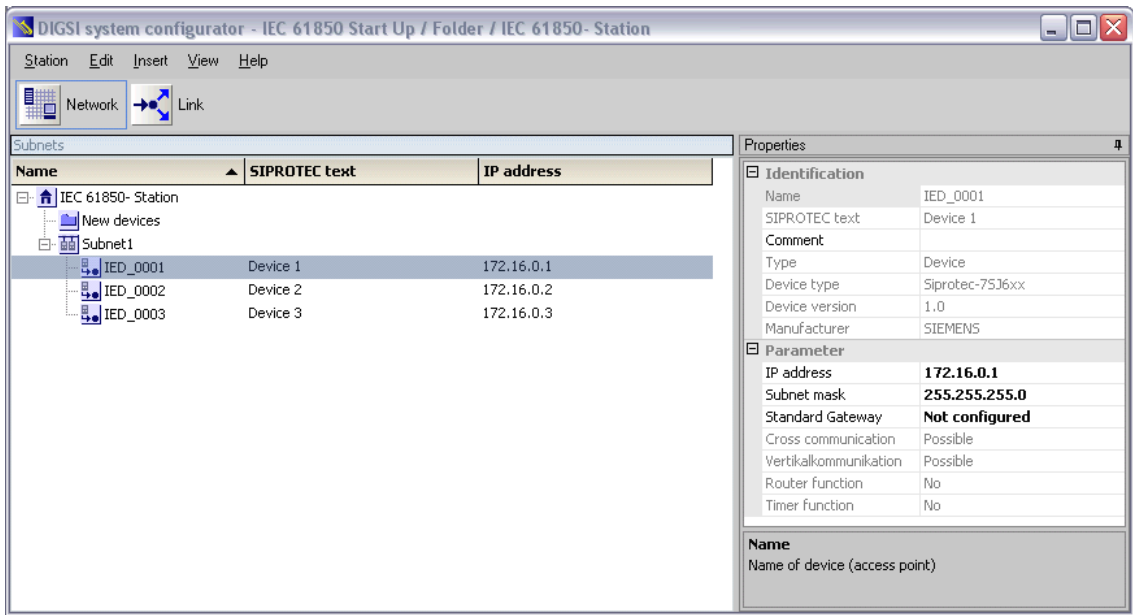
Provision is also made for the export of station and device descriptions. With a few mouse-clicks, you store the data in files - naturally, absolutely standard-conforming in SCL! That way, they can be further processed directly by an automation system such as SICAM PAS or can be evaluated by configuration tools from other vendors.

The station element is also the entry to information on communication connections and data links between the communicators. This information is processed with a special module, the DIGSI System Configurator. As you are already used to with the CFC or Display Editor, the DIGSI System Configurator is also seamlessly integrated in DIGSI 4. A double-click on the Station symbol starts it and the station data is loaded.

## Smooth Configurator

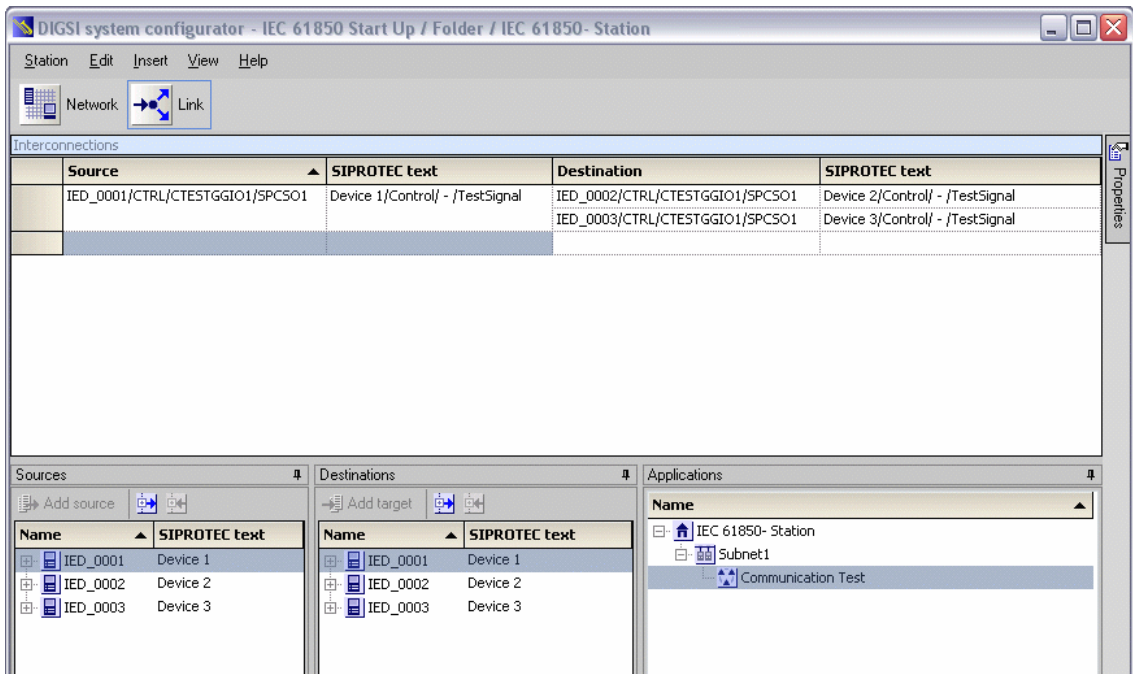
With the DIGSI System Configurator you define the network structure as well as the communication properties of the communicators and subnets and link data objects of individual communicators. The DIGSI System Configurator is divided into two work areas that also correspond to two basic task areas: Network and Link.

In the **Network** area you structure, as required, the communication network of the IEC 61850 station by inserting or deleting subnets. You assign communicators which you previously selected for the station in the DIGSI Manager to these subnets. You define the communicator addressing as well as other values of relevant properties.



The current network from the point of view of the DIGSI System Configurator

In the **Link** area, you define which communicators exchange which data with one another. For this you link data objects of individual communicators. That means, you link information between a source and a destination. So-called applications are used for the structuring of the links. An application is a group of links to a specific application purpose such as an interlock or a measured-value task.



Here, contacts are made: The **Link** area of the DIGSI System Configurator

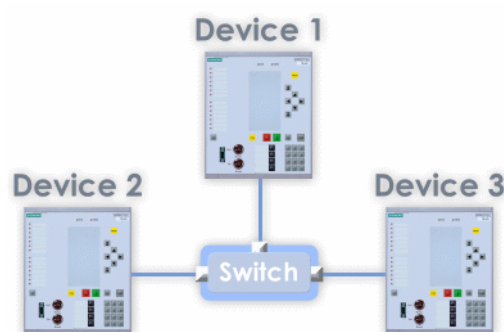


## The Proof is in the Example

Instead of a difficult example that will make you want to drop out, we offer you a simple example to drop in. We'll show you a simple and at the same time effective method with which you can check whether the communication between the communicators exists. This method also comes in very handy in real life. For what we are about to demonstrate with only three devices can, in principle, be expanded to any number of communicators.

### Mission Possible

We're going to start from a configuration with three devices that are connected with one another via Ethernet. As you have already discovered in Chapter 4, you need a switch when there are more than two communicators. With that, we have the setup of an experiment as shown in the following figure.



Three-way conference: The devices communicate with one another via a switch

The communication test itself begins by pressing the F1 key on Device 1. The illumination of LED 1 on the same device confirms this press. What is important here is that an indication is sent to the two other devices at the same time. An LED on each of these two devices is then to signal that it has received the indication. Pressing the F2 key on Device 1 cancels the indication and the LEDs on all devices are reset.

### Wire & Co.

To find a conventional solution using wiring as an equivalent is not so easy, in fact, it is not possible. This is because you wouldn't be going through the system interface. Instead, you would be going from a binary output on Device 1 to several binary inputs on Devices 2 and 3.

That way you create a bowl of spaghetti that always has the same taste. This would mean that with this one elaborate cabling you would only have the benefit of testing the cabling alone but not the fundamental communications capability between the devices. For each application you would need a separate cabling. The benefit of the solution Ethernet plus IEC 61850 becomes all the more clear: Only one cable for each communicator for all data to all applications!

---

## Shopping List

First of all, you need three SIPROTEC 4 devices. In our example, we are going to make use of three identical devices of the type **7SJ63**. The device types don't have to be identical, of course, it simply makes life easier for the author.

If, as a critical practitioner, you would like to verify the methods in your own experiment, then of course you would take the same devices in DIGSI 4 that you already have available to you. What is important, however, is that the devices you use are at least of Version 4.6. All devices below this don't know what to do with IEC 61850.

Likewise, you will need the communication module EN100. Which connection technology you decide, either electrical or optical, makes no difference to the functionality.

Last but not least, procure a switch whose connection fits the EN100 modules and the appropriate cable material (Optical fiber or Ethernet patch cable - Category 5). We assume that you have already purchased DIGSI 4.6 in the Professional Version plus the DIGSI System Configurator. And with that, lots of fun and success with the following exercise.



# Creating a Project and Structuring It

# 6

First of all take a deep breath and lean back! For in spite of the addition of IEC 61850, configuring with DIGSI 4 remains as simple as ever. Now we'll take you through the compulsaries ...

## Work ...

... for we find ourselves at the beginning of our task. In this chapter we will explain the first steps on the way to our goal:

- ❑ Step 1: Create a project
- ❑ Step 2: Insert SIPROTEC 4 devices
- ❑ Step 3: Insert IEC 61850 station
- ❑ Step 4: Enter devices as communicators in the IEC 61850 station

If you're only interested in this first part of our task, then jump directly to Page 24 to the text in the colored background.

## ... and Pleasure

For those of you who have a little more time, you will also find additional answers in this chapter to the following questions:

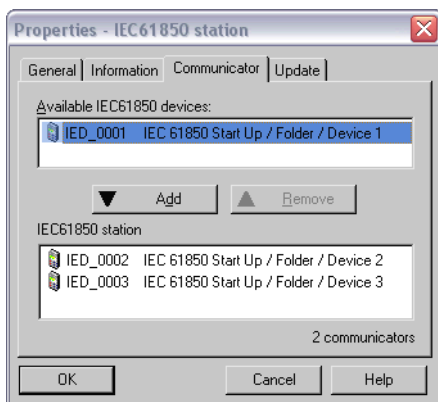
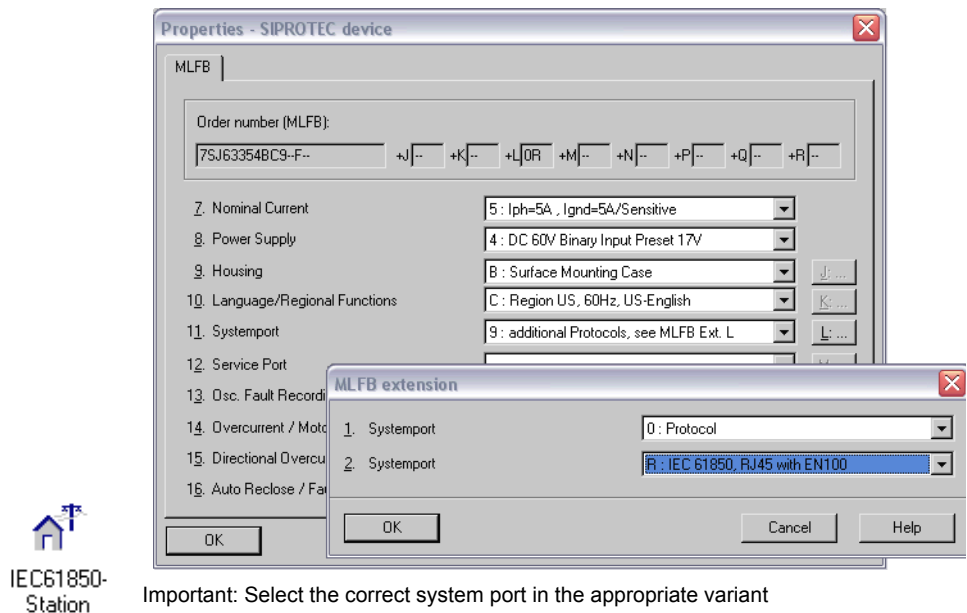
- ❑ Which ingredients are necessary for a functioning communication according to IEC 61850?
- ❑ How do I integrate devices with other functionalities and from other vendors into a project?
- ❑ Can I simultaneously manage devices with different protocols within a project?

## First You Take

Let's talk about the ingredients that you need for a communication via Ethernet and IEC 61850. At the top of the list are the IEC 61850-capable devices. So that we can even talk about cross-device communication, there should at least be two. If they are SIPROTEC 4 devices, insert them, just like all other SIPROTEC devices, using drag & drop from the Device Catalog into the project.

Please always select the MLFB number that is stamped on your device - even if you should have exchanged the original communications module for an EN100 module in the mean time. In this case, first select the identifying letter for the original port. Then, open the Properties dialog of the inserted device and select the module **EN100** as the port in the **Communication Modules** tab.

Likewise, there is no fundamental difference in the operation when entering the order identifier. What is important, is that you select the correct system port. For this type of communication you require a port of the type **EN100**, that is currently available in an electrical and an optical variant.



Join the party: You select the communicators with a mouse click

New in the hit-parade of project elements is the IEC 61850 Station, which you insert just like all other candidates via the menu. One such station unites several devices that can communicate with one another via Ethernet according to IEC 61850.

Which devices these are, in particular, can easily be defined by you in the station's Properties dialog. There, the names of the devices that come into question are displayed as potential communicators. With just a few mouse clicks, you assign these to the station. You can see the dialog for this in the picture to the left.

---

The station element is also the gateway to information about communication connections and data links between the communicators. In Chapter 9, we will show you how you edit this with the DIGSI System Configurator.

### Open for Others

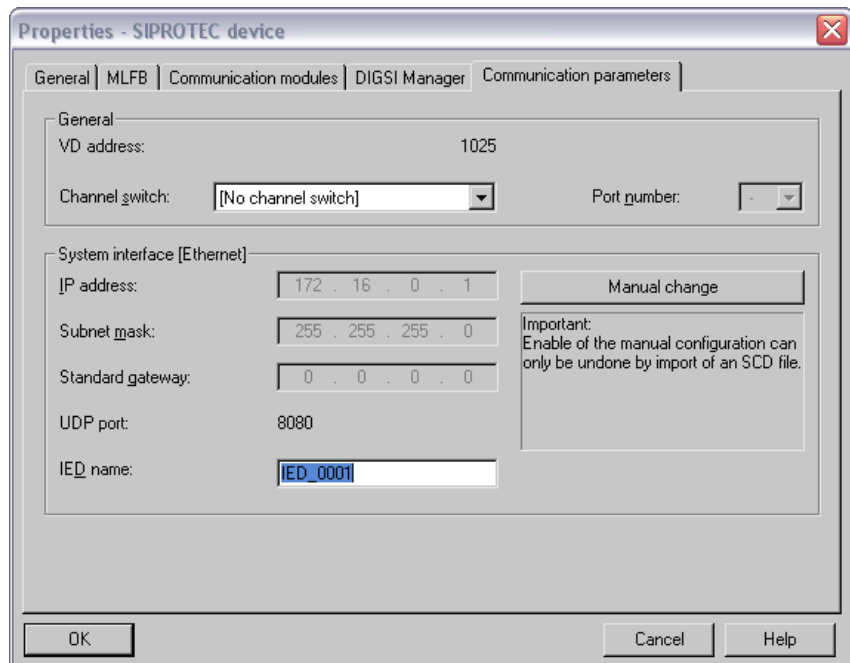
With DIGSI as of Version 4.6, we make it possible for you to integrate IEC 61850-conforming devices with different functionalities and even from other vendors into a project and with that in a switchgear system. For this, you import the device's description and then add it as a communicator of the station - just as you do for the SIPROTEC 4 devices.

An example for another communicator would be a time server (SNTP Server), that provides a common time base for all communicators.

We even make it really easy for you because you insert a different IEC 61850 communicator into the project structure just like any other object. Simply select **Insert new object** → **Other IEC 61850 communicator** from the shortcut menu. This opens a familiar File dialog. With this, you select the device description in the format **.icd**. An icon for the communicator is inserted in the project. This is not unlike a normal device icon, even if it is somewhat palefaced.

### Who's Who?

Within an IEC 61850 subnet, each communicator requires a unique name. One such name may only consist of a maximum of eight letters or numbers. No 'umlauts' (modified vowels) or spaces are allowed and the first character must be a letter. Since we didn't want to restrict the essentially free handling in the assignment of a device name within a project, there was only one solution: There had to be an additional name, the so-called IED name.



The IED name is part of the communication parameters of a SIPROTEC 4 device and is used in IEC 61850 as the device name

---

As you can see in the picture above, you define this IED name in the Properties dialog of the respective device. If, however, you're not fussy about the name, you don't have to rack your brains about it. The DIGSI Manager automatically defines a unique name. This consists of the prefix **IED\_** and a four digit consecutive number.

## Multicultural

For our example, we will create a new project. This is not to say that you can't also integrate an IEC 61850-conforming communication via Ethernet into an already existing project. It goes without saying that devices with different protocols or that use different transmission paths for communication can be included in the same project and thus in the same system, but these devices simply cannot be integrated in an IEC 61850 station.

## Down to Work

After these general but necessary implementations, we now serve you the first part of our task as a reward. During this part you will create a new project and insert SIPROTEC 4 devices and an IEC 61850 station.

- Click on **File** → **New**. In the dialog box **New**, enter **IEC 61850 StartUp** as the project name, for example. Then click **OK** and the DIGSI Manager already generates a new project window. The List View already conveniently contains a folder. We will leave its apt name - **Folder**, since we don't really want to spend any more time on the topology of the project. Instead, you should insert three devices and a station in the project as quickly as possible.
- To do so, right-click on the folder. From the shortcut menu, select **Insert new object** → **SIPROTEC device** to display the device catalog. One after the other, open the folders **SIPROTEC Devices**, **7SJ-overcur./motor prot.** and **7SJ633**. In the last folder you will find, among other things, an entry for Version 4.6. Click on this entry and keeping the mouse button pressed down, drag the entry onto the folder.
- Before the SIPROTEC 4 device is placed in the desired position, you must define the device model in DIGSI, just like you would for any other SIPROTEC devices. To do so, the **MLFB** tab in the dialog box **Properties - SIPROTEC 4 Device** is displayed. Which settings you finally select is not really relevant for our example. After all, we're not going to worry about special parameters and functions. What is essential is the selection of the system port. Select the setting **additional Protocols**. This selection opens a second dialog with two drop-down lists. In the upper list, select the setting **Protocol**. In the lower list, select **IEC 61850** with the required port variant.
- As soon as you have closed both dialog boxes, an icon for the SIPROTEC 4 device is displayed. To speed up the process just a bit for the other two devices, we will make use of the functions **Copy** and **Insert**. This means that you select the device icon that was just inserted, copy it onto the clipboard using the shortcut menu, for example, and then inserting it twice. Just on the side: In our example we are inserting all three devices into one and the same folder simply for a better overview. The participating devices can of course be scattered anywhere within the project.

- 
- Now please change the names of the three devices to **Device 1**, **Device 2** and **Device 3** so that they fit appropriately to the automatically assigned IED names. This has no influence on the later communication, it just makes it clearer to follow. You can easily check the IED names and, if necessary, also change them. Simply right-click on a device icon to open the shortcut menu and select **Object Properties**. In the **Communication parameters** tab of the Properties dialog you will find the parameter **IED name**, which you may edit taking the already described rules into consideration. You could, for example, enter the names **Device1**, **Device2** and **Device3** there.
  - Now we don't want to leave our three devices out in the cold any longer. Instead, we want to make sure that they finally have a roof over their heads. To do this, you have to insert an IEC 61850 station. This is done once again with the shortcut menu, **Insert new object** → **IEC 61850 Station**. And voilà, a cute little house with its own powerline appears in the project. This isn't enough though, for no one (except you and us) knows that these three devices are to communicate with one another within the station. Therefore, you have to enter the devices in the station as communicators.
  - Open the station's Properties dialog and select the **Communicator** tab. Since our three devices fulfill all the requirements for an IEC 61850-conforming communication, they are already displayed as potential communicators. Now you just have to accept them into the list of actual communicators. This is done with a double-click on each individual device name one after the other. Alternatively, you can simultaneously select the names of all three devices and click **Add**. A quick click on **OK** closes the dialog box and the first step is already completed.

Now that was easy, wasn't it? Then off to the next chapter!



# Routing Information in Communicators

# 7

Our method for checking communication requires the exchange of indications between the participating devices. You define which indications, how they are triggered and where they go - separately for each device - with the Configuration Matrix.

## Work ...

In the second part of our task, to which we welcome you most graciously, you will discover,

- ❑ how you insert the required user-defined indications in an equally user-defined information group,
- ❑ to which sources and destinations you must route this information and
- ❑ how you assign particular information to IEC 61850 data objects.

If that's enough for now, turn to Page 31 and get going.

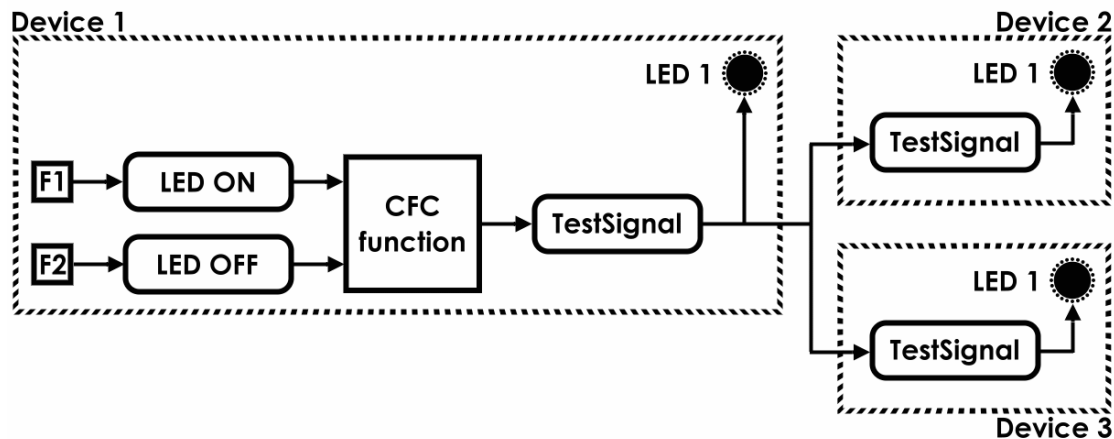
## ... and Pleasure

For the rest of you, we will first enlighten you on the following correlations:

- ❑ How do the required indications relate to our defined problem?
- ❑ How do the function keys relate to a CFC function?
- ❑ How does the information routed to the system interface relate to the IEC 61850 data objects?

## Indications are...

Before we dive into things, let's take a look at which indications are required for which device and how you have to route them. We have prepared a small but surely helpful graphic. By the way: all identifiers that we have used here are simply suggestions. If you would rather call your indications Mutt and Jeff, no one is going to stop you. The only thing you have to consider is the limitation to a certain number of characters since our programming people like to turn every byte over twice (You know: a byte saved is a byte earned).



Clearly arranged: Indications in the devices

## Messenger

As soon as you press the F1 key on Device 1, an indication, which we will call **TestSignal** in our example, is to be sent via the system interface. The indication is to continuously exist until it is explicitly canceled. You could keep your finger pressed on the function key for as long as you like or you could lay the device down horizontally and place a weight on the key. It wouldn't matter because the key only responds with a short signal no matter how long you press it. For that reason, we chose to activate the indication **TestSignal** with the F1 key and to cancel the indication with the F2 key as our solution.

## Little Helper

A simple CFC function is to help us. We will configure it as we go along but right now we will display it as a secretive blackbox. This CFC function detects the short signal and makes sure that the indication **TestSignal** exists until the F2 key is pressed.

## Links

So that our function can even decide something, it must first know which key was pressed. As a link between the keys and the function, we therefore need two additional indications. In our case they are **LED ON** and **LED OFF**. These two indications serve the CFC function as input signals and are sent as soon as the associated function key is pressed.



## Let There be Light

Oh yes, and before we forget, we also have to route the indication **TestSignal** to LED 1 of Device 1, of course unlatched. This causes the LED to light up, however, only as long as the indication actually exists. That means that after the F2 key is pressed, LED 1 on Device 1 goes dark.

That's all we have to do for Device 1. Now on to the receiving devices Device 2 and Device 3.

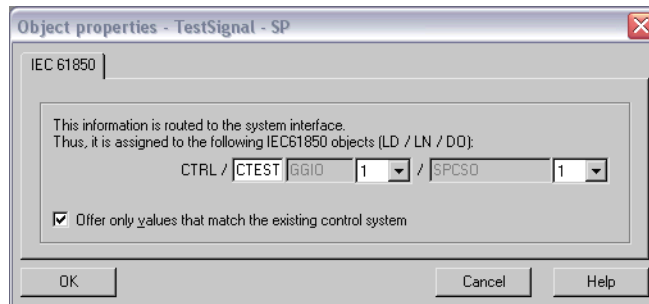
## The Object's the Thing

Imagine you would press the F1 key. The indication **TestSignal** starts its journey through the Ethernet cable. Of course, an actual text message is not transmitted, rather only a data object which is equivalent to our indication. This data object is, in turn, assigned to an indication in each of the other two devices. In our example we will call the indication **Indication from Device 1**. As soon and as long as Devices 2 and 3 receive the data object, the indication exists in these devices. Now we only have to route this indication to LED 1 in each of the two devices and our communication's check is complete.

In Chapter 3 you already learned that logical nodes are made up of individual data objects and that the entire communication between different communicators is only possible through the linking of data objects. Consequently, the transportation of a user-defined indication from one communicator to another must take place in some form via data objects.

## Common Denominator

Unlike the perennial question of which came first, the chicken or the egg, in our case it is very clear: first there was user or vendor-specific indications, and then the attempt to bring everything down to one denominator. So that you don't have to say good-bye to things you are familiar with, you can continue to create user-defined indications in DIGSI 4. Only when you route these indications to the system interface do you have to assign the respective information to an IEC 61850-conforming data object.



Assigning to IEC 61850 data objects: DIGSI 4 shows the way

---

## Relationship Wizard

For all that, DIGSI 4 does a lot of the work for you (see picture above). For as soon as you route a self-defined information to one of the columns for the system interface, a dialog box appears and in it the basic components of the path are already specified for the respective information. You will remember: According to IEC 61850 an information is represented based on a complete path with the following syntax:

### **Logical Device (LD) / Logical Node (LN) / Data Object (DO).**

Since the assignment of an information to the individual components of the path is defined through the standard, DIGSI 4 displays this without you having to worry about it. You only have to define an individual prefix for the logical node and select, if necessary, instance numbers, for the logical node and the data object respectively. Prefix? Instance number?

## About Prefixes ...

With different prefixes, you group information according to functional aspects which you can then find easier in the DIGSI System Configurator. One thing is for sure, you're going to have to get use to the fact that a multitude of individual information awaits you there. If, for example, you would like to configure a switchgear interlocking, you could assign the prefix **SI** (= **S**witchgear **I**nterlocking), for example, for all user-defined information required for this. A prefix may consist of a maximum of 6 characters; special characters such as (;-) are once again not allowed.

## ... and Instances

You surely already know the term 'instance' from one or the other program on your PC which you can start repeatedly, in other words in several instances, even though the program code is only physically installed on the computer once. In the same way, a logical node can also be used repeatedly. How often, is initially specified through the number of available instances. The instance, nevertheless, refers to the combination of logical node and prefix. As a result, the frequency increases through the use of different prefixes.

For each instance, there are a number of data objects available depending on the type of logical node. If this amount is used up the next higher instance number for the logical node is automatically selected. But even a data object can be used in several instances. For that reason, you can quickly see that you don't have to worry right now about any possible limitations in the assignment of information to data objects.

## Our Recommendation

By the way, DIGSI 4 suggests an instance number and in the process also takes into account the prefix that is entered. Our recommendation for simple and effective working: allocate good prefix texts and let DIGSI 4 do the rest. And another thing: DIGSI 4 remembers all of the complete paths of IEC 61850 objects that have, up until now, been assigned user-defined information within a project. Even if one such information has been deleted from the configuration matrix, the associated path continues to be considered as already allocated. This is to prevent two different pieces of information from being simultaneously assigned to the same IEC 61850 object, which in turn could result in error interpretations with the rest of the communicators in a station.

---

If the option **Offer only values that ...** is selected in the dialog that is displayed, then only those instances that have not been allocated up until now are offered for selection. A tip from a professional: You should only deselect this option if you are sure that modified assignments are also updated with the rest of the communicators.

### Exception to the Rule

The assignment to a particular logical node depends of course on the type of information. Certain information, such as commands, is assigned to several logical nodes. It is therefore quite possible that several standard identifiers are displayed in the assignment dialog. This changes nothing for you: You allocate only one (common) prefix and one (common) instance.

### Down to Work

In the first section of this second part, we will show you how you define, name and then route all indications for Device 1. The explanations for Devices 2 and 3 follow after a short intermezzo in a second section.

- First of all, open Device 1. The fastest way to do this is to double-click the device's icon. In the dialog for selecting the type of connection, select the option **Offline**. Now click on **OK** and you're off: The DIGSI 4 Gerätebearbeitung is started and the data of Device 1 is loaded.
- In the List View of the Device Editor, double-click **Settings** and then **Masking I/O** - and voilà, you can see the device matrix. Somewhat reserved though, since we have opened the device for the first time. If the monotonous grey that is slightly broken up by a grid doesn't bother you (or if you really like it), leave it as it is. For the rest of you who want to have it all, you can satisfy your personal needs with **View** → **Expand** → **All**. In any case, you should cast a quick glance at the toolbar to make sure that the settings **Indications and commands only** and **No filter** have been selected in the selection list. If not, then please make the changes accordingly!
- We now have to define a series of indications. You could, in principle, store these in any of the existing groups - which would however make little sense with regard to the clarity and above all the relocatability. So, it's best to generate a new information group in which you can store all of the indications needed for the communication test. To do so, right-click on the button of the group in front of which or after which you would like to place the new group. From the shortcut menu, select the appropriate command **Insert group – in front** or **Insert group – after**.
- In a classical two-liner dialog, you now baptize your group with a modern double name. The short text is displayed directly on the group button, the long text in a tooltip, as soon as the mouse pointer lingers on the button. You have free choice in the names of the group, just like all other names in our example, when you take into consideration the limitations through the maximum number of characters. Our suggestion for the names of the info-group are **CommTest** and **Communications test**.

- Our information group is so far still a group without information. We want to change that now and make use of the Information Catalog. In the menubar, click on **Insert** → **Information**. With this action you display the Catalog, which, when you give it a second glance you will realize is built up similar to the Device Catalog of the DIGSI Manager.
- For the required indications, you drag the respectively appropriate indications type from the Information Catalog to the group button and give the information that is generated through this a name. The indications **LED ON** and **LED OFF** are both of the type **Internal Single Point**; the message **TestSignal** on the other hand is of the type **Single Point**. Let's start with **LED ON**. In the Catalog, open the folder **Annunciations** and then the folder **Tagging**. Click on the indication type **Internal ON/OFF (IntSP)** and drag this to the button **CommTest**. As soon as you let go of the mouse key, a new indication is inserted in the group.
- Now double-click on the display text **Int SP OO** that is allocated as standard and change it to **LED ON**. As a long text you could type in **Indication is pending**, for example. In the same manner, you can now generate the two other required indications. For the indication **TestSignal** please select the type **Single Point Open/Close (SP)** that you will find in the folder **Single Point**. Our suggestion for the long text: **Test signal to devices 1 and 2**.
- All necessary indications are now generated and named. Now we have to route these to the correct sources and destinations. Please call to mind once more our overview about the necessary indications and routings. The two indications **LED ON** and **LED OFF** are sent with a press of the F1 or F2 keys. For that reason, we route the first of these indications to F1 as source, the second to F2 as source. Both indications have a common destination, namely our still to be configured CFC function in which they have to compete against one another. So please route both indications to CFC as destination.
- The result of the CFC function then decides about the state of the indication **TestSignal**, that is, whether or not this indication is pending. As source for this indication you must therefore also choose CFC.
- This indication has two destinations in all: On the one hand, the LED 1 on the device is to light up as long as the indication is pending. Therefore choose LED 1 as the first destination, for the routing type please choose **unlatched**. On the other hand, and this is the purpose of our whole exercise, the indication must be passed on to Devices 2 and 3 via the system interface. So place a checkmark in the column **Destination System Interface**.
- No sooner have you done this, then DIGSI 4 automatically opens a Properties dialog. In it you will be prompted to assign the routed indication to a corresponding IEC 61850 data object. We have already enlightened you on the purpose of this step. Those of you coming in halfway through who are merely occupied with the task should, as required, go back a few paragraphs and read up on this. The dialog already suggests an object in the form of a complete path, you only have to define the prefix.

In our example we will use **CTEST**. You should also use this prefix for the other devices. An identical prefix is not absolutely necessary but it does provide far more clarity.

- And now one of the most important steps altogether: Save your inputs!

With that, you have generated and routed all indications for Device 1. Indications and routings in the relevant area of the matrix should now look as follows.

	Information				Source		Destination																							
	Number	Display text	L	Type	Bl	F	S	C	BO	LEDs														B	S	X	C	D	CM	
										1	2	3	4	5	6	7	8	9	10	11	12	13	14							
Device, General							*	*										*							*		*			
EN100-Modul 1	009.0100.01	Failure Modul		IntSP																										
	009.0101.01	Fail Ch1		IntSP																										
	009.0102.01	Fail Ch2		IntSP																										
CommTest		LED ON		IntSP		1																								
		LED OFF		IntSP		2																								
		TestSignal		SP				X		U															X					

Intermediate state: Indications and their routings for Device 1

We will now close this device and take care of the indications for the other two devices. This may seem a bit unergonomical to you since we still have to configure a CFC function for Device 1. And you're right. In a normal case, you would first of all completely parameterize Device 1 before you would begin with Device 2. We feel it becomes much more clear, however, when we first do all routings and then go back once more to Device 1.

### Down to Work (2nd. Shift)

If your children (or you and your children or even only you) like to watch Teletubbies then the instruction at the end of all instructions will be familiar to you. For then it is **Once More!** For ultimately, you insert one indication each of the same type into both Devices 2 and 3 and also route it to the same sources and destinations. For that reason we will only show you the procedure for Device 2 and then say **Once More!**

- Let's briefly look at what we are already familiar with and know. Open Device 2 with the option **Offline** and double-click on **Masking I/O** in the DIGSI 4 Gerätebearbeitung in order to display the matrix. Optically set up the matrix so that it appeals to you.
- Even if we only require one single indication, it is surely a benefit if we place it in its own information group. Where you insert this group between the already existing groups is not so important. Not absolutely necessary, but all the more meaningful is to also identify this group as **CommTest**.
- Now open the Information Catalog and grab a message of the type **External Single Point ExSP (Open/Close)** that you drag onto the group button. Suggestion for display text / long text: **TestSignal / Indication from Device 1**.

- The input for the indication comes from outside via the system interface. For that reason, please put your first 'x' in the column for **Source System Interface**. You are already familiar with the dialog that is now displayed. Simply complete the prefix and its best to use the same one as you used for Device 1. If you decided to use our suggestion, it was **CTEST**. A click on **OK** and on we go.
- For the second and therefore last routing we ask you to enter a U in the column **Destination LED 1**. For ultimately, LED 1 is to light up as soon as and only for as long as the indication is pending.
- Save your entries, close the device and then: Once More!

**PS 1**

Here a visual impression of indications and routings in the Devices 2 and 3.

Device, General	Information				Source			Destination																											
	Number	Display text	L	Type	BI	F	S	C	BD	LEDs														B	S	X	C	D	CM						
										1	2	3	4	5	6	7	8	9	10	11	12	13	14												
						x	x																												
EN100-Modul 1	009.0100.01	Failure Modul		IntSP																															
	009.0101.01	Fail Ch1		IntSP																															
	009.0102.01	Fail Ch2		IntSP																															
CommTest		TestSignal		ExSP		X			U																										

Identical twins: Same indications and routings for Device 2 and Device 3

**PS 2**

If you are interested in how and when you route IEC 61850-conforming indications on the system interface without being confronted by instance numbers and prefixes, then please read Chapter 12.

## Creating a CFC Function

So that a continuous ON or OFF state is supplied when you press the F1 or F2 keys, you now have to create a simple CFC function for Device 1.

### Work ...

Once again, we offer you our advice to learn how to

- ❑ create and open a CFC chart,
- ❑ drag the necessary function block from the Catalog to the chart and to link this with the input and output information and
- ❑ trigger the compilation of the chart.

If you really don't want to know any more, then it's off to Page 37.

### ... and Pleasure

For those of you who are hungry for knowledge, we will first briefly explain

- ❑ the function that we will configure and
- ❑ nothing else (sorry, but the function is really so simple)!

## Basic Questions

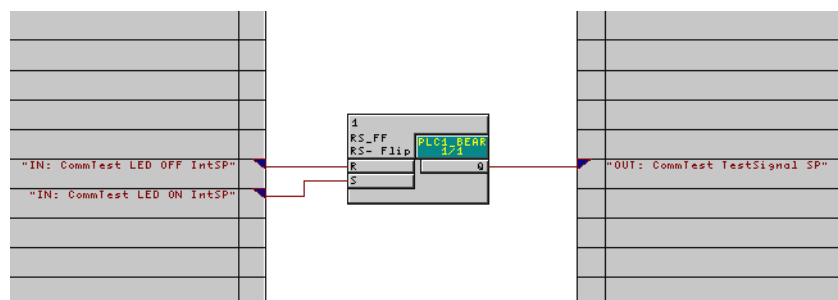
How you basically work with the CFC Editor is explained in detail in the **DIGSI 4 Start Up** manual. For that reason, we will concentrate, after an amusing and brief crash course, immediately on the working instructions. (if this crash course from the above-mentioned manual should sound familiar to you then here is a tip to all freelance authors: copy and paste is a clever way of being paid twice for one and the same piece of work!)

## Crash Course

The fundamental procedure for creating logic functions is simply described: First of all, you route all information that you require as input value or as results for your logic function in the device matrix to CFC as destination or as source. This we have already done in Chapter 7. Next, insert a new CFC chart in the DIGSI 4 Gerätebearbeitung and open it with the **DIGSI 4 CFC**. You link the individual information with function blocks that are summarized in diverse types in a catalog. Using drag & drop, you place these in the CFC plan, parameterize them and then interconnect them with the input and output information. How the blocks function internally is not really important. Finally, you select the correct run sequence for the chart. Last but not least, the chart is compiled into a language that is comprehensible for the SIPROTEC 4 device and stored along with the parameter set.

## All the Rage: Flip-Flops

We implement the given requirement with a block of the type RS-Flip-Flop. We'll interconnect the SET input of the Flip-Flop with the information **LED ON**. On the other hand, we'll connect the RESET input with the information **LED OFF**. A connection between the output of the Flip-Flop and the indication **TestSignal** makes the function perfect. A press of the F1 key causes the indication to be pending, a press of the F2 key resets it. And since actions speak louder than words, we'll show you now what you have to do! The completed CFC function should look like the one in the picture below.



Taken care of in a jiffy: The necessary logic function in the CFC Editor



## Down to Work

During this part, you will insert a function block into a new chart, parameterize it and link it with the described indications. After that, you will start the compilation.

- In the List View of the DIGSI 4 Gerätebearbeitung, select the entry **CFC**. With **Insert** → **CFC chart** from the toolbar, generate a new, still empty CFC chart. Change the name of the chart to **Test signal ON OFF**. With CFC charts, you have to open the **Object Properties** dialog box via the shortcut menu. In it, you can then make the name change.
- With a double-click on the name of the chart you open it and then you will have a pearly-white expanse in front of you. On the right side of the **DIGSI 4 CFC**, the Catalog with the function blocks should be visible. If it is not, click on **View** → **Catalog** in the toolbar. In the catalog, double-click on **Memory**, select the block **RS-FF**, and while keeping the mouse button pressed down, drag the block onto the page. The positioning of the block is ultimately a question of aesthetics but not of the function.
- The SET input of the function block must now be interconnected with the indication **LED ON**. Therefore, right-click on the connection **S**. In the shortcut menu click on **Interconnection to Address** and that way open the dialog box **Select left border**. Select the indication **LED ON** and click **OK**. In the left margin, the entry for the interconnected information appears. The interconnection itself is visualized by a connection line between the information and the input of the function block. Then, in the same manner, interconnect the input **R** (reset) with the indication **LED OFF**.
- What we need now is a unique result of our logic function. For this, we have to interconnect the output of the function block with an indication. Right-click on the output **Q** of the function block. In the shortcut menu, click on **Interconnection to Address**. The associated dialog box now has the title **Select right border** as a consequence. Here, select the indication **TestSignal** and click **OK**. The desired interconnection is displayed.
- The logic function is now complete. However, you must check whether the matching run sequence is set for the function. You will find more detailed information on the topic of run sequences in the DIGSI 4 Start-Up manual. Just this much here: Logic functions are executed on different run sequences in order to, among other things, optimally utilize the processor power of a SIPROTEC device. The name of the currently set sequence is displayed at the bottom right in the statusbar. If you see **PLC1\_Bearb** there, then everything is OK. Otherwise, you have to assign the function to exactly this run sequence. To do so, use **Edit** → **Run Sequence** to switch to the Runtime Editor.
- In the tree view, select the name of the run sequence that is currently set. That way, a folder icon with the name **TestSignal ON OFF**, in other words the name of our function, is displayed in the list view. Using drag & drop, place this folder on **PLC1\_Bearb** in the tree view. And like magic, the function finds itself on the correct run sequence.

- 
- So far, the function only exists in a form that the SIPROTEC 4 device cannot understand. You still have to convert this into a language that is suitable for the device. Strictly speaking, you only have to give the command for this, the **DIGSI 4 CFC** does the rest. In the menubar, click on **Chart → Compile → Charts as Program**. The **DIGSI 4 CFC** now generates an executable code from the existing chart. Together with the parameter set, it is loaded into the SIPROTEC 4 device.
  - As soon as the compilation is completed, close the **DIGSI 4 CFC** and save your work in the DIGSI 4 Gerätebearbeitung. Then, close the device and device editor and return to the DIGSI Manager.

# Routing Infos between Communicators

# 9

In Chapter 7, we showed you which indications and routings are necessary for each of the participating devices. With that alone, we cannot achieve a functioning communication, for so far, the individual information is more or less standing around helplessly. For that reason, we have to link the information between the communicators and we'll do so with a little help from our friend, the DIGSI System Configurator.

## Work ...

Off to the finish. In this chapter we will show you

- ❑ how you open the DIGSI System Configurator,
- ❑ how you insert a GOOSE application and
- ❑ how you create all the necessary links for this.

If you're only interested in the rudimentary task at hand, you can go directly to Page 46.

## ... and Pleasure

If your interest goes beyond pure friendship, the following information on the DIGSI System Configurator is free-of-charge:

- ❑ Under which start conditions does it start its work?
- ❑ Which additional information does it keep in readiness?
- ❑ How can you force its user interface?

---

## A Double-click Opens Better

In Chapter 6 we entered the individual devices that are to communicate with one another as communicators in an IEC 61850 station. In keeping with the operating philosophy according to which we open an object (device, branch control picture, CFC chart) in order to edit the associated data, here as well, we will open the station – and thereby start the DIGSI System Configurator. Once again, the optimal integration of all necessary tools becomes obvious. A double-click on the station icon is enough to start the DIGSI System Configurator.

## Starting Conditions

But wait! Cut! Pause! We wanted to pass on a few hints. So that you can even start the DIGSI System Configurator, you must have previously opened each SIPROTEC 4 device that is entered as a communicator at least once. Why? Quite simply: The DIGSI System Configurator gets its information about the individual communicators from the respective device descriptions (ICD files) which we already introduced in Chapter 3. That's why it is important that an ICD file exists for every communicator when you open an IEC 61850 station. For other IEC 61850 communicators, that is, not SIPROTEC 4 devices, these ICD files are imported into the project. With SIPROTEC 4 devices they are generated as soon as you open the device to edit it. It is also important that the ICD file is current. If, for example, you change the IED name in the DIGSI Manager, you have to update the ICD file by once more opening up the device with the DIGSI 4 Gerätebearbeitung. You also have to make sure that none of the participating devices is being edited when you go to open the station. The DIGSI Manager will mercilessly stop you. And now let's go on!

## Personal Views

You'll have to have a little bit of patience before you can come face to face with the DIGSI System Configurator, for, from all the many data in XML format, it must first create a clear and informative view. The more communicators a station has, the more links there are between the communicators, and the longer it takes, unfortunately. But you will be rewarded with a really beautiful view. The DIGSI System Configurator comes along in a stylish, transparent outfit and gives us a positive foretaste of what the complete DIGSI 4 will look like in the near future.

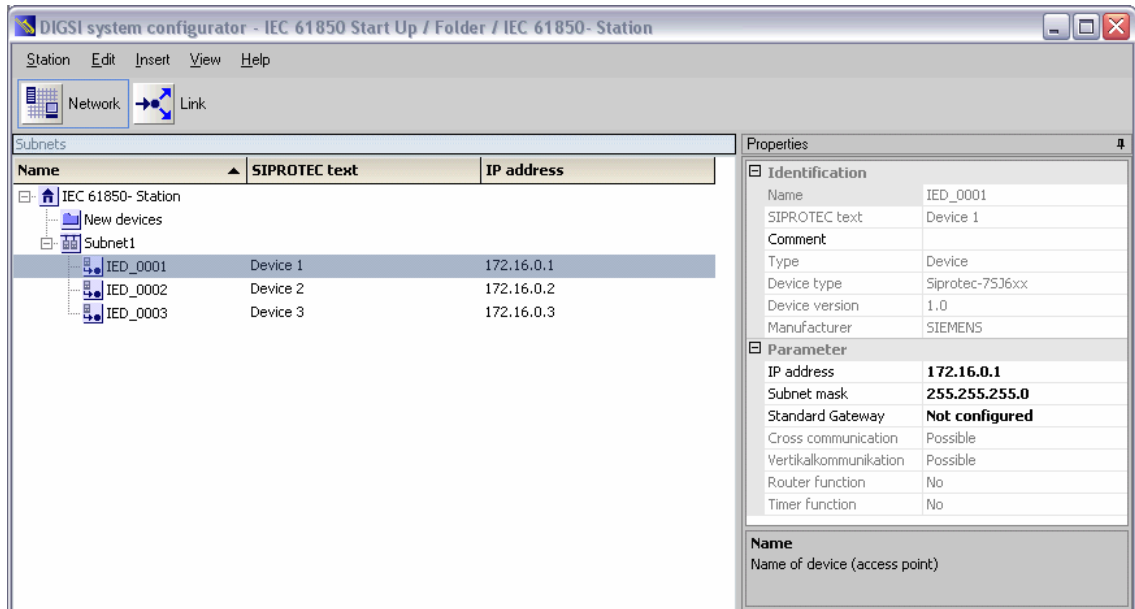
In the area of the menubar, you will immediately notice two amply dimensioned buttons. A mouse-click on either of these and you quickly switch between the work areas **Network** and **Link**, as already described in Chapter 4.

Right now, we want to take a closer look at these two areas and explain how you go about using them.

## A1 Network

The **Network** area shows you the current network structure on the left side. In our case, this consists of only one subnet. If you open this with a mouse-click, for example, you will get all important, basic information at a glance. In the **Name** column, you see the IED names of the three devices that you or the DIGSI Manager assigned in the DIGSI project. To the right of this you find the names of the devices as they are displayed in the project window of the DIGSI Manager.

If you're interested in details, the Properties window on the right side helps you further. With the mouse, you select a particular element within the network structure and the Properties window immediately supplies you with the appropriate information.



Subnets, devices, IP addresses: The **Network** area supplies basic information at a glance!

Why don't you select a communicator in the **Subnets** window - **IED\_0001**, for example. The Properties window then shows you, among other things, the device type or even the manufacturer in the **Identification** section. The manufacturer could be interesting because, according to IEC 61850, devices from different manufacturers can take part in the communication.

### Cleanliness is next to Godliness

A little lower you will see a series of parameters and their values. Very important is, of course, the IP address, for it establishes order in the network. Invalid or doubly-assigned IP addresses are the beginning of the end for a functioning communication in the Ethernet network. That is why the DIGSI System Configurator suggests a correct and unique IP address for every communicator when you open a station for the first time. For most applications and, of course, for our practice example, you can accept the given addresses unchanged.

Please note that only IEC 61850 communicators of a DIGSI project are displayed and taken into consideration when an address is assigned. Other communicators of the network, such as, switches, serial hubs or PCs naturally also require valid IP addresses. These, however, are not managed in the system configurator.

---

If, however, you would like to change the address because you are expanding the network structure with new devices or with new subnets in which you are going to move the existing communicators back and forth, the DIGSI System Configurator doesn't leave you hanging. The magic function is called **Automatic addressing**. You merely assign an initial IP address for the subnet. To do so, select the subnet in the Network view. In the Properties window, enter the desired value in the field **IP start address** and then press the button - the one titled **auto IP** of course. The DIGSI System Configurator does the rest and assigns every communicator in the selected subnet a correct IP address.

### Urge to Expand

When you open a station for the first time, its network structure consists solely of one subnet. If you want to expand the existing network by additional subnets, it's no problem. First select the icon for the station. Then right-click and select **Add subnet** from the shortcut menu. You then distribute the individual communicators amongst the different subnets using drag & drop. Finally, you check the IP addresses and change them if it's necessary as described above.

But how do new communicators get into the game? For this, you first of all have to close the DIGSI System Configurator. In the DIGSI Manager you insert new devices into the project and assign them to the station. After you open the station with the DIGSI System Configurator, you'll always find these communicators in the folder **New devices**. Using drag & drop, you shift these into the respective subnets. Of course, you can also use the standard commands **Cut** and **Paste**.

### Accessible

Now everything in life could be soooo easy, and then reality comes crashing in (= IEC 61850) and teaches us the opposite. For strictly speaking, you don't really see any communicators or devices in the network structure. Instead, you see the so-called access points to these. According to IEC 61850, a device can have several access points with different IP addresses in each case and, in that way, for example, be integrated in various subnets simultaneously.

If a device has more than one access point and two or more of these access points are in the same subnet, the suffix **Px** is displayed, whereby **x** can be a number between 1 and infinity. For example: **IEC\_0005/p3**.

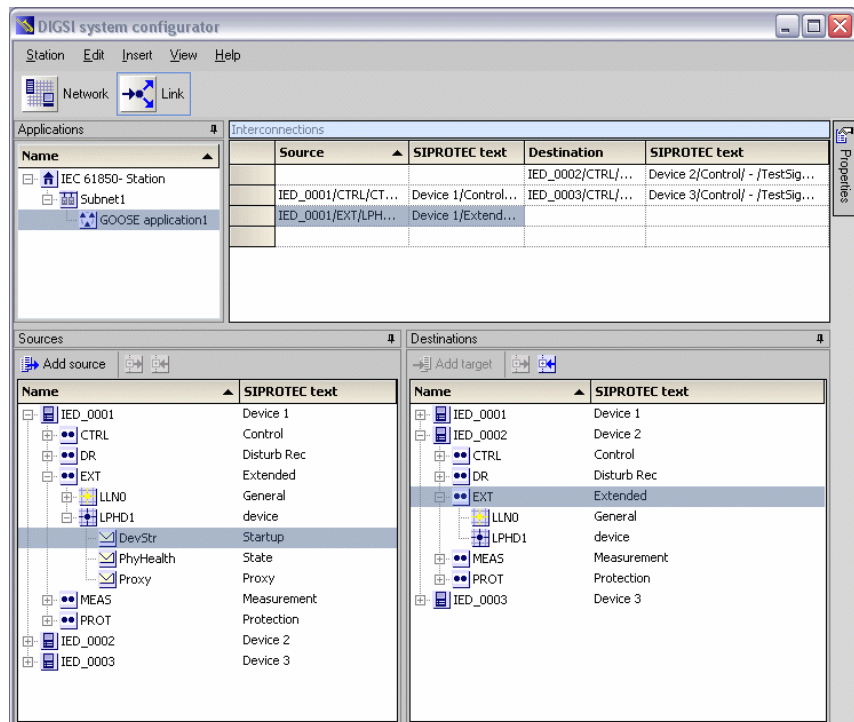
The SIPROTEC 4 devices used in our example each only have one access point. We will therefore continue to speak of devices or even communicators instead of access points. Please bear in mind, however, that this is a simplification.

## Matchmaker

In the **Link** area you define which information is to be exchanged between the individual communicators. As you have already learned in the previous chapters, the data objects of different communicators have to be linked with one another for this.

In the traditional sense, you wire a contact as source with a binary input as destination. However, in our case, we are dealing with a logical wiring that is done via the Ethernet bus.

To now create a link, select a data object of Communicator A as source and a matching data object of Communicator B as destination. The result of the interconnection is displayed in the Interconnections Table. Source and destination objects are clearly arranged in separate windows in the form of hierarchical tree structures. The sequence in which you interconnect sources and destinations with one another is open and is completely up to you. You could, for example, first select all source information and then assign destination information to them. Vice-versa is also possible, or, you could select any mix of sequence. We will limit ourselves in the instructions, however, to describing how you first select a source object and then assign one or more destination objects to it.



In the **Link** area, you define sources and destinations for an application

## From Case to Case

Interconnections are always compiled as application-related. Basically, there are two categories of applications: Report Applications and GOOSE Applications. You'll find out in Chapter 11 what report applications are all about. At this point, we are mainly interested in the GOOSE application. GOOSE is the abbreviation for Generic Object Oriented Sub-

station Event. With this type of application, information on events can be transmitted quickly between different communicators. And that is what we need for our example.

A GOOSE application is always a subordinate of a certain subnet. This is because GOOSE communication, according to the standard, must technically always be limited to one subnet. In other words, it cannot be active beyond the limits of the subnet. You can, of course, generate several GOOSE applications within a subnet. After all, these applications serve the structuring of interconnections according to your application purpose. Examples for this are certain interlocks, measured value tasks, or, as in our case, a communications test.

Theoretically, you can insert as many GOOSE applications as you like. The only restriction you will have is when you run out of data objects. For every data object can only always be assigned to one interconnection. You may be able to link one source object with several destination objects, but, a destination object can only always be assigned to exactly one source object. At any rate, you are permitted to use one and the same source object in different applications.

## Self Determination

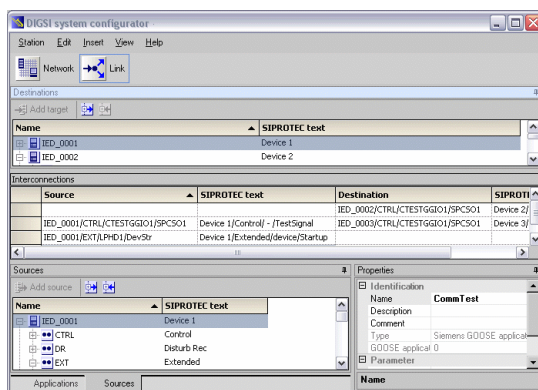
Before we go on to the work instructions, a few words about the operator interface of the DIGSI System Configurator. You can adjust this interface to meet your needs. In this way you can vary the size and the position of the individual windows, can combine various windows into one single one, can switch as required from the static to the dynamic view and can even continue working with the familiar arrangement after a complete restart, for, your personal interface layout is stored while the DIGSI System Configurator is closed.

Admittedly, according to the rules of editing, the last sentence is far too long. But it simply had to be said. And to make up for it, the whole thing will now be repeated so you can take notes.

## Sliding Windows

Each of the two work areas is divided into several windows that, with regard to their scaling, are reciprocal actions to one another. For every

work area there is a main window which is always visible. In the **Network** area it is the **Subnets** window; in the **Link** area, on the other hand, it is the **Interconnections** window.



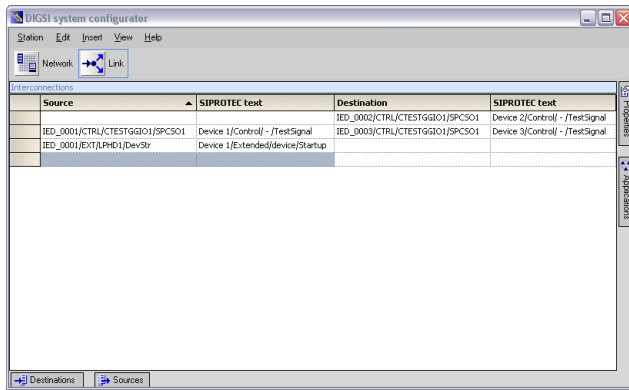
Nothing is impossible: Windows can be shifted and can be displayed as dynamic or static.

The rest of the windows can, independent of one another, be shown either continuously or they can be dynamically hidden or shown. You can recognize these windows by the symbol of a stick pin in the right-hand corner of the title bar. To switch between the two display options, you simply click on this stick pin symbol.



If the stick pin is pointing down, the static display is set for this window. If, however, the stick pin is pointing to the left, the window is dynamically hidden or shown.

For every window that is set up as dynamic, a tab is displayed on the left, lower, or right hand edge of the work area. As soon as you move the mouse pointer over one of these tabs, the associated window appears.



Adaptable: A double-click on the column boundaries in the tables optimizes the respective column width.

When you click in the window it gets the focus. After that, the window remains visible until another window is given the focus. If, however, the mouse pointer leaves the window without the window having been given the focus, the window automatically disappears.

Of course, you can also change the arrangement of the windows to one another. Because of the interdependency of the windows, by shifting one window you also change the position of other windows. As well, you can also position windows without overlapping one on top of the other. Tabs then automatically appear on the edge of the window. You can then activate one of these windows by clicking on the associated tab.

In the title bar of Window A, which is to be shifted, simply position the mouse pointer. With the left mouse button pressed down, drag the mouse pointer to the edge of Window B, onto which Window A is to dock. If the two windows are to be positioned without overlapping, drag the mouse pointer onto the title bar of Window B.

## Law and Order

If the **Network** area is structurally still quite simple, it is quite possible that you could easily lose the general view in the **Link** area after wildly shifting things around. Our developers also seem to have come to this conclusion and they have preprogrammed an out for you through a menu command. If you would like to return to the standard view, then simply select **View** → **Restore Standard** from the menubar. Without asking you for further permission, the system then puts all the windows back to the system's default sizes and positions. In addition, the continuous display is activated for all windows.

---

## Down to Work

If you have been diligently keeping pace with us up until now and you can already see the DIGSI System Configurator open before you, then you can skip the next two steps without giving it a thought.

- Make sure that none of the three devices is open for editing. Then start the DIGSI System Configurator by double-clicking on the icon of the IEC 61850 station. Alternatively, you can also select the command **Open Object** from the shortcut menu of the IEC 61850 station.
- In the **Network** area of the DIGSI System Configurator leave all settings as they are. The same is true for the IP addresses.
- Now switch to the **Link** area. The easiest way is to click on the large button in the upper area of the DIGSI System Configurator. If you haven't already done so, then scale the DIGSI System Configurator to the complete window size.
- In the **Applications** window open one after the other the station and the subnet and you will then see the entry **GOOSE-Application1**. This is an application without interconnections that is generated by the DIGSI System Configurator as a standard. Select this entry and in the Properties window enter **Communication test** as the name. This name will then automatically be adopted in the tree structure in the **Applications** window. You won't be needing the **Applications** and **Properties** windows for the time being. Therefore make yourself a little room for the interconnections. Simply click on the stick pins in these windows to do so.
- For the application that you just named, you will now choose source and destination objects. For these objects it is a matter of that information which you had defined in the configuration matrix for the individual communicators and that you had routed on the system interface. Concretely, the indication **Test signal to Devices 2 and 3** is to be sent from Device 1 (the source) to both Devices 2 and 3 (the destinations). If you remember: After routing to a system interface, this indication was assigned to a data object with the following complete path: **CTRL / CTESTGGIO1 / SPCSO1**
- The first thing you have to do is select the data object **SPCSO1** for the participant **Device 1** in the **Sources** window. To do so, open the level **IED\_0001**. This level corresponds to Device 1. The identifiers of all logical devices will be displayed as further levels. The required data object is assigned to the logical device **CTRL** (Control). Therefore also open this level.
- Now you will see the names of all logical nodes included in this logic device. The logical node relevant for our example is easily identified by the prefix **CTEST**. Its complete name is **CTESTGGIO1**. Open this level. It contains the data object **SPCSO1** which corresponds to the indication **Test signal to Devices 2 and 3**. To help you in the selection, the display text **TestSignal** is shown in the column SIPROTEC text which you had assigned for this indication in the configuration matrix.

- 
- Select the data object's identifier and click the **Add source** button in the toolbar of the **Sources** window. The object, including the complete path, is entered in the column **Source** of the **Interconnections** window.
  - As the indication **Test signal to Devices 2 and 3** already implies by its identifier, it is to be transferred to the two communicators **Device 2** and **Device 3**. Our link will therefore have two destinations. You select the two destination objects, in this case **SPCSO1** of both communicators **IED\_0002** and **IED\_0003**, the same as you did above and then accept them in the Interconnections Table.
  - For Device 2 this means, open the levels **IED\_0002**, **CTRL** and **CTESTGGIO1** one after the other and thus bring the destination object **SPCSO1** to the forefront. In addition to the method already shown for placing an object in the Interconnections Table, you can also do the same thing by double-clicking on the object's identifier. Because of its monogamous habits, the object is indicated by a different symbol in the destinations catalog as soon as it is entered in the list. This means that this object is no longer available for any other relationship.
  - For Device 3, first open the three levels as already described so that you can once again get to the object **SPCSO1**. To place it in the Interconnections Table we will now try method Number 3: Drag & Drop. Simply select the object, drag it to the free cell beneath the already existing destination object and let go of the mouse button. Voilà! This object as well withdraws from any further interconnections for the time being.
  - Now almost everything has been said and done, at least, what is necessary for our task. For that reason, you can now close the DIGSI System Configurator. Your entries will automatically be checked for plausibility and then stored. Since you have made changes, a notice will pop up that the parameter sets of the individual station communicators have to be updated. The parameter set update is done collectively for all devices. To do so, in the DIGSI Manager, open the Properties dialog of the IEC 61850 station and select the **Update** tab. Simply click on **Update all parameter sets** and let things take their course.



## Up to the Devices

# 10

Everything that you have done up until now is located somewhere in the caverns of your PC's hard drive in the form of bits and bytes. And even now, none of the participating devices knows what's about to happen to it. We're about to change that.

### Limited Mandatory Event

In so far as you haven't viewed the work up until now as purely a dry run, but are quite interested in the test phase, then this chapter is mandatory for you. This is true even though we are no longer explicitly placing the individual steps on a grey background.

### Checkpoint

We are going to assume that you have fulfilled all of the hardware requirements that were described in Chapter 4:

- SIPROTEC 4 device with firmware version as of 4.6
- An interface of the type EN100 built into the device

As well you have a switch and three connector cables (switch <> device) at the ready as well as the serial cable that was supplied along with DIGSI 4.

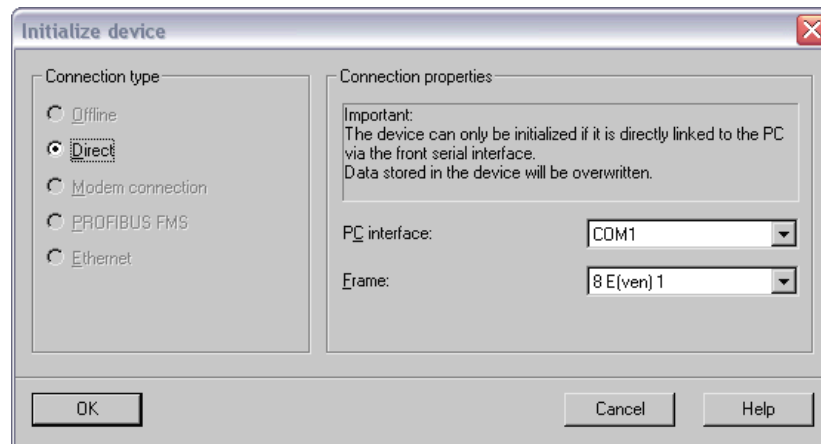
### The First Time

The data that we have prepared and parameterized is now to be transferred one after the other into the respective devices. Since we had created the devices new in the project, we first have to initialize them. Initialization gives your SIPROTEC 4 device its own identity. Not only that the complete data set is transferred into the device in the process, during the initialization, the device is uniquely addressed.

We have extensively described the initialization of a device in the DIGSI 4 Start Up manual. For all of you who are not familiar with this, here's one simple rule to remember: As soon as you want to build up a connection between a newly created *virtual* device in the project structure and a *real* device, you must *initialize* the *real* device once with the data of the *virtual* device. After that, you can build up and disconnect a connection between the PC and the device as often as you like.

## We have Ignition

Now connect Device 1 to the PC. To do so, take the DIGSI 4 cable and plug it into a free serial PC interface (alias COM Port). Plug the other end into the front interface on the SIPROTEC 4 device. In the project, select the icon for Device 1 and then choose the command **Device** → **Initialize** from the menubar.



With the **Initialize device** dialog box you can check once more the current communication settings

The **Initialize device** dialog box displays the current settings for the PC interface (set according to the COM port chosen by you) and the Frame (please leave at **8E(ven)1**). Close this dialog box by clicking **OK**. A message will pop up that existing data in the SIPROTEC 4 device will be deleted by the initialization. If these values are not relevant for you or if they were already saved in files, then click on **Yes**.

## The Right Words

After several intermediary messages you will be prompted to enter a password. As long as you haven't made any changes to the passwords, type in zero six times. Then click **OK**.

So as not to get too much of a coffee high, we recommend that you set the fastest baud rate in the device.

Now you can lean back and enjoy a good cup of coffee without feeling guilty because the amount of data to be transferred is large and the transmission will take some time. After all, the bits have to squeeze themselves through the serial cable one after the other. You only have to put up with this for the initialization, though. Once you have established communication via the Ethernet, you can of course use this way to transfer modified parameters to the devices. For then, after all, everything will go much faster.

## Multivisual

By the way, you cannot only follow the transmission procedure on the monitor, you can also follow it on the SIPROTEC 4 device's display. Once the initialization is completed, the default display will again appear. The connection between the PC and the device was only of a temporary nature and once the initialization is completed, it is disconnected.

---

## Helpful Connections

Ethernet patch cables provide the necessary connection between switch and devices, assuming of course that you have decided on the electrical version. These come in various colors and lengths depending on your preferences. What is important in your selection is that you use 1:1 wired cable, no cross-over cable! Otherwise you will have little or no joy in our communication test. If you decided on light as the transportation vehicle, then of course you will grab fiber-optic cables. As a rule, you would establish an electrical connection between the switch and the PC.

## Live-wires

Now for the moment of truth. On Device 1, press the F1 key. In addition to its own LED 1, the two first LEDs on Devices 2 and 3 should also light up. With a press of the F2 key, they should switch off again.

## No Sign of Life

It doesn't work? Then here four tips that may help you further.

- Tip 1: Check the routings and other settings  
Compare once more your settings with our specifications. Maybe an error has crept in or you have forgotten a routing or a link.
- Tip 2: Test by simplifying the setup.  
For the time being, only connect Device 1 with one of the two other devices, directly via a switch. What is important is that you use a twisted patch cable. If this works, then the error lies either in the original cable or in the switch.
- Tip 3: Configure an additional CFC function  
A simple CFC function in Devices 2 and 3 informs you about the status of the indication **TestSignal**. To do so, in both of these devices route this indication to CFC as the destination. In the **DIGSI 4 CFC** grab a block of the type **SI\_GET\_STATUS**, and interconnect its input with the indication **TestSignal**. The block's output has the status **High**, as long as the status of the indication **TestSignal** is not current. In other words, as long as the indication doesn't arrive at the device. You can now interconnect the output with a user-defined indication (type **Internal Single Point**) which you previously routed to CFC as source. In addition, route this indication to LED 2 as destination. The result of this whole thing is that LED 2 remains lit until the indication **TestSignal** arrives at the device. So if you press the F1 key on Device 1 and the LED 2 on the two other devices continues to be lit, then the communication line has a fault in it.
- Tip 4: Harass our hotline!  
Don't be shy, call. You'll find the telephone number at the beginning of this book.

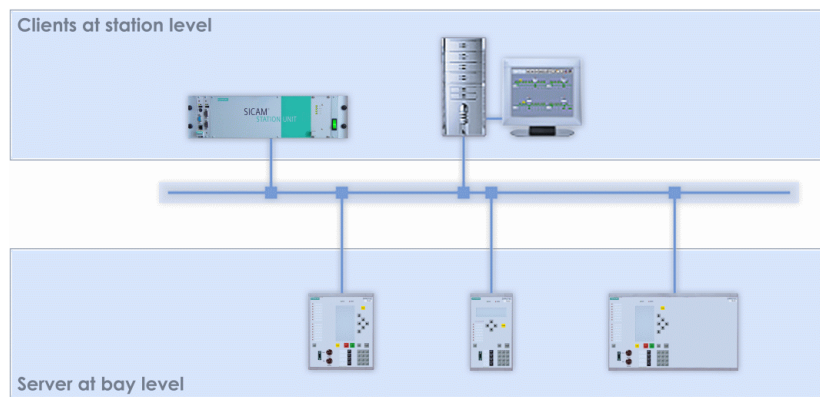




# Unveiled: The Control Center Report

# 11

Household report, health report, innovation report, we all know about these reports that quite often have a lasting influence on our lives. But what about the Control Center Report? We feel it has become better one because a clear language is spoken, namely IEC 61850. For example, from a one-sided Master-Slave liaison has evolved an enlightened Client-Server relationship. (And that is only one of the many benefits.)



Server and Clients: Everyone gets it all

## This Way and That

All devices at the bay level record data and supply it. They are the servers in our relationship. The automation system at the station level wants the data, gets it and processes it. This is the client, and unlike Highlander there can be several of them. But there's more: A client can demand data from all servers, vice-versa, a server can provide data to all clients. And: many tasks are taken care of directly between the servers. They communicate through GOOSE, without needing the client.

**On Recommendation**

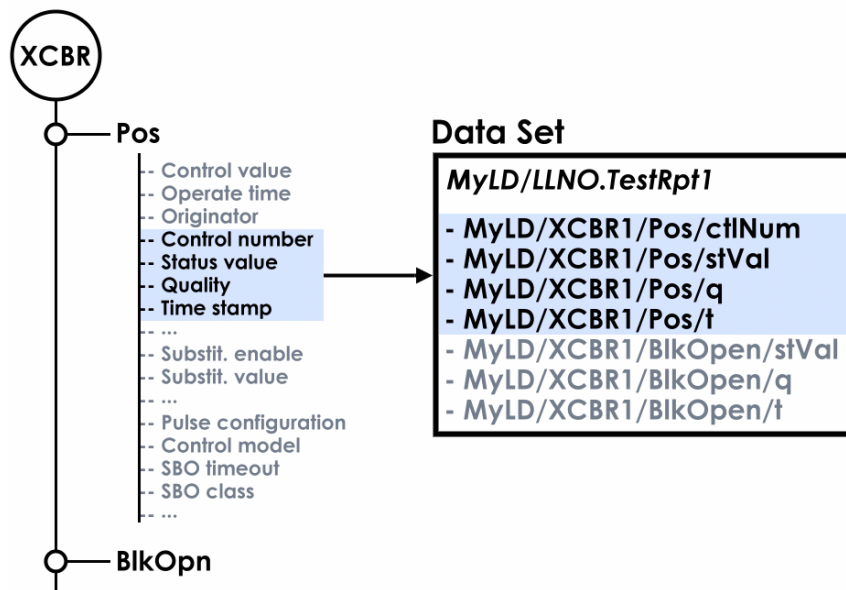
For an optimally coordinated solution, we of course recommend SICAM PAS as the station automation to supplement your SIPROTEC 4 devices. For that reason, in addition to the basic procedure for data exchange, we will especially explain how this works in connection with SICAM PAS. To be sure, we won't leave you standing out in the cold should you, contrary to expectation, have decided against SICAM PAS. You will read about Data Sets, Unbuffered Reports, Report Control Blocks and much, much more. So, everyone put everything away and pay attention.

**Beam It Up, Server**

The transmission of data from the server (such as, a SIPROTEC 4 device) to the client (such as, a SICAM PAS) is called Reporting. In reporting, the data is not just individually and randomly sent through time and space. Where would we end up? But which data is transferred how, when and why? After some philosophical observation we discover that this depends on the following two influences:

1. the compilation of so-called Data Sets.
2. certain conditions that are defined in the so-called Report Control Block (RCB).

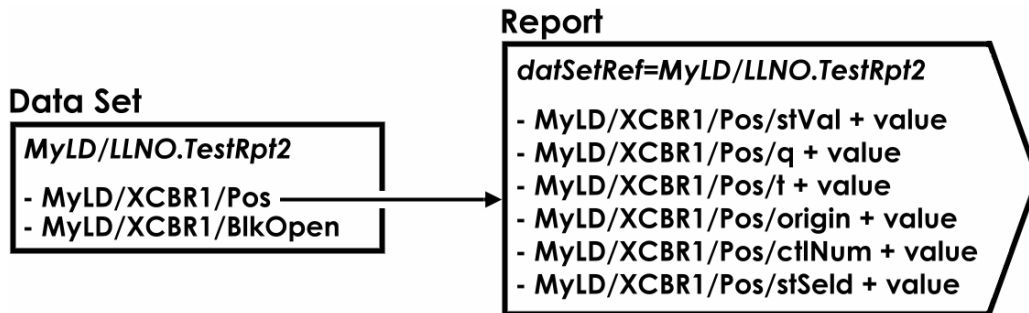
Data are always transferred in groups. The standard calls these groupings Data Sets. We have an example to explain.



The Data Set groups required data objects and attributes

To the left of the picture, you can see the node **XCBR** (circuit-breaker) with one of its data objects, namely **Pos**. Some of the existing data attributes for this data object are represented. To the right of this we have formed a data set. As a rule, the data set also contains additional data objects of the node, in our example **BlkOpen**.

In the data set, as shown in the figure, data attributes can be explicitly designated to one data object. In this case, precisely the name of this attribute and its value would be transmitted. SICAM PAS, on the other hand, takes the opportunity to only include the data object in the data set without naming individual data attributes. The names and current values of all attributes that belong to the object are transmitted even if the transmission condition was fulfilled for only one attribute. For you it means an information windfall with limited configuration complexity at the same time. A small graphic should make it clearer.



All for one, and one for all: If the value of an attribute changes, then the values of all attributes are transmitted

### Self-timer

As already mentioned above, different conditions play a decisive role in the transmission of data. Possible transmission conditions which the standard specifies are, for example, Data-Change or Data-Update. The **Data-Change** condition is fulfilled when the value of an attribute has actually changed. On the other hand, the **Data-Update** condition is already fulfilled when a new value merely exists, which doesn't mean to say that this is different to the previous value. This is of interest in the recording of counter values and the harmonics analysis. Other conditions are Quality-Change or Integrity.

Data Set and RCB, of course, belong in the server, for after all, it is the one that is to gather the data and pass it on to the client. But how do Data Set and RCB get there? For that, there are basically two versions.

### User-friendly ...

In the more user-friendly of the two, the client creates the data set and the RCB in the server as the initialization for the data transmission. Then the RCB is activated in the server by an explicit command from the client. Data transmission then takes place by means of the trigger conditions defined in the RCB until the RCB is again explicitly deactivated by the client. This version provides the opportunity to change the data set with the help of the client depending on the requirement. In the everyday language of the standard, this version is therefore also referred to as Dynamic Creation.

### ... or Plain

The somewhat plainer alternative is that the data set is transferred into the server with the parameter set, for example, where it remains unchanged for the time being. This is also referred to as a Configured Data Set. As a rule, you only make use of this option when the client does not support the Dynamic Creation version.

---

**Export/Import Ltd.** SICAM PAS as a client, naturally uses Dynamic Creation. After you have imported the station description into SICAM PAS UI, you compile a data set for every server in the **Routing** area. So that you don't have to create each data set individually for functionally identical servers with the same parameterization, SICAM PAS UI helps you out with suitable export and import functions.

**In the Worst Case** In so far as you really want to use a station automation XYZ (name changed by the editor) that quite possibly only supports Configured Data Sets, you can still fall back on SIPROTEC 4 devices. You then create the data set with the System Configurator. In the **Link** area, you first insert a Report application. For this you then select data objects exclusively as sources. The data set generated in this manner is transferred along with the rest of the parameters into the device.

For station automation XYZ you then, as a rule, also require the device description (ICD) and the station description (SCD). You can easily export both of these with the DIGSI Manager. To generate the device description, right-click on the device icon to open the shortcut menu and select **Export device**. In the file dialog choose **ICD** as the file type. Also choose a storage area and change the suggested name as required. A click on **OK** and you're off to the races. You export the station description in a similar manner. From the shortcut menu for the IEC 61850 station, select **Export IEC 61850 station**. SCD is your only choice for file type. Then choose a storage area, define the file name and click **OK** - that's it!

## What Else You Should Know

In this chapter, we have put together a few things that, while they don't directly refer to our example, we feel are important in the whole context of this book.

**You should know, what you have to pay attention to with direct Ethernet communication between PC and device.**

With a simple, direct Ethernet connection between PC and SIPROTEC 4 device, you need neither an IEC 61850 station nor do you have to make any IEC 61850-conforming settings. Still, you have to pay attention to some important details.

First, insert the appropriate device from the Device Catalog into a project. When you select the system interface, you will have to decide on an EN100 module in the electrical version because, as a rule, the PC doesn't have an optical interface. Once you've inserted the device, open the Properties dialog. In the **Communication parameters** tab, enter meaningful values for the IP address and the subnet mask. To do so, you must first activate the input fields by clicking on **Enter manually**.

The screenshot shows the 'Properties - SIPROTEC device' dialog box with the 'Communication parameters' tab selected. The 'General' section shows 'VD address: 1025' and 'Channel switch: [No channel switch]'. The 'System interface [Ethernet]' section contains the following fields: 'IP address: 172 . 16 . 0 . 1', 'Subnet mask: 255 . 255 . 255 . 0', 'Standard gateway: 0 . 0 . 0 . 0', 'UDP port: 8080', and 'IED name: IED\_0001'. A warning box on the right side of the dialog reads: 'Warning: This device is not allocated to a station. The address will be given by System Configurator. Parameters entered manually will not be checked!'. At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

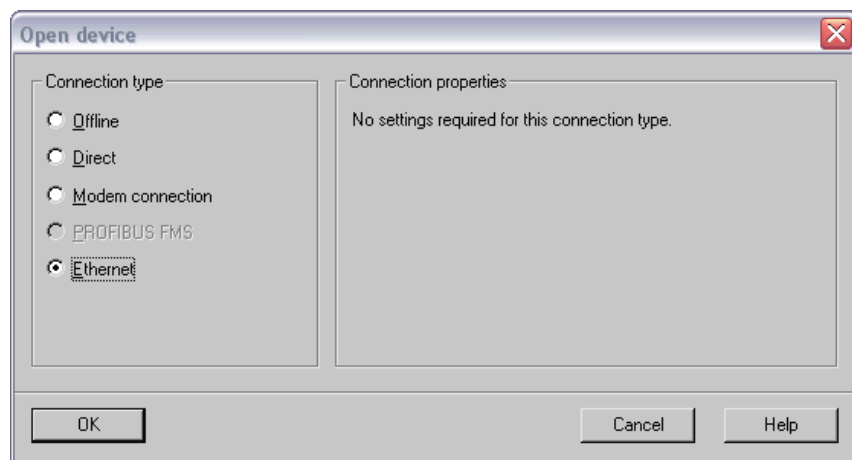
IP address and subnet mask must be manually defined with the direct connection via Ethernet

---

The subnet mask must be set up identically for the PC and the device. For our case we will select the value 255.255.255.0. The IP addresses will differ in the last position. For the device, we select, for example, 192.168.10.1, for the PC 192.168.10.10. You must define the IP address and the subnet mask for the PC in Windows under Network Settings.

One thing we want to make clear: The values named are examples, you can just as easily use different ones. What is important, however, is that you use the same subnet mask for each participant in a subnet and that each participant has a unique, valid IP address.

Now you have to initialize the device and it must be done via a **serial** connection. We've already described how this works in Chapter 10. Once the device is initialized, you can naturally exchange all additional data directly via Ethernet. But remember though, you'll need a **twisted** patch cable for such a connection.



Always good: The connection set up via Ethernet requires no additional settings

With the PC and device connected with one another via this twisted patch cable, you can start the connection build up. You do this as usual with a double-click on the device icon, for example. In the Connection dialog, choose the **Ethernet** option, click **OK** and you're off to the races.

---

**You should know, what we mean by a 1:1 mapping.**

When it's up to IEC 61850, a SIPROTEC information is often broken down into a series of individual information. Here an example. The indication **50/51 PH OFF** works in the following data objects and influences their status:

Protection/PIOC1/Mod  
Protection/PIOC1/Beh  
Protection/PIOC2/Mod  
Protection/PIOC2/Beh  
Protection/PTOC1/Mod  
Protection/PTOC1/Beh

The data objects of this information can be routed under the individual communicators in the DIGSI System Configurator. Since these are standard-conforming data objects, communicators of any vendor can communicate with one another.

In case certain information is only to be exchanged between SIPROTEC 4 devices, there is a simpler way of map information 1:1 to the syntax specified by IEC 61850. You route factory-specified information in the destination direction to the column **X**. Information routed in this way is assigned to the logical device **Extended**. The information has a standard-conforming syntax, its name components however are selected in such a way that, as a rule, it can only be understood by other SIPROTEC 4 devices.

The above-mentioned indication **50/51 PH OFF** appears much simpler in a 1:1 mapping: **Extended/PDGGIO28/SPCSO23**. This representation leans on the table of indications of the IEC 61850 that you are already familiar with. Your benefit in the 1:1 mapping: You only have to create one connection in the DIGSI System Configurator for the exchange of the indication **50/51 PH OFF** between two communicators.

**You should know, what you have to pay attention to when you make changes to settings later on.**

As soon as you modify a setting for a device, several further steps are required so that communication via Ethernet continues to work. Let's assume that you have changed the IED name of a device. Then it is necessary to update the ICD file of the device because the IED name is a component of the device description. To do so, open the device offline in the DIGSI 4 Gerätebearbeitung and then close it again. In doing this, DIGSI 4 brings the ICD file up to the current state. To be sure, this file also has to be updated when, for example, you make changes to routings. In this case, you are already working with the Device Editor so you don't have to give any more thought to updating.

The station description, as well, must get a face-lift. For you that means: Open the station and then close it again. If necessary, you have to adapt a few connections between these two actions.

---

Once again, the parameter sets of the individual devices participating in the communication have to be updated in DIGSI 4. Once you've done this, you can transfer the parameter sets into the devices. To do so, first open a device offline and then in the DIGSI 4 Gerätebearbeitung select the command **Device** → **DIGSI -> Device**. In the Connection dialog that appears, choose the option **Ethernet** and voilà you can transfer the parameter set into the device via the existing Ethernet connection.

**You should know,**

**how you can get more information and how you can purchase our great products.**

If you're interested in more information or you would like to purchase one of our products, here is a selection of order numbers on the topic of Ethernet and IEC 61850.

- Manual **DIGSI 4 Start Up**  
C50417-G1100-C152-A2
- CD **DIGSI 4 YOU - Start Up**  
E50417-A1174-C329-A1
- CD **SIPROTEC 4 YOU - Start Up**  
E50417-A1174-C215-A2
- CD **SIPROTEC 4 YOU - Ethernet + IEC 61850**  
E50417-A1174-C314-A2
- Ethernet Module **EN100-O** (optical version)  
C53207-A322-B180-1
- Ethernet Module **EN100-E** (electrical version)  
C53207-A351-D675-2
- DIGSI 4 **Basis**  
7XS5400-0AA00
- DIGSI 4 **Professional**  
7XS5402-0AA00
- DIGSI 4 **System Configurator**  
7XS5460-0AA00
- DIGSI 4 **Upgrade** from Basis to Complete  
(= Professional + System Configurator)  
7XS5408-0AA00

**Sorry to Say  
Goodbye ...**

Well, that's that once more! If we were able to help you out again this time around then we've achieved our goal. On this note, till we meet again in the next book!



## Looking Back

Halt, stop, WAIT! We almost forgot something! So that you have something to look back on what you've achieved, we've put together a compact sequence graphic for you here. You can also use this as a guide for your further projects.

