RTU500 series

RTU500 series Remote Terminal Unit
User manual
RTUtil500

Power and productivity
for a better world™

Revision

Document identity:		1KGT 150 801 V003 1
Revision	Date	Description
0	03/2013	Initial version for Release 11.0
1	06/2013	Codepage information removed
2	12/2014	New layout
		Updated chapter Excel interface for Release 11.3
3	01/2015	Updated Options dialog screenshot
		Updated screenshots in chapter 'Excel Sheet Block for Line Address and Host Parameters'
	10/2015	Added chapter 'PLC Engineering'

Contents

1 Getting started

1.1 RTUtil500

The basic topics of the RTUtil500 are:

- Configuration and data engineering tool for RTU500 series projects
- The principles of user interface structuring according to EN 81346-1
- The user interface of RTUtil500 is an application based on the Microsoft standard presentation format
- Documentation of all project steps
- External data interface concept
- Multilingual tool (user interface and help files)
- Delivery version with setup, installation and un-installation program on CD-ROM

The system requirements for the data engineering tool RTUtil500, particularly the free disc space, depends on the project size. Basic requirements are:

- Operating system: Microsoft Windows XP Professional with SP3 or Microsoft Windows 7 with SP1
- Memory: 64 MB RAM
- Processor: Pentium class
- Hard disc: > 200MB free disc space
- Hard lock: dongle (USB port) for MULTIPROG wt export

1.2 Installation

RTUtil500 is developed for the Windows operating systems and will be distributed as CD-ROM version. Next to the RTUtil500 setup program, there are additional software components on the CD. For installation of the complete RTU engineering platform (containing also the communication between PC and RTU) see also the documentation "RTU500 series Web Server User's Guide".

Usually the configuration menu starts automatically after inserting the CD to the drive. To start it manually, insert the CD to the drive, select the root directory and start "RTUtil.exe“.

To install only RTUtil500 manually, call the setup program from the distributed media. If the CD drive is „D:\“, the directory to start „setup.exe“ is „D:\Program\RTUtil500“.

After starting the setup program, follow the steps of the installation dialogs:

1. Acknowledge welcome page
2. Select the program path where the files should be installed
3. Select the program group of RTUtil500
4. Start copying of the files to the local hard disc

The setup program will copy all the files needed from CD to your local disc. Also all required registry entries are done by the setup program. To use Multiprog WT export interface of RTUtil500 a dongle is needed.

1.3 Getting help on RTUtil500

Applications written for Windows usually provide context sensitive help, allowing the user to get help on a particular window, dialog box, command or toolbar button. RTUtil500 provides this help as text messages shown in the status bar and tool tips. Tool tips are the tiny popup windows that present short descriptions of a toolbar button's purpose when the user positions the mouse on a button for some time. RTUtil500 provides all these features in the language chosen by the user.

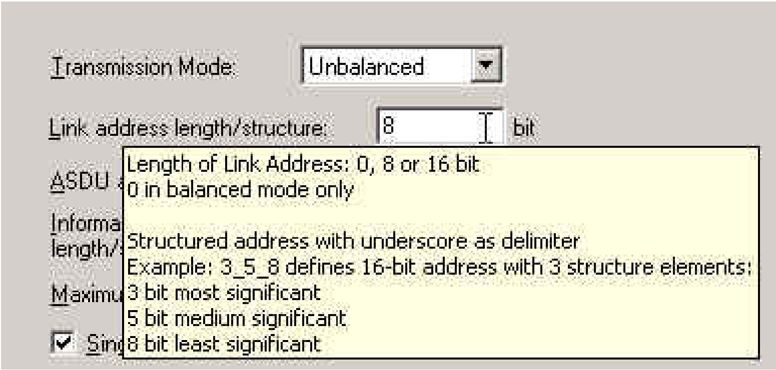


Figure 1: Example of a Tool Tip

2 General overview

2.1 RTUtil500 structure

RTUtil500 enables the user to control the whole engineering process of an RTU based system. All configuration data is managed from RTUtil500. To do so, several functional concepts for data storage, structuring and presentation are needed.

To meet the requirements, the internal software structure of RTUtil500 is split up into different function parts as shown in the figure below.

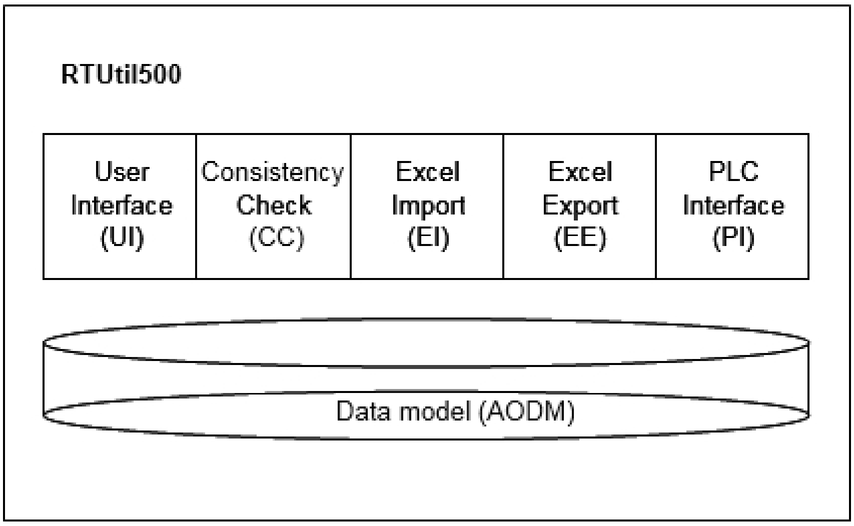


Figure 2: Overview RTUtil500 Components

2.2 General Data Structuring and View

The general view of the user to the engineering data is implemented on the basis of international standard EN 81346-1. This standard describes the structuring principles and reference designations for industrial systems, installations and equipment.

In the user interface this standard is presented in trees showing the RTU from different points of view. To describe the whole process the view is split up into three trees. EN 81346-1 defines how to split up a system (function-, product- and location-oriented structure).

The user interface structure offers three trees to describe the system structure usually used for an RTU.

- **Network Tree**
  The network tree shows the lines and protocols for routing the data points through the network.
- **Signal Tree**
  In the signal tree the location and designation of signals is shown. The signal location describes the place of the data points in the primary process.
- **Hardware Tree**
  The hardware tree presents the structure of an RTU with the levels cabinet, rack, board and the reference to the data points defined in the signal tree.

The structuring in trees allows a common presentation format and a general user interface of the RTU data and the environment.

2.3 General Tree Structure

The following example shows the implementation of structuring principles according to EN 81346-1. In "Fig. 3: Network - RTU - Process" the project views down to the RTU internal configuration are shown. This is only a small example of an RTU network. 'RTU 01' is the concentrator station in this network. The right side of the picture presents the electrical process (one bay in a station, with the double point indication Q0). This scenario will be built up in the three trees: network tree, signal tree and hardware tree.

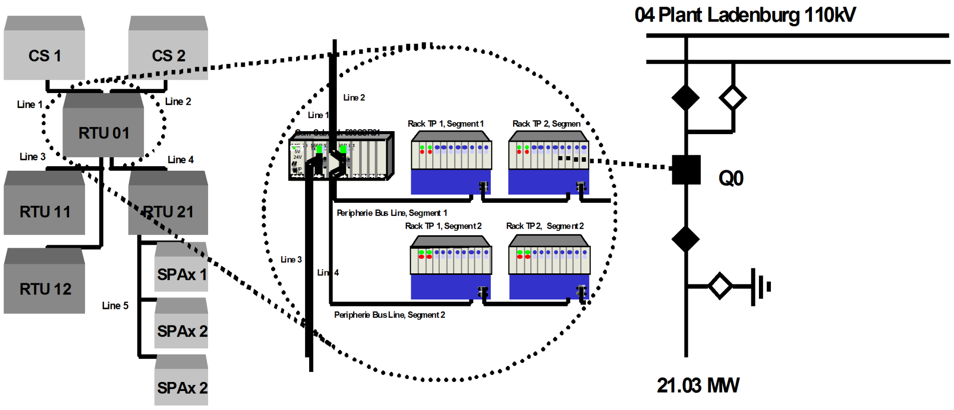


Figure 3: Network - RTU - Process

The root node of a tree is similar to the tree type (e. g. signal tree, hardware tree).

The network tree is the representation of the network structure. The concentrator 'RTU 01' is chosen as root node in the network tree, because the concentrator RTU is normally a unique starting point in an RTU station network. The hosts (central systems) are represented below the root RTU.

In signal tree the structuring of the electrical process and the naming of the single point indication is made. The name for every data point is derived from the structure of the electrical process.

Some nodes have different presentation locations (references) in several trees. These nodes are the line and station nodes from the network tree and the data point nodes from the signal tree. Line and station nodes have references in the hardware tree and in the network tree. The data point nodes have references in the hardware tree and in the signal tree.

The hardware tree describes the internal structure of the station nodes (e. g. 'RTU 01'). The RTU and line node types in the hardware tree are linked nodes from the network tree. The double point indication Q0 in the electrical process is linked from the signal tree to a binary input board in the hardware tree.

3 Data Engineering Concept

The engineering of RTU500 series data contains several dependency steps that demand a sequence in the data engineering process. To understand the engineering steps see also chapter "Engineering Example".

The basic data engineering steps are:

1 Project configuration (start)
- Set the project environment data.
- Before building the trees the signal tree structure has to be defined (number of levels and naming of these levels).

2 Build up the tree structures
- Build up the station network topology in the network tree. Choose the lines and the communication protocols between the stations. The network tree is needed for routing the process data points through the RTU network.
- Definition of data points in signal tree. The result of this definition is the unique object identifier (ObjID) for every data point.
- Definition of all RTU's and IED's with their data points in the hardware tree. The hardware tree contains the full description of the RTU hardware in detail with cabinet, racks and boards. To build up the hardware tree the standard data entry functions or data import functionality could be used.
- The link steps in the hardware tree build up the relations between the trees.
- Link data definition sources (e. g. RTU stations and IED's) and Line nodes from the network tree to the hardware tree.
- Link all the data points (e. g. single and double indications) from signal tree to the IO-boards of the hardware tree. To speed up data point linking more than one data point can be linked to a board in one step
- While adding or linking new data points to hardware tree the automatic signal routing functionality for this data point will be executed. The signal routing depends on the topology and the communication protocols in the network tree.

3 Set parameters, addresses
- Set the addresses for the process data points
- Set single parameters for several tree objects

4 Start the consistency check.
5 Build the RTU-files for each RTU.

During the whole engineering process RTUtil500 supports the user to avoid data inconsistencies. The basic functions that support the user during the engineering process are:

1 Avoiding to build up a wrong tree
2 Check ranges for each parameter
3 Check dependencies between the tree nodes
4 Check dependencies between parameters
5 Check the complete consistency of a project before generating the RTU-files

4 Tree Functionalities

4.1 The trees

The special concepts for the different trees are based on the general presentation format and general functions like add, copy, link or delete items.

The presentation of engineering data in different tree structures results in a complete new concept of the engineering process. General rules, as described in the data engineering concept, are required to build up a project and to use the powerful functionality of the trees.

4.2 Network Tree

A network tree enables the user to build up the whole RTU station network topology and to handle the connectivity data, the protocol parameters, and the filter functionality of a project. To do so, several functionalities are required. The user has to be supported by a set of functions and dialogs to enter the data for building up the network tree.

The network tree gives no information about the RTUs internal network. The station network, which is build up with the network tree, is clearly distinguished from the RTUs internal network, which is described in hardware tree. The network tree gives an overview of a complete station network. The internal structure of an RTU is built up in the hardware tree. All information about the connection between racks, the I/O bus network and the system bus configuration is part of the hardware tree.

4.2.1 Engineering Functions in Network Tree

- Build up station network topology
- Set protocols to line objects
- Route data points through the station network (this function can be launched from several engineering steps, see detailed actions description)
- Set filter functions for the data points

4.2.2 Network Tree Structure

The two standard tree node types to build up the network tree are the station and the line nodes. E. g. station nodes are like RTU stations, network control center (NCC) or IED’s (e. g. SPABus). From the users point of view some of the station nodes are only data sources (like IED’s), others are data consumers like the network control center and some could be both data sources and hosts.

With these two element types the complete station network topology can be built up in the network tree. The starting point for the network tree is the "node RTU" in a station network. The node RTU is the root node for the station network. More than one node RTU is possible but not common.

Via the host interface of an RTU several central systems could be connected. The assignment of board interface and line will be done in the hardware tree. The network tree only represents the general topology and connection of stations.

The sub interface allows connecting sub stations and IED’s. The hardware board with the interface also depends on the RTU type. The assignment will also be done in the hardware tree.

4.2.3 Actions in Network Tree

The network tree data entry is performed by the general functions (add, copy, link and delete nodes). In addition to these functions special views to the data points are needed, to select filter functionalities for data points.

To add the station and line nodes in the network tree the 'Add Item...' function is normally used. It is also possible to link an existing station from hardware tree to the network tree. By adding several stations and connecting them with lines, the station network topology is build up.

To choose a protocol for a line element, select the line object. Next to the tree window the parameter window (parameter pane) for the line object appears. Choose the protocol type for the line selected in this window.

To launch automatic protocol routing take the following steps. Add a new data point to an RTU in hardware tree. Add a new line to an RTU in network tree. Then the protocol routing for all data points of this line is carried out.

4.2.4 Links and Link Actions in Network Tree

Links in the network tree are made to the hardware tree, when the hardware configuration data of an RTU is entered. The usual way is to build up the network topology in the network tree and link the nodes to the hardware tree.

See the network tree as an overview of the complete station network and the hardware tree as a list of data sources for this network. All tree nodes except the central system node have to be linked to the hardware tree.

First link all data sources (RTU’s and IED’s) to the hardware tree. Afterwards the hardware structure has to be built, to link the lines to the RTUs communication interface boards (e. g. serial line interface). To assist the user in these link actions RTUtil500 gives information in the 'Link Item...' dialog, which elements from the network tree have to be linked to the hardware tree.

The user can get this information from the Link Item... dialog. In the Link Item... dialog a list of nodes that can be linked to the selected node and aren't linked to that node yet is presented. So, if the user selects the hardware tree root node and calls the link node dialog, he will get information about the not linked station nodes from network tree.

4.3 Signal Tree

The signal tree is built up to guarantee a unique object identifier for each data point in a system. The structure of the signal tree depends on the project. The specific signal structure is defined at the beginning of a project. The signal tree gives an overview of the amount of IO-data in a RTUtil500 project.

4.3.1 Engineering Functions

- Build up the process structure. Set the number of levels and their naming.
- Give a unique identifier to every data point in the system.

4.3.2 Structure

The signal tree structure is project dependent. The signal tree mirrors the location of the data points in the process. At the beginning of a project, the user has to build up the signal tree according to the project specific process structure. He determines the number of levels in this tree and their meaning.

ADVICE

These settings cannot be changed during the project any more.

The length of the data point name is configurable from 1 to 128 characters. The maximum length of the unique object identifier is 32 characters. It can start with a configurable position within the character string.

The goal of the signal tree is to get an unique object identifier for every data point in a system. For an electrical distribution process the structure could be as shown here.

Level	Description	Number of Characters	Identifier (example)
1.	Area	3	E2_
2.	Station	6	MANNH_
3.	Bay	6	TRAN10
4.	./.	17	Q0

Table 1: Signal tree structure

4.3.3 Actions

The "Project Wizard" assists users launching a new project and building up the particular signal tree structure. The number of levels can be selected and the types of these levels are specified here.

To add tree levels and data points to the signal tree, the common action functions like 'Add Item...' and 'Link Item...' are used.

Another common way to build up the signal tree structure is to import data from given lists. In this case the data has to be available in a common format like Excel and the signal tree structure has to be adapted to the data structure.

4.3.4 Links and Link actions

The function of the signal tree is to provide a unique object ID for a data point. Having obtained this object ID the user links the data points from the signal tree to the appropriate board in the hardware tree (e. g. a single point indication to a binary input board) and thus connects a data point with its data source (RTU, IED, etc.). During this process and with a correctly built network tree the automatic protocol routing action is triggered by the linking of a data point.

To support the user, the RTUtil500 provides information on how to link elements from the signal tree with the hardware tree and which boards are required to link the data.

4.4 Hardware Tree

The structure of the RTU hardware is displayed in the hardware tree. It informs about the internal structure of an RTU. The communication structure of an RTU and the IO-hardware needed for the data points shall be distinguished.

The structuring gives an overview of the whole RTU and enables the user to locate every data point in the station. All configuration data needed for a single RTU is presented in the hardware tree.

4.4.1 Engineering Functions

- Display the internal hardware structure of the RTU.
- Build up the RTUs communication structure: internal RTU network with communication boards, I/O bus, system bus,
- Build up IO-hardware: add data points (DPs) to the RTU or link them from Signal Tree.

4.4.2 Structure

The hardware tree structure is the hierarchical structure of the RTU. It starts with the RTU as one unit and includes the levels: cabinet, rack, board, interfaces and data points. Next to the tree nodes the parameter panes have the specific data entry controls needed to parameterize the nodes.

4.4.3 Actions

The general actions 'Link Item...' and 'Add Item...' are used to build up the hardware tree. The hardware tree can be built up separately (stand alone) from the network tree and the signal tree. For small projects with only one RTU and few data points no signal tree is needed.

To use the functionality of both, network and signal tree, (automatic signal routing and unique object-ID) it is recommended to build these trees first, and link their data to the hardware tree.

4.4.4 Links and Link Actions

The signal tree and the network tree are the data sources to build up the hardware tree. The network tree determines the dependencies of the communication. The signal tree defines the IO-data. The following engineering steps shall be distinguished: the communication data is built up first and then the IO-data.

The RTU's and lines are linked from network to hardware tree. The other direction is in some cases possible, too. It is important to use the same station in the network and in the hardware tree.

The links from the network tree have to be performed before linking the IO-data to the hardware tree, because the automatic signal routing goes through the station network. Automatic signal routing will be successful only if the communication hardware is complete.

Linking data points from signal tree to hardware tree will start the automatic signal routing, if the communication structure for an RTU is complete. The data points get the unique object ID from the signal tree. To speed up data point linking from signal tree to hardware tree a special link mechanism to link more than one data point is available.

4.4.5 Parameter in Hardware Tree

There is a parameter concept for every tree node. For several nodes in the hardware tree this parameter concept is extended.

RTU Segment (only RTU560):

The segment is an internal RTU structuring principle. For the RTU560 it is possible to set up 1-32 segments, each including 1-6 extension racks.

To build several segments for one RTU, different I/O bus lines have to be connected to a CMU board. An interface of one of these communication boards shall be assigned to one segment. There may be up to four segments. The I/O bus is able to connect six extension racks to one segment.

5 Batch Interface

5.1 Overview

It is possible, to start the configuration tool RTUtil500 out of a DOS window, in order to supply parameters to the application.

5.2 Open Project File

The configuration tool RTUtil500 will be started, opening a project file, whose name and path is supplied in the batch call. Afterwards the user can do the necessary manual changes.

Parameter: **-pr**

Call: **< Path to the tool RTUtil500 > -pr < "Project-file-name.rtu" >**

Example:

```
D:\Programs\RTUtil500\bin\RTUtil500 -pr
"D:\Programs\RTUtil500\proj\User_Manual.rtu"
```

This batch procedure call will result in the following actions:

1 The tool RTUtil500, installed at **D:\Programs\RTUtil500** will be started
2 The Project **D:\Programs\RTUtil500\proj\User_Manual.rtu** will be loaded.
3 After the project is loaded into the tool, the user can make the necessary changes.

5.3 Excel Import

The complete Excel import is a closed loop, which can be started and finalized automatically without any user action. The Excel import file will be loaded into a RTUtil500 project file, and the necessary configuration data for the download into the RTU (*.rcd) is generated. Additional optional parameters can be used, in order to decide if the plausibility check is necessary or not, or that the packed project file will be added to the configuration data. The complete Excel import batch procedure is initialized by the parameter **–it**. The following table describes the complete set of necessary/optional parameters.

Command	Meaning	Remarks	Option
-it	Import Type	Fixed parameter for the Excel-Import: XLS (Used for future extensions)	
-pt	Project File	Path and name of the project file (*.rtu)	
-if	Import File	Path and name of the Excel import file (*.xlsx)	
-tf	Target File	Path and name of the output project file (*.rtu)	
-rtu	Remote Terminal Unit	Name of the RTU, to be imported into the configu-ration	
-ln	Lines	The names of the Excel worksheets to be import-ed for this RTU. (Multiple choice possible)	
-kt	Keep Target	Do not overwrite the target file (*.rtu), if it already exists	Yes

Command	Meaning	Remarks	Option
-pcl	Plausibility Check Log	Perform the plausibility check, write the results of the check into the log-file	Yes
-rdf	RTU Download File	Path and name of the configuration output file (*.rcd)	Yes
-ip	Include Project	Include the packed project file (*.rtu) into the con-figuration file (*.rcd)	Yes

Example:

```
D:\Programs\RTUtil\bin\RTUtil500
-it XLS
-pt "D:\RTUtil_Files\RTUC.rtu"
-if "D:\RTUtil_Files\Excel_C_Sub.xlsx"
-tf "D:\RTUtil_Files\RTUC_Project.rtu"
-kt
-rtu "C"
-ln "Signals"
-ln "Sub_Line"
-pcl "D:\RTUtil_Files\RTUC_Project_Check"
-rdf "D:\RTUtil_Files\RTUC_Project"
-ip
```

These command lines will result in the following Excel import:

1 Use the project file "D:\RTUtil_Files\RTUC.rtu" (-pt "D:\RTUtil_Files\RTUC.rtu”).
2 Import the Excel file "D:\RTUtil_Files\Excel_C_Sub.xlsx" (-if "D:\RTUtil Files\Excel_C_Sub.xlsx”).
3 Store the results into the project file "D:\RTUtil_Files\RTUC_Project.rtu" (-tf "D:\RTUtil_Files\RTUC_Project.rtu”).
4 Do not over-write the project file, if it already exists (-kt).
5 Perform the import for the RTU with the name "C" (-rtu "C”).
6 Use the Excel sheet "Signals" for the import (-ln "Signals").
7 Use the Excel sheet "Sub_Line" for the import (-ln "Sub_Line").
8 Perform the plausibility check, write the results into the log-file "D:\RTUtil_Files\RTUC_Project_Check.log" (-pcl "D:\RTUtil_Files\RTUC_Project_Check”).
9 Create the configuration files for the RTU with the file names "D:\RTUtil_Files\RTUC_Project.rcd" (-rdf "D:\RTUtil_Files\RTUC_Project”).
10 Included the packed project file into the configuration file "D:\RTUtil_Files\RTUC_Project.rcd" (-ip).

5.4 CSV Import as Batch Procedure

The CSV Import can also be started and finalized automatically without any user action. The following table describes the complete set of necessary/optional parameters.

Command	Meaning	Remarks	Option
-pt	Pattern File	Path and name of the pattern file (*.rtu)	
-it	Import Type	Fixed parameter for the CSV-Import: CSV	
-if	Import CSV File	Path and name of the CSV import file (*.csv)	

Command	Meaning	Remarks	Option
-tf	Target Excel File	Path and name of the Excel file (*.xlsx)	
-kt	Keep Target	Do not over-write the pattern file (*.rtu), if it already exists	Yes
-rtu	Remote Terminal Unit	Name of the RTU for which the data should be imported.	
-ln	Line	The name of the line for which the data should be imported.	

Example:

```
C:\Program Files\ABB\RTUtil500\bin\RTUtil500
-pt "D:\RTUtil_Files\RTUCSVImport.rtu"
-it CSV
-if "D:\RTUtil_Files\StationXYZ.csv"
-kt
-tf "D:\RTUtil_Files\StationYXZ.xlsx"
-rtu "HostRTU"
-ln "Line 104 to CS1"
```

5.5 Build Configuration Files as Batch Procedure

RTU500 series configuration files can also be built automatically. The following table describes the complete set of necessary/optional parameters.

Command	Meaning	Remarks	Option
-pr	Project file	Path and name of the project file (*.rtu)	
-rtu	Remote terminal unit	Name of the RTU, the configuration files should be built	
-pcl	Plausibility check log	Perform the plausibility check, write the results of the check into the log-file	Yes
-rdf	RTU download file	Path and name of the configuration output file (*.rcd)	
-ip	Include project	Include the packed project file (*.rtu) into the configuration file (*.rcd)	Yes

Example:

```
C:\Program Files\ABB\RTUtil500\bin\RTUtil500
-pr "D:\RTUtil_Files\User_Manual.rtu"
-rtu "Router_Mannheim"
-pcl "D:\RTUtil_Files\ConCheck"
-rdf "D:\RTUtil_Files\Config"
```

-ip

6 User Interface

6.1 RTUtil500 User Interface

The tool for the RTU is a standard Windows application (Win32 application). The mainframe window includes all the views to the project data. Look and feel of the application's mainframe is known from other Win32 applications. Standard windows control mechanisms are used whenever it was feasible and useful, so that the user needs less time to start projects with the new tool.

The context help functionality is based on 'Tool Tips' and allows context sensitive help support.

The general data view and structuring supports the common engineering sequence. A user interface that contains all functionalities is the basis for the RTU configuration and the whole engineering process. The user interface includes the presentation objects and views for structured data presentation according to EN 81346-1.

This chapter gives an overview to the user interface presentation format. The main windows of RTUtil500 and the general tree structures will be shown in this chapter. Also an overview to the menus in RTUtil500 will be given.

6.2 RTUtil500 Workspace

The RTUtil500 workspace is a frame for a standard windows application. Where it is feasible, common window controls are used. The functionality will be completed by special functions required to configure an RTU.

"Fig. 4: RTUtil500 Workspace" gives an overview about the general construction of the RTUtil500 user interface. The several parts and their structure will be part of the following chapters.

The menu of the RTUtil500 user interface enables the user to select all functions, which are needed to configure an RTU. A shorter way to activate the same functions, which are available in the menus, is to select them directly with a toolbar button.

The tree windows allow the view to the several trees of an RTU project. There is a view to the network tree, to the hardware tree and to the signal tree available. Several tree windows may be open at the same time.

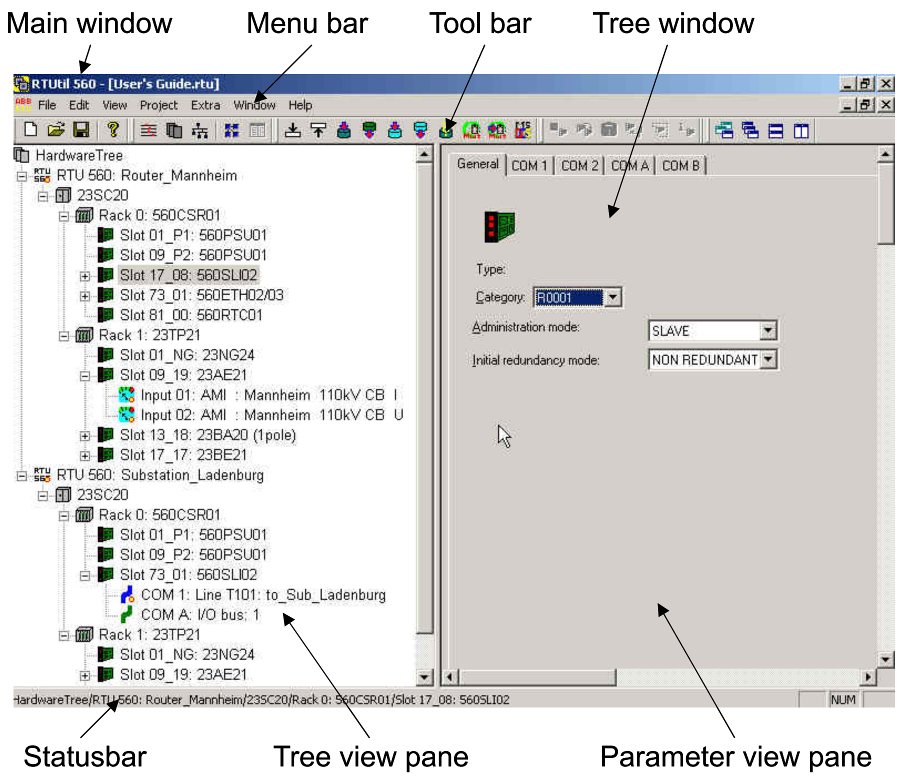


Figure 4: RTUtil500 Workspace

6.3 Data Presentation Windows

The general structure of data presentation format following EN 81346-1 is implemented in the different tree views. They are used building up the structure to configure RTUs, their signals, the station network up to the central system and the automation functions. The data in these structures is in hierarchical order, so that it is easy to present the items in a tree. The different views are:

- Signal tree
- Hardware tree
- Network tree

Next to the general tree windows there are list windows to support data entry functions. For data input and presentation there are different mechanisms to realize.

6.3.1 Tree Window

The views in an RTUtil500 project mirror the two general engineering steps. The first step is to build up the general data structure; the second is to parameterize the data. Between these two steps there are some automatisms from RTUtil500. By this general separation the working window is split into two parts called panes. These two panes support the common engineering steps:

1 Build up the structure of project data in trees - tree view (left side)
2 Enter properties and parameter to the nodes - parameter view (right side)

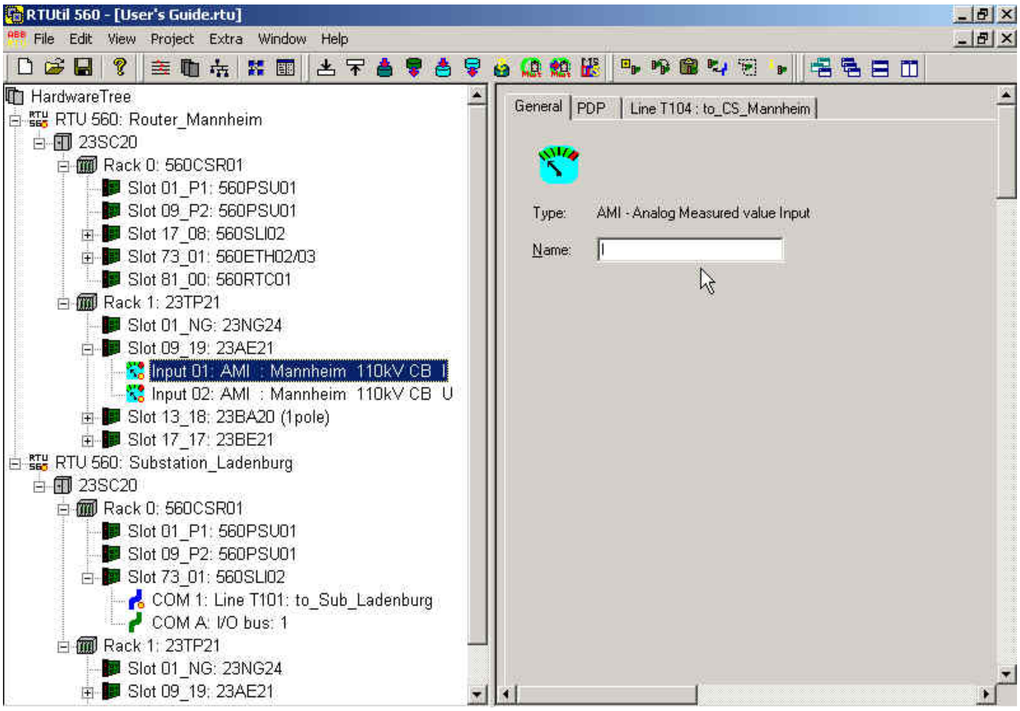


Figure 5: Tree Window

6.3.2 Tree View

In the left pane of a tree presentation window one of the tree views is shown. The user is able to open several windows with the same tree view or with different tree views. This functionality enables the user to show links between trees, copy item groups and find nodes.

The presentation style for every tree node gives the basic information that a user needs to identify the node. The structure of the general node presentation is:

- **Image for node type**
  An icon for the node type, e.g. an icon showing a 'board' or 'rack'. Icons are used for easy identification of elements, so that the localization of a tree node is no problem.
- **Local Type (alternatively)**
  The local type can be seen as predefinition for a node and gives information about the location of a node. It is used to limit the kind of types and number of nodes that can be added as children to an existing node. E.g. a '560CSR01' node has a local type 'Slot 10_P1' where only the 'Power supply unit board - 560PSU01' node can be added.
  The local type is a tree dependent part of the node presentation. This means that there can be different local types for one node. This behavior is only important for the linked node types like data points and line nodes. The data points normally have a local type in the hardware tree presentation. E.g. a single indication in hardware tree has a local type which describes if the data point is an input or an output. In the signal tree the local type of the hardware tree is not part of the presentation format for the same node.

- **Item Type**
  The node type specifies the behavior of a tree node. The node type is normally hard coded with an internal type ID. The user only sees the short description for a node type in the tree view. E.g. 'SPI' for a single indication.
  For some items that don't have a specific behavior and only are structuring help nodes, like the items in signal tree, it is possible to change the node type text. E.g. the default text of the level nodes ('Area', 'Plant') in signal tree will be defined at the beginning of a new project.
- **Item Name**
  The item name is an identifier for the tree items. Not all items in a tree could have a name. Names can be given to several node types.
  For data points the name could be a structured object identifier that is used in an external interface. To support structured naming special functionality is added in signal tree. The name for data point items can be divided in an auto naming part and a freely chosen part. The auto naming part is given from the signal tree structure and cannot be influenced directly. The path of the signal tree gives the identifier for a tree node, e.g. a single indication have a unique identifier from signal tree path, like 'E_1_110KV_0101'. The free part is a name for the tree item chosen by the user. Both name parts together have to be a unique identifier for a data point.

6.3.3 Parameter View

The right pane of the tree window is the parameter view. The general data entry forms give information about the properties and actual parameters of a node. Dependent on the item type and its localization in the tree, the dialog changes the presentation form. For easy node parameter handling, the dialog has a common structure for every node.

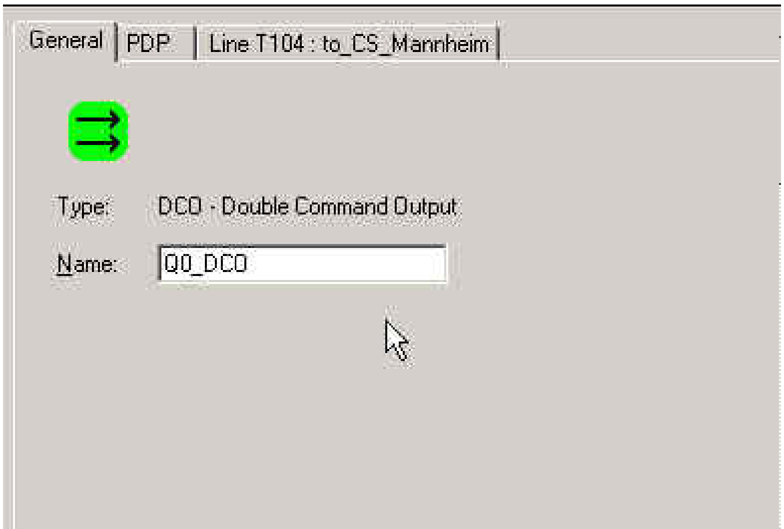


Figure 6: General page for a tree node

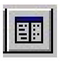
The presentation form is a property sheet. The first site for every tree item shows 'General' properties, like icon, item type and name. So there is a general view to get information of every item.

The other property pages actually available for the user depend on the parameters of an item. Normally there is one page for a group of parameters (e. g. process parameters 'PDP', protocol parameters 'Line...' etc.).

6.3.4 Data Point View

The view over all data points is available for one of the RTU's, IED's and Lines selected in the network tree. Another way is to select one of the RTU's or IED's in the network tree and list the data points from this view.

The '**Data Point List'** will be presented with the following dialogs:

- by menu, **View > Data Point List**
- by tool bar, click the symbol 

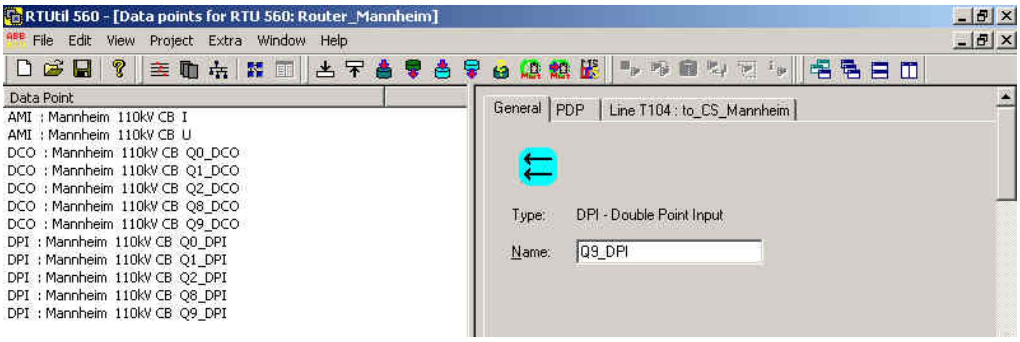


Figure 7: View of data point list

7 Moving in Trees and Selecting Items

To build up the data in trees there are rules how to move in the trees, input new nodes and change their parameters.

There are two fundamental differences in user control regarding the selection of an item in the tree. There is a very easy way with standard mouse control for users working the first time with RTUtil500. And there is control over all functions and moves with the keyboard, so that a person who is familiar with RTUtil500 can work very efficiently. Every function can be executed either with the mouse or the keyboard. In the following table the controls to move in the tree, select an item and switching between the two panes of working window are described.

Action	Mouse	Keyboard
Open branch	double click the item click the „+“ in front of the item	arrow key right
Close branch	double click the item click the „-“ in front of the item	arrow key left
select item	click item	move to the next item in the tree with the arrow keys up and down use page up and down for a faster move

8 Edit Functions

8.1 Add - Delete - Copy - Link Actions

The tree actions are the basic functionality's to build up every tree structure. The general functions are '**Add**', '**Delete**', '**Copy & Paste**' and '**Link an Item**'. These functionalities can be launched by several mechanisms described in the following chapters.

8.2 Add Item

To add items to a tree, the user has to select an item in one of the tree view panes. Then he can choose the start level for data entry. After this the '**Add Item...**' dialog for the selected node can be launched. There are four common ways, known from windows standard applications to get the dialog.

- by menu, **Edit/Add Item...**
- by tool bar, click 
- by shortcut keys **CTRL+A**
- with the right mouse button Pop up Menu, with the mouse click the right button, a pop up menu appears, select **Add Item...**

In the dialog the user can enter all the data one level below the selected tree node. The selected node is the parent node for the new entries. The add node dialog presents a list of possible child node types. The list of child node types depends on the parent node type and the actual structure.

After entering one item there is the chance to add the same node by clicking the '**Add**' button with the mouse or push the return key. The written item will be inserted in the tree structure. The dialog stays active at the selected hierarchical level. To stop data entry, push the '**Done**' button. The dialog will cancel and the control returns to the tree.

While the '**Add Item...**' dialog is open the user is able to move in the '**Tree View**' pane of the '**Tree Window**'. The dialog will change the format dependent on the selected node in the Tree View pane.

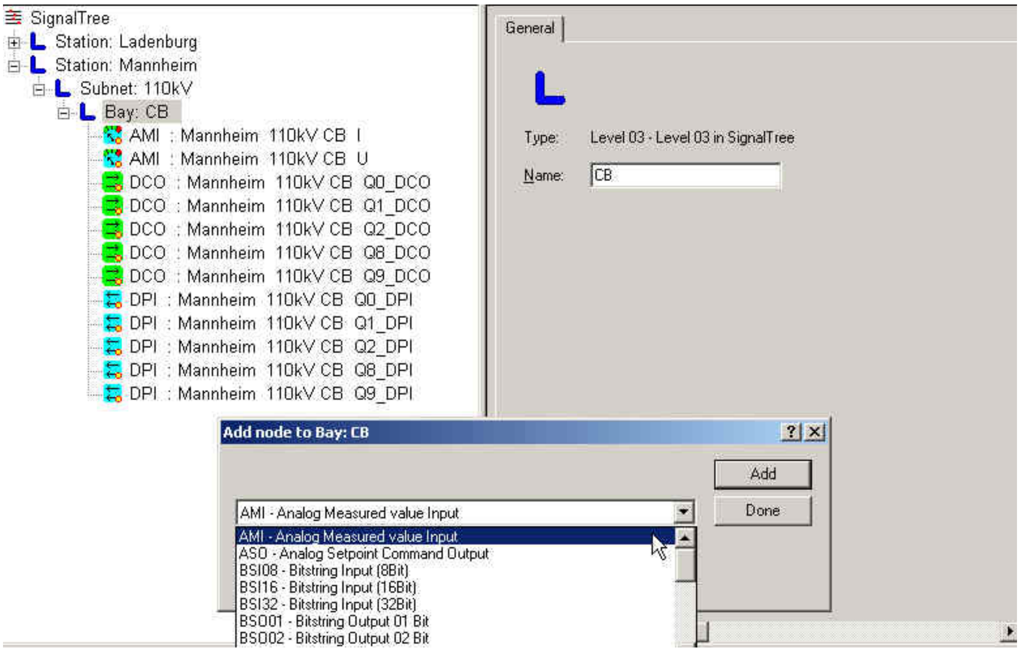
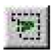


Figure 8: Add Item dialog

8.3 Delete Item (Branch)

To **delete** a node or a branch the user has to select a tree node or a group of tree nodes and call the delete item mechanism by one of the following ways:

- by menu, **Edit/Delete Item...**
- by tool bar, 
- by Shortcut key DEL
- by Pop Up Menu, with the mouse click the right button, a pop up menu appears, select **Delete Item...**

After selecting an item and calling the '**Delete Node'** function, a popup window appears and the user will be asked, if he is sure he wants to delete the selected item. If he presses the '**Yes'** button the item will be deleted. If the selected tree node has child items a second popup window appears and asks for confirmation to delete the whole branch.

8.4 Copy/Paste Item (Branch)

Copying nodes, branches and groups of items is an essential function to save time by doing same things only once. The copy concept of RTUtil500 gives the user a powerful function for data engineering. A basic rule is that the user is supported by indications from RTUtil500 during copying.

While copying a node only data is copied (no links will copy). There are no hidden functionalities and no automatisms are triggered, like it is the case when linking nodes between the trees.

If the node has child nodes, the complete branch will be copied.

The Copy/Paste action could be launched in several ways:

- by menu, first Copy the item to the clipboard, than Paste it to the target, **Edit/Copy Item**, **Edit/Paste**
- by tool bar, first Copy the item to the clipboard, than Paste it to the target, 
- by Shortcut keys, first Copy the item to the clipboard, than Paste it to the target, **CTRL+C**, **CTRL+V**
- by Pop Up Menu, select item, with the mouse click the right button, a pop up menu appears, select **Copy Item** to copy the item to the clipboard, than **Paste** it to the target
- by Drag & Drop, select the node with the mouse, keep left mouse button pressed, drag the item to the target, release the mouse button to drop the item. This is the Drag & Drop method.

The drag & drop functionality is the same as in standard Windows tools like the file Explorer.

If the node is copied via drag & drop functionality, only the **Drop** dialog appears, if the detailed position of the node is required (if a local type has to be chosen).

8.5 Link Item

Link Items in different trees in several places is one of the basics of structuring the configuration following EN 81346-1. So this function has to be used in every project. Some nodes of the project have to be inserted several times in different trees. These nodes are the station nodes (RTU’s and IED’s), the line nodes and all data point nodes.

After the given engineering example the station and line nodes will be linked from the network tree to the hardware tree. The data point nodes will have to be linked from the signal tree to the hardware tree.

The 'Link' action could be launched in several ways:

- by menu, **Edit/Link Item...**
- by tool bar, 
- by shortcut key, **CTRL+L**
- by Pop Up Menu, select item, with the mouse click the right button, a pop up menu appears, select '**Link Item...'**
- by Drag & Drop, select the node with the mouse, keep left mouse button pressed, drag the item to the target, press **STRG + SHIFT** to force link item, drop the item on the target

The selected item is the target node of the link action. To select the node that should be linked to the target node, the '**Link Item...'** dialog appears. In the drop down list of the edit control all the nodes that can be linked to the selected target node appear. The user has to choose a node of this list. To link the chosen node to the selected node target press the '**Link'** button.

If the node will be linked via drag & drop functionality, the 'Link Item' dialog only appears, if the detailed position of the node is required (if a local type is there).

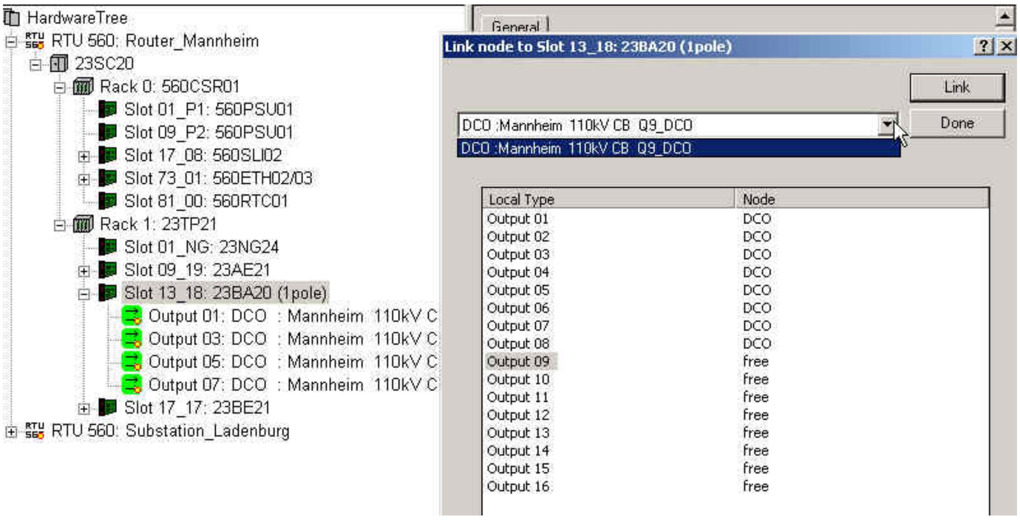


Figure 9: Link item dialog

After linking an item a small yellow circle on the icon of the item informs the user that the item is already linked.

 Line icon before linking

 Line icon after linking, marked with a red circle

8.6 Replace Item

The ‘Replace’ function is only available for some nodes e.g. for replacement of old communication units. The user can easily replace an item without need of re-configuring all node settings, adding and linking all sub-lines.

To **replace** a node the user has to select a tree node and call the replace item mechanism by one of the following ways:

- by menu, **Edit/Replace Item...**
- by shortcut key, **CTRL+R**
- by Pop Up Menu, with the mouse click the right button, a pop up menu appears, select **Replace Item...**

After selecting an item and calling the '**Replace Item'** function, a popup window with a list of replaceable items appears. The list is empty if the ‘replace’ function is not supported for the node.

After the user selected the new item type and pressed the 'Replace**'** button another security question appears if the item should really be replaced. By selecting ‘Yes’ the selected item is replaced by the new item.

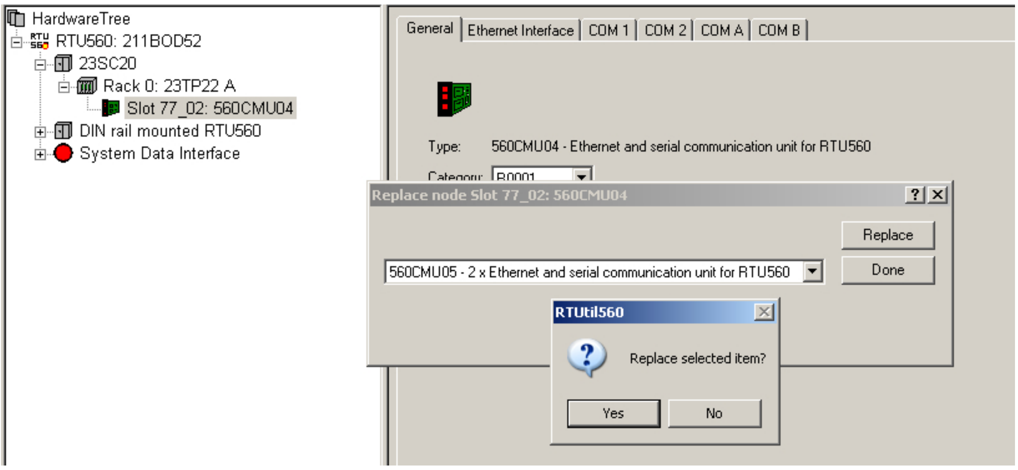


Figure 10: Replace item dialog

Position info, parameter settings as well as subordinated added and linked nodes are overtaken from the replaced element.

8.7 Move Item (Branch)

To reflect the correct position of DIN rail elements within parent DIN rail these elements can be moved. The function is only supported for DIN rail elements.

To **move up / move down** a node or a branch the user has to select a Din rail node and call the move item mechanism by one of the following ways:

- by menu, **Edit/Move up**
  by menu, **Edit/Move down**
- by Shortcut key **STRG+U, STRG+#** - move up
  by Shortcut key **STRG+D, STRG+#** - move down
- by Pop Up Menu, with the mouse click the right button, a pop up menu appears, select **Move up / Move down**

After selecting an item and calling the '**Move up'** function, the selected item is moved before the previous element in the tree view. Nothing happens if the selected item is the first item within the parent DIN rail.

After selecting an item and calling the '**Move down'** function, the selected item is moved after the next element in the tree view. Nothing happens if the selected item is the last item within the parent DIN rail.

9 Project Functions

9.1 Project Settings

The '**Project Settings'** dialog presents with the tab for 'Initialize Project' the general information of the project. In the tab 'Initialize Signal Tree' the defined data point structure for the project is reported.

To get the **Project Settings** dialog by menu, select **Project/Settings...**

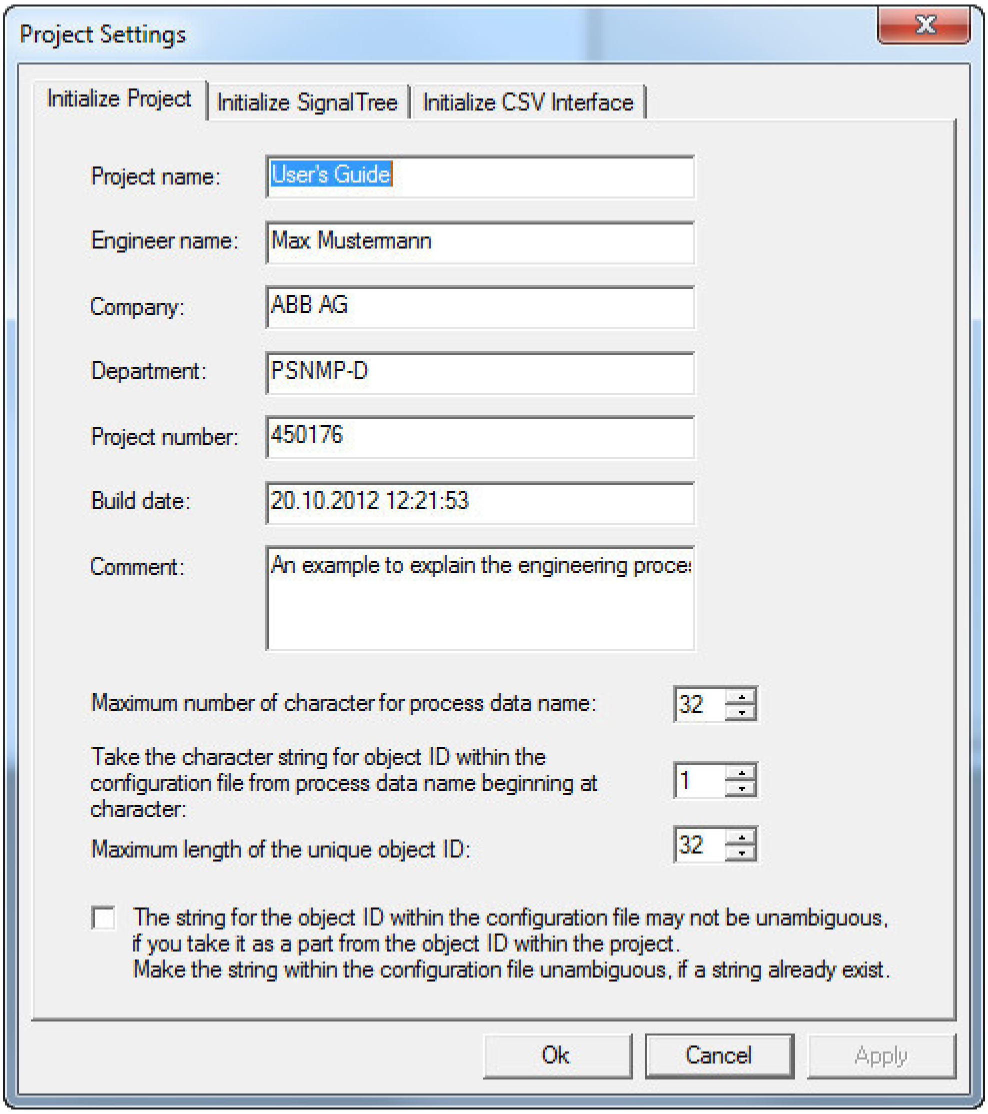


Figure 11: Project Settings Dialog

9.2 Check Consistency

Before starting the generation of the loadable files, a consistency check of the data is recommended.

Start consistency check:

- by menu, **Project/Check Consistency...**
- by tool bar, click 

The result of the consistency check is shown in a separate window with the following error levels:

- Error
- Warning
- Information

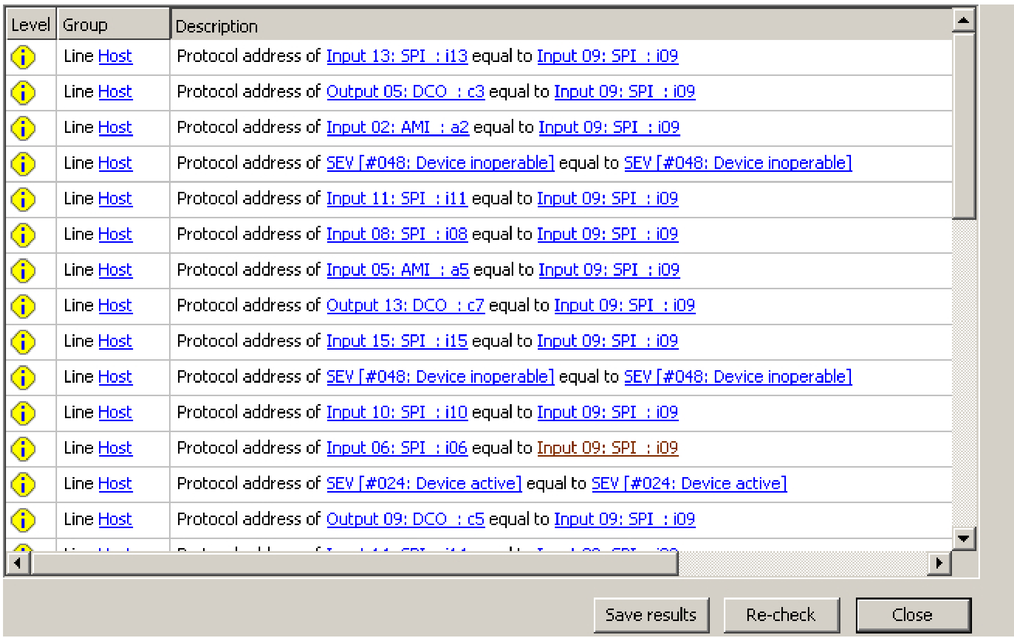


Figure 12: Consistency check results

The entries in this list are linked to the nodes in the trees of the checked project. The concerned node is selected, if the highlighted name within the concerned entry is clicking with the mouse cursor.

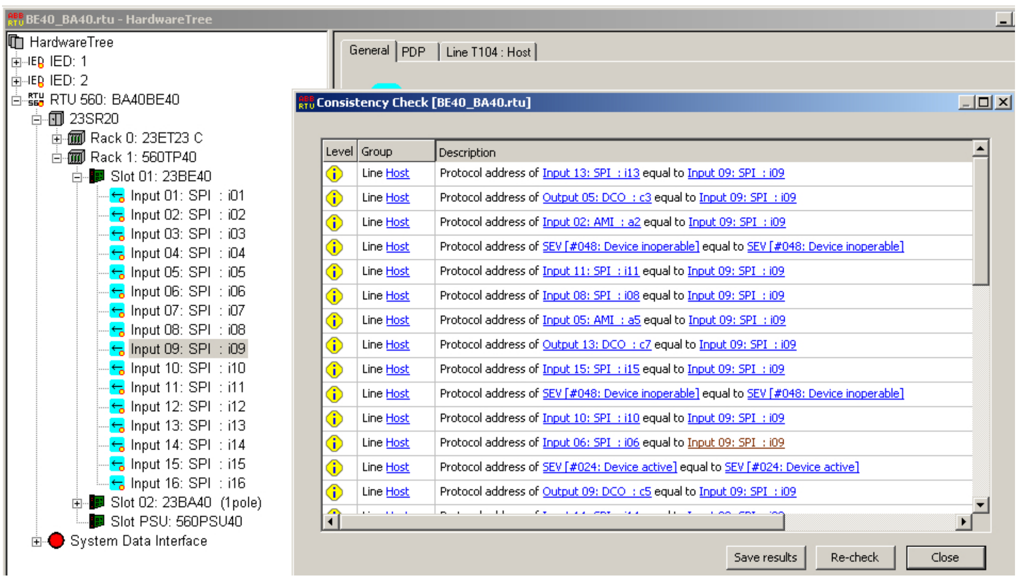


Figure 13: Links between consistency check results and user interface

9.3 Build the RTU Files

Start the generation of the RTU files:

- by menu, **Project/Build RTU file...**
- by tool bar, click 

Strings in RTU files are stored in UTF-8 format.

9.3.1 File Generating Steps

1 On the first page select the RTU from the project.

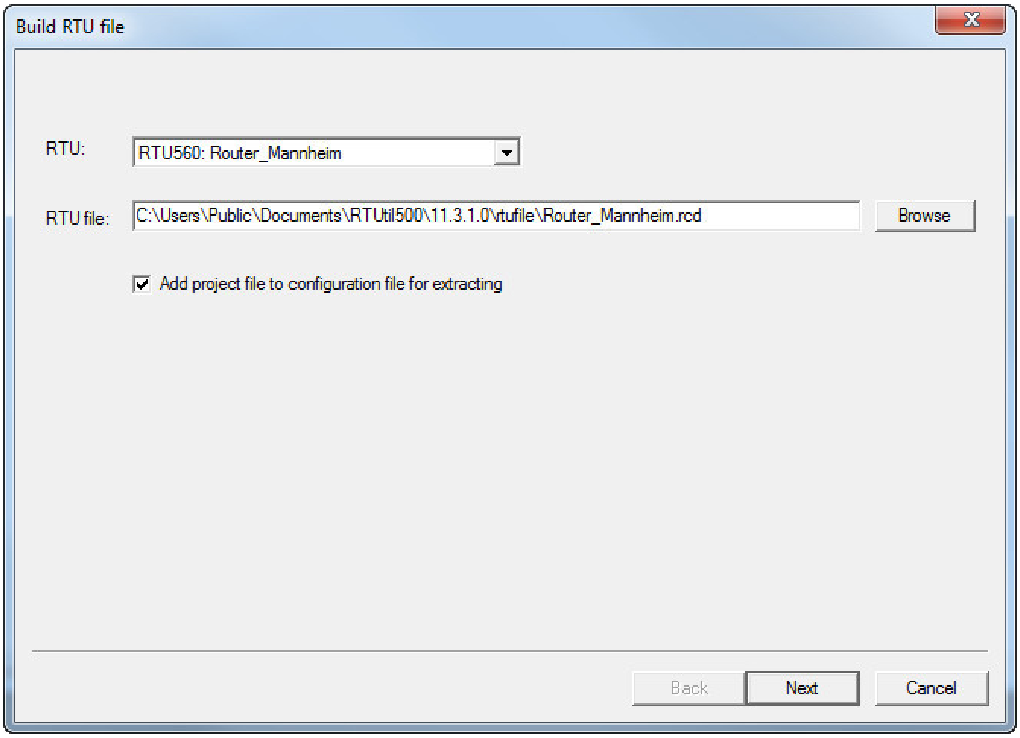


Figure 14: First page of the Build RTU file dialog

2 Choose a file name for the RTU file. If the option '**Add project file...**' is enabled, the whole project is saved (I/O data with all project information, see chapter ‘’). If the option is disabled only the I/O data will be stored in the RTU-file.

3 Click on **Next** button

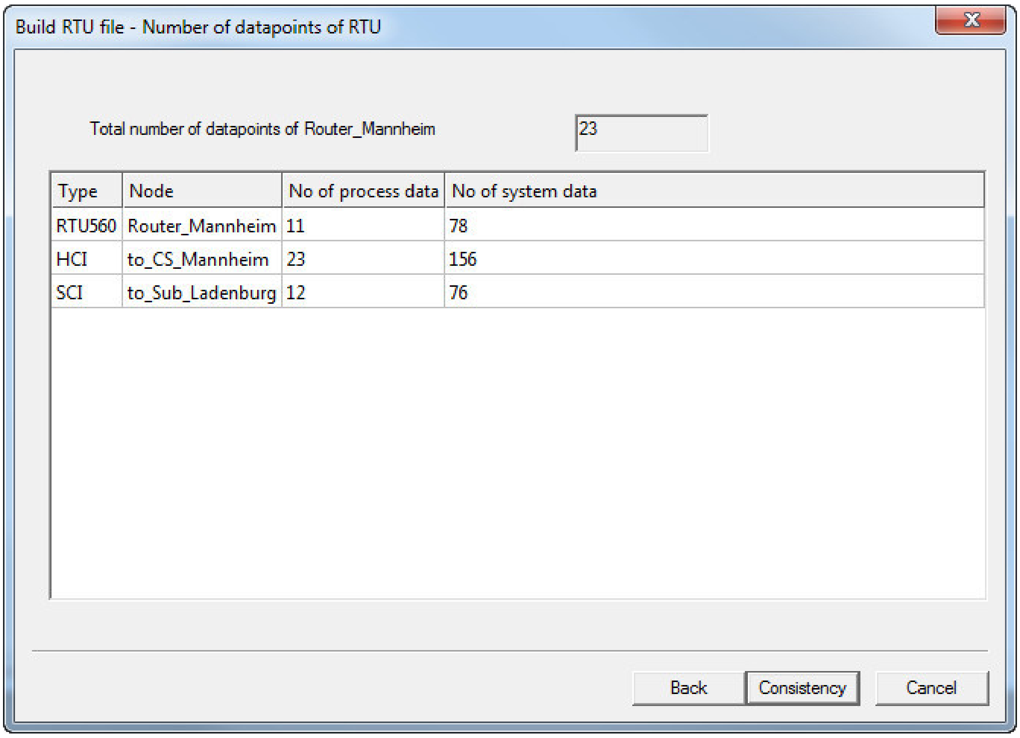


Figure 15: Second page of the Build RTU file dialog

4 Before the RTU files are generated the number of data points is calculated and a consistency check will be done by pressing the button 'Consistency'.
If the results of the consistency check should be saved, it is possible to specify a log file where the results are stored as text for further evaluation. This feature is by default disabled and can be enabled by selecting 'Write check result to Log File'.
More information about the consistency check can be found in chapter "9.2".

5 Acknowledge the message that the files have been generated and close the dialog with '**Ok'**.

9.3.2 The RTU-Files

The RCD-File (RTU Configuration Data) contains all the information to establish the communication. The items of the RCD-File are:

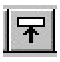
- Project information
- The RTU node with parameters and information of the system data
- Cabinet nodes
- Rack nodes with address and I/O bus connection
- Supply boards
- Communication boards with all parameter entries for the interfaces
- Line nodes with protocol specific parameters
- Sub- and Host station I/O devices with information of system data
- I/O devices of type I/O board
- Process data points with all parameters (PDP, protocol addresses)

10 Extra Functions

10.1 Extract the RTU files

RTUtil500 is able to read a RTU file *.RCD from the subdirectory ...\rtufile or any other user defined directory. RTUtil500 separates the project information from this file, if the switch 'Save information into RTU file for extracting a whole project' was enabled. The project information will be stored on the subdirectory ...\proj with the original file name or with a new name defined by the user.

Start extracting RTU file:

- by menu, **Extra/Extract RTU file...**
- by tool bar, click 

The standard dialog asks the user for subdirectory name and file name.

10.2 Data Interface – Excel Import

The RTUtil500 Excel data interface enables the user to use Excel sheets to manage RTU data. The interface imports data points to a given hardware template.

To do so, in a first step the hardware of the RTU is built up with the RTUtil500. The Excel import defines the process data points for the given hardware.

Start Excel Import:

- by menu, **Extra/Excel Import...**
- by tool bar, click 

Steps of Excel Import:

1 Start RTUtil500 Excel Import function from the menu or with the toolbar button.
2 Select the RTU
3 Select the Excel import file and click on the Next button

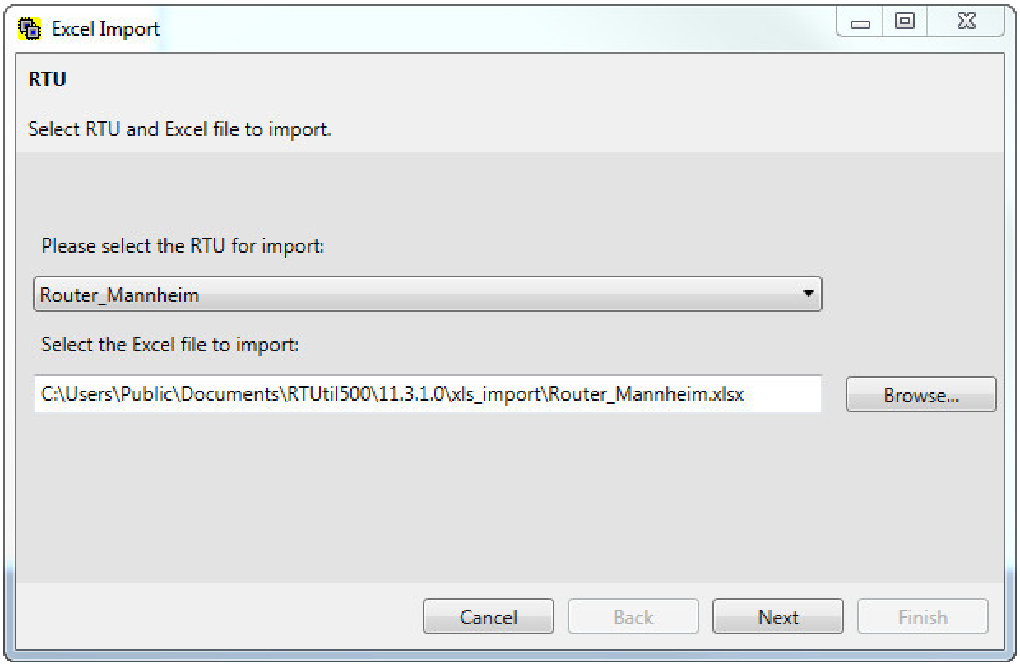


Figure 16: Excel import wizard step 1

4 Enable, disable log file generation and click on the Next button

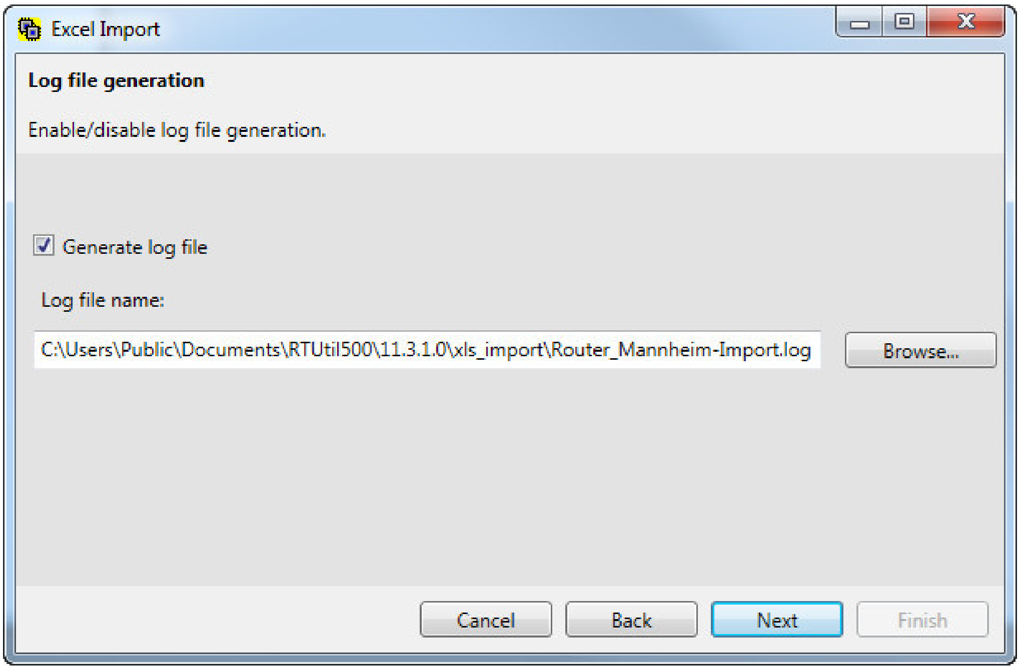


Figure 17: Excel import wizard step 2

5 The next page shows the import progress. After Excel import is done click on Open log file to get a detailed explanation, click on the Next button to see the Import result.

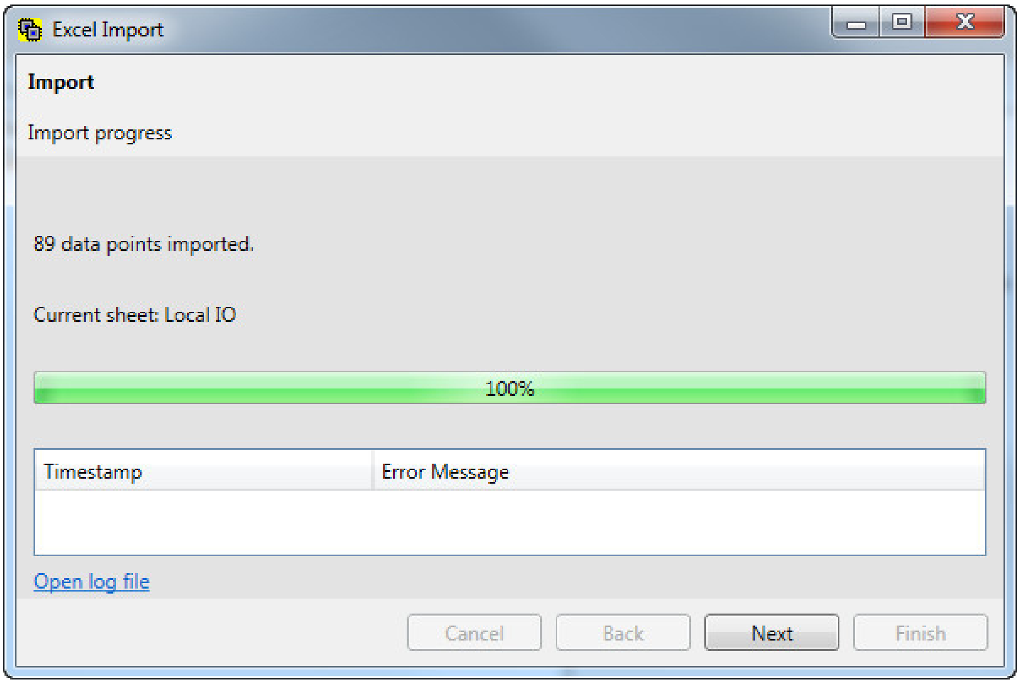


Figure 18: Excel import wizard step 3

7 The last page shows an overview of the changes done by the Excel import.

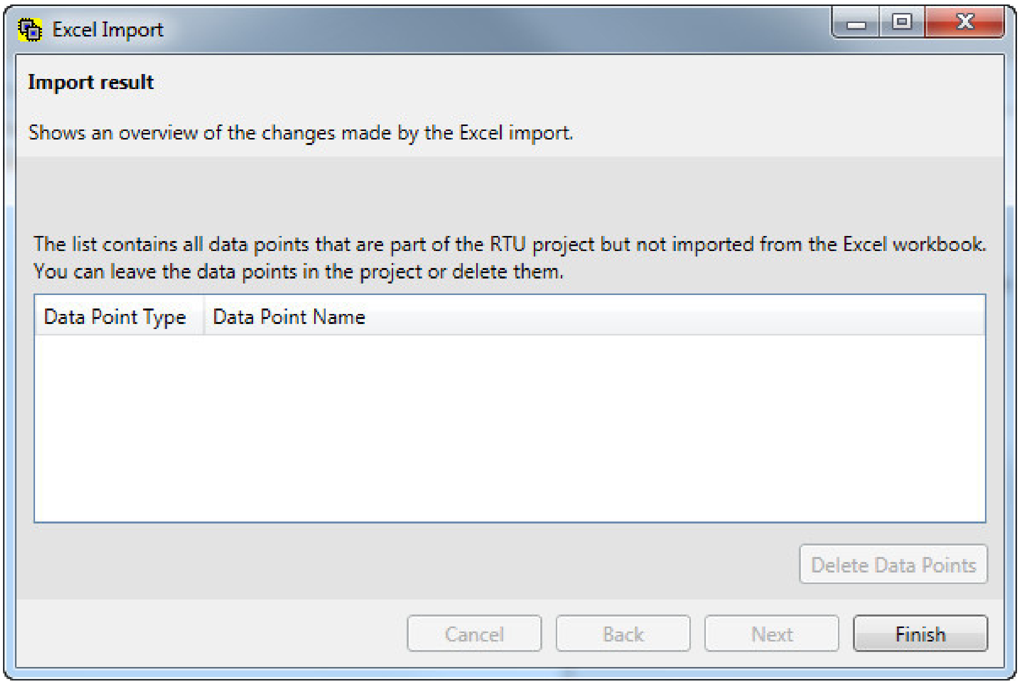


Figure 19: Excel import wizard step 4

10.3 Data Interface – Excel Export

Data points included in the project file can be exported to Excel worksheets by using the Excel Export function.

Start Excel Export:

- by menu, **Extras/Excel Export**...
- by tool bar, click 

Steps of Excel Export:

1 Start Excel Export function from the menu or with the toolbar button.
2 Select the RTU
3 Select the Excel export file and click on the Next button

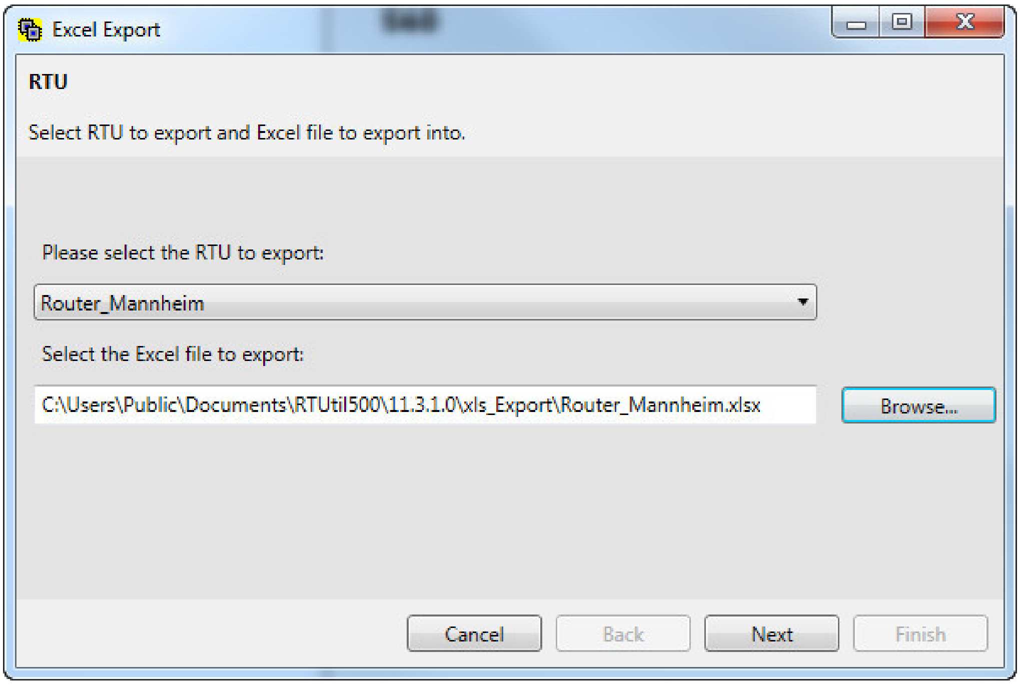


Figure 20: Excel Export wizard step 1

4 Enable, disable log file generation and click on the Next button

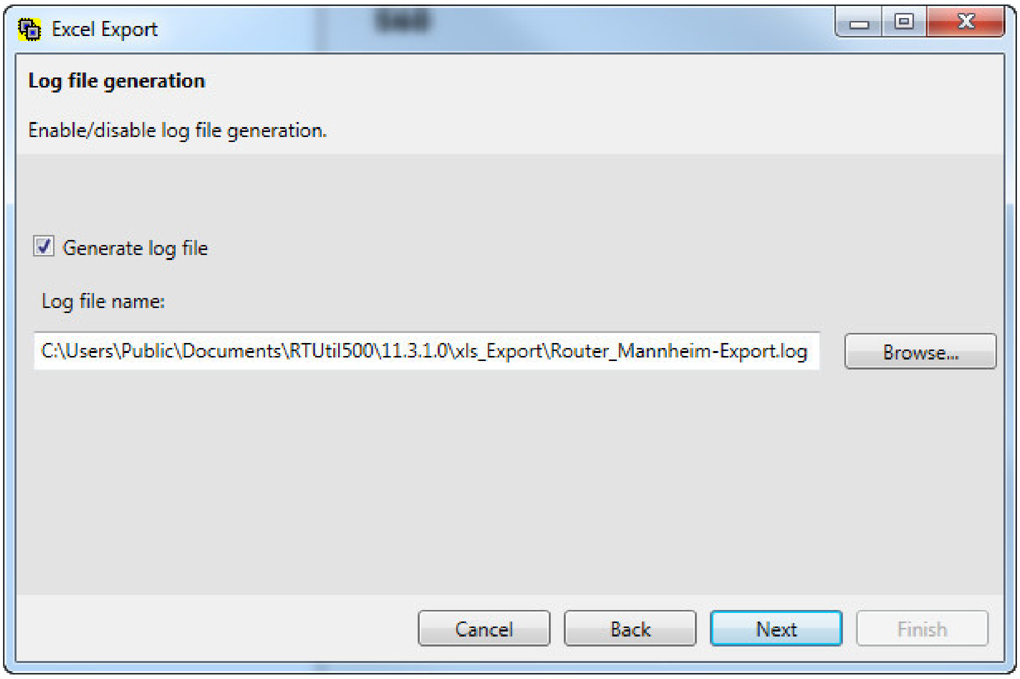


Figure 21: Excel Export wizard step 2

5 The next page shows the export progress. After Excel export is done click on Open log file to get a detailed explanation, click on the Next button to see the Export result.

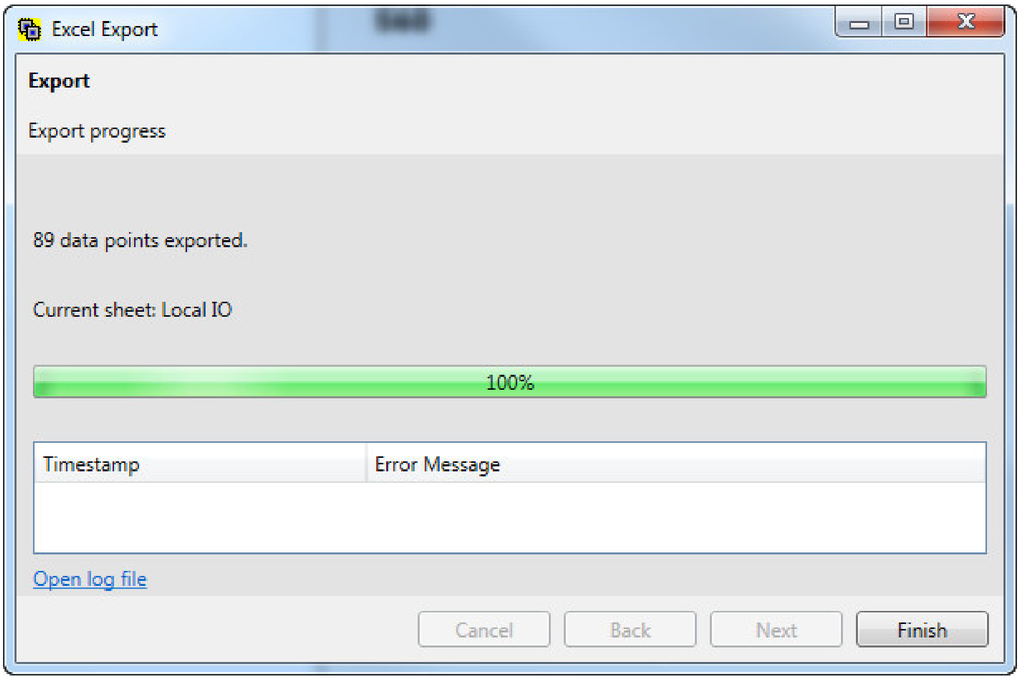


Figure 22: Excel Export wizard step 3

10.4 Data Interface – Excel Hardware Export

Hardware included in the project file can be exported to Excel worksheets by using the Excel Hardware Export function. The export is done into a predefined Excel file, available in the patterns subdirectory:

- …..\patterns\ExcelExport.xlsx for hardware data

Start Excel Hardware Export:

– by menu, **Extras/Excel Hardware Export**...

Steps of Excel Hardware Export:

1 Start Excel Hardware Export function for the project from the menu.
2 Select the RTU and click on Next button

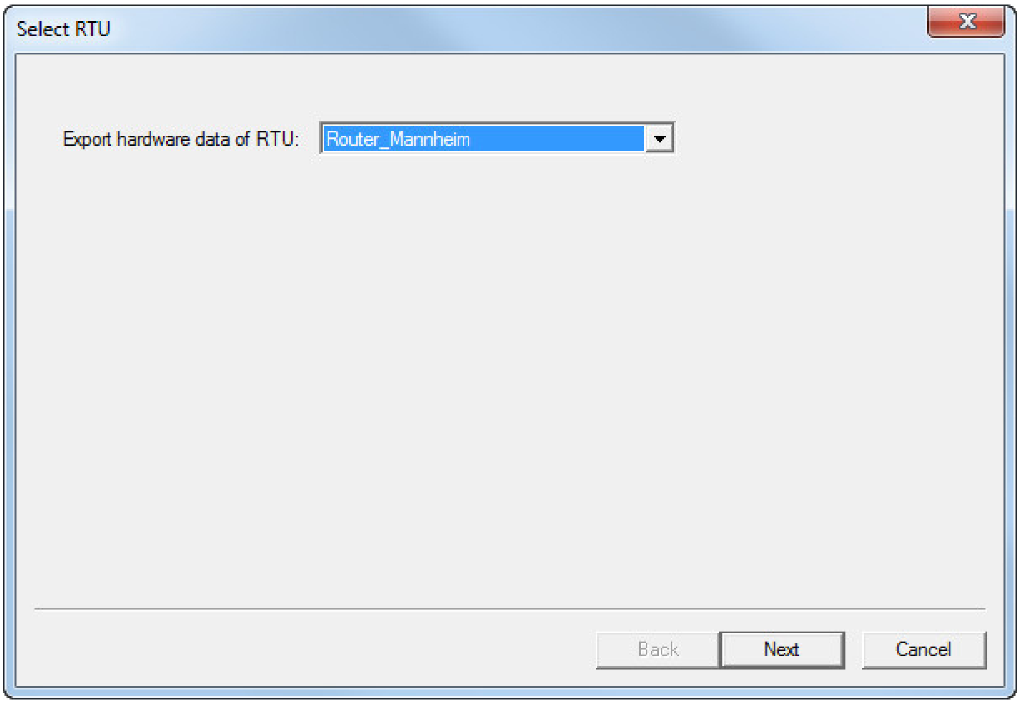


Figure 23: Excel Hardware Export wizard step 1

3 Enter path- and file name for the output file
4 The row number for the first entry should be at least 7.
5 Click on Start

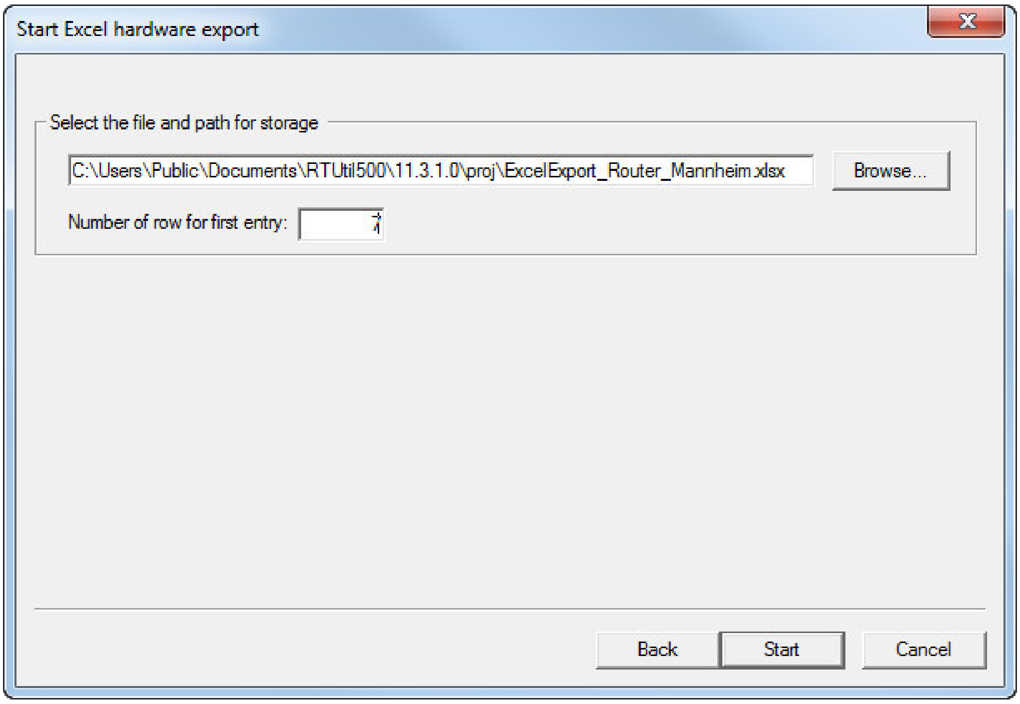


Figure 24: Excel Hardware Export wizard step 2

6 The export result is shown on the last wizard page.

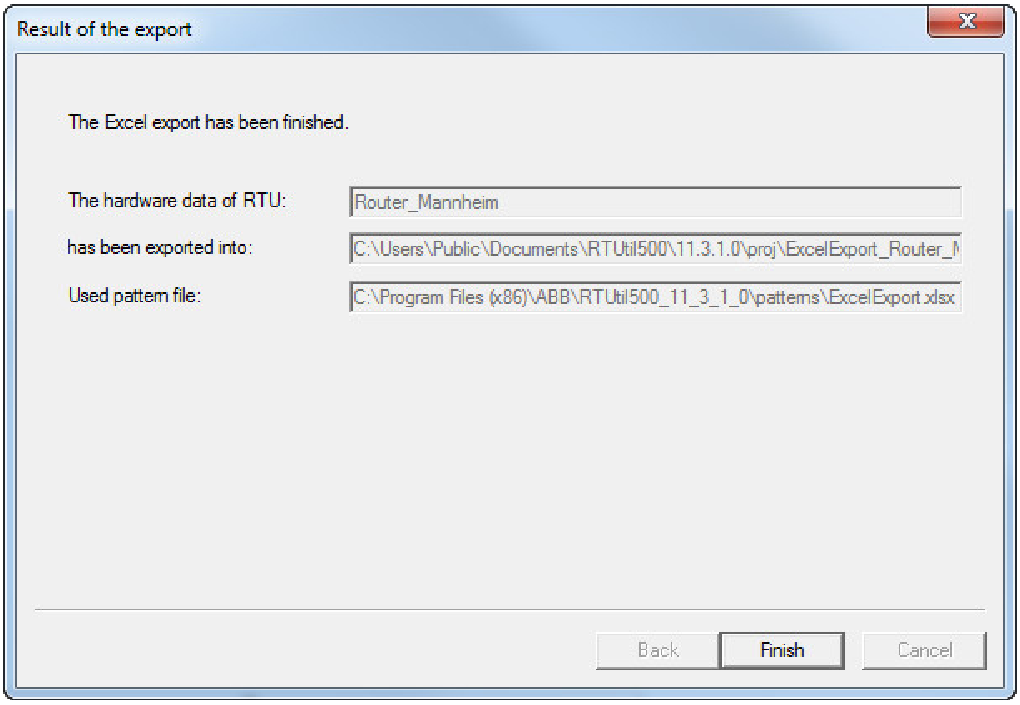


Figure 25: Excel Hardware Export wizard step 3

The exported Excel file contains four worksheets:

- Station Parameter sheet, Global parameter of the RTU
- Line Parameter sheet, Parameter of the communication lines

- Line Parameter General sheet, Parameter of the general communication lines
- HW Table sheet, Hardware parameter of the RTU

Unit type	Object	Name (only RTU, cabinet)	Name	Bus segment	Rack address	Slot / device address	IO bus address	Administration mode	Disable web server	CMU number	Initial redundancy mode	Redundance partner board	Mode	N out of 16 check (only 23BA20 board)	CSC of relays 1..8 (only 23BA20 1 pole)	CSC of relays 9..16 (only 23BA20 1 pole)	CSC (only 23BA20 2 pole, matrix)
								Mode						23 BA 20 Parameter			
Object		Name		IO bus address				only communication board						only 23BA20 board			
RTU types, Cabinet types, Rack types, Board types		32 ASCII		1..32	0..7	0..32		Master, Slave	Y/N	1 .. 16	Active; Standby, non redundant	64 ASCII		1..15	10 ASCII	10 ASCII	10 ASCII
HWUT		*HWNA*		*HWBS*	*HWRA*	*HWSA*		*HWAM*	*HWDW*	*HWCN*	*HWRM*	*HWRB*		*HWNO*	*HWC1*	*HWC2*	*HWCS*
RTU560		Router_Mannheim															
23SC20		Cabinet_1															
560CSR01																	
560PSU01				1	0	11											
560CMU05				1	0	9			N	1	NON	0					
560CMU05				1	0	8			N	8	NON	0					
23TP21																	
23NG24				1	1	21											
23AE21				1	1	19											
23BA20 (1pole)				1	1	18								-1	0	0	
23BE21				1	1	17											

Figure 26: Example Excel HW Table sheet

ADVICE
It is not possible to import hardware data into RTUtil500 project file.

10.5 Data Interface – MULTIPROG wt Export

The PLC information of an RTUtil500 project has to be exported in a project file with the extension "*.mwt".

Select the menu entry 'MULTIPROG wt Export...' of the menu bar element 'Extras' in RTUtil500. The MULTIPROG wt Export wizard starts.

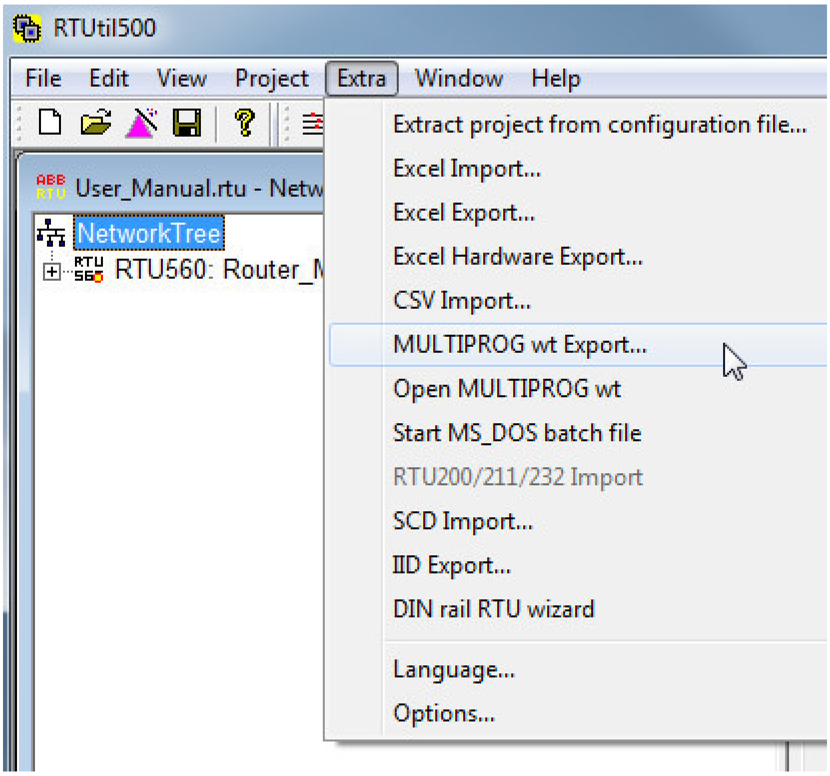
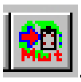


Figure 27: MULTIPROG wt Export menu entry

You can also use the toolbar icon .

The following dialog asks for the MULTIPROG wt project.

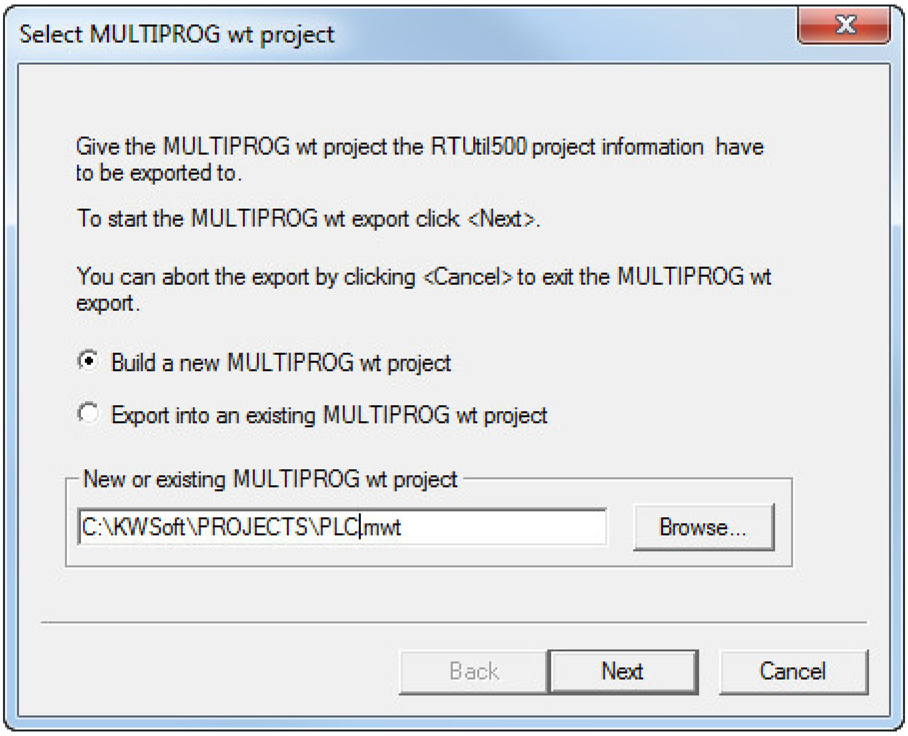


Figure 28: Select MULTIPROG wt project

You can build a new MULTIPROG wt project and export into this project or you can select an existing project for the export. The extension of the MULTIPROG wt project is "*.mwt".

Now click the 'Start' button and the export function runs and presents the following result dialog.

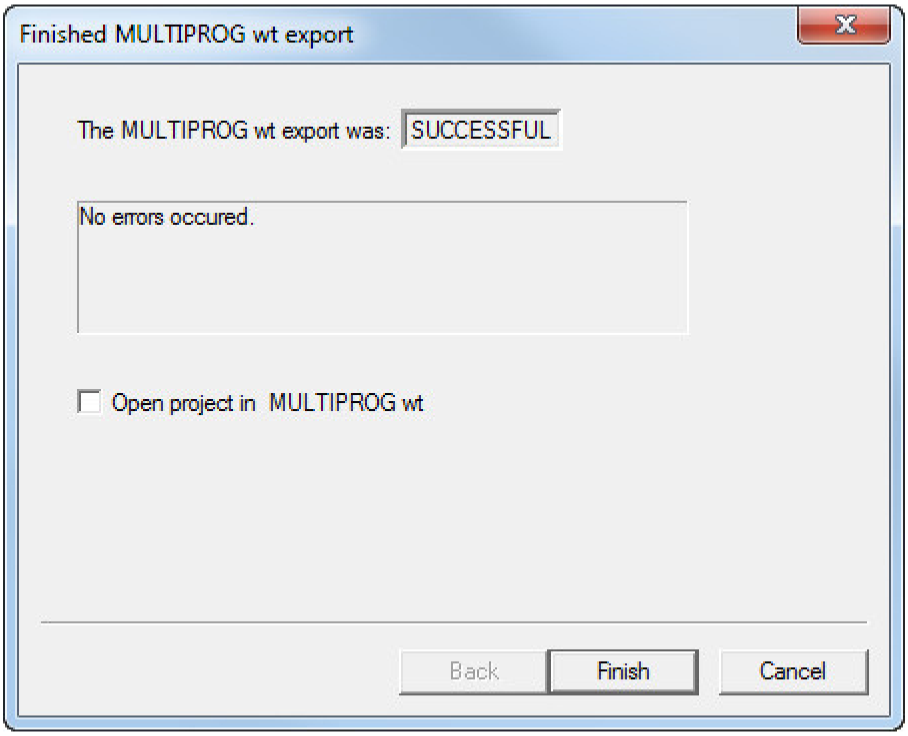
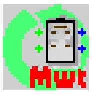


Figure 29: Finished MULTIPROG wt export

The last dialog shows the result of the MULTIPROG wt export. The export can be successful or maybe was aborted. If it was aborted an error description appears. When 'Open project in MULTIPROG wt' is selected the MUTLTIPROG wt project will be open after clicking the '**Finish'** button.

10.6 Extra – Open MULTIPROG wt

Starts RTU500 series PLC programming and test tool MULTIPROG wt.

Related toolbar icon to start MULTIPROG wt: 

10.7 Extra – Start MS_DOS batch file

Starts – if present in the RTUtil500 program directory path's subfolder \batch – after reply for batch processing parameters, the DOS batch file RTUtil500_Batch.bat.

Related toolbar icon to start the batch job: 

10.8 Extra – RTU200/232 Configuration Import

The configuration tool RTUtil500 can be used to convert configuration files of the RTU200 or RTU232 to a standard RTU500 series project file.

Related toolbar icon to start the configuration import: 

The configuration import for RTU200 or RTU232 configuration is described in detail in the ‘User’s Guide’ document 1KGT 150 612.

10.9 Extra – SCD Import

This function is in RTUtil500 the starting point for the engineering of an IEC61850 sub device communication interface. The detailed description of the IEC61850 engineering process could be found below in the chapter “IEC61850 Engineering”.

10.10 Extra – Language

In this dialog, the language of RTUtil500's user interface is selected. The selection takes effect after restart of RTUtil500.

RTUtil500 menu item: Extra/Language...

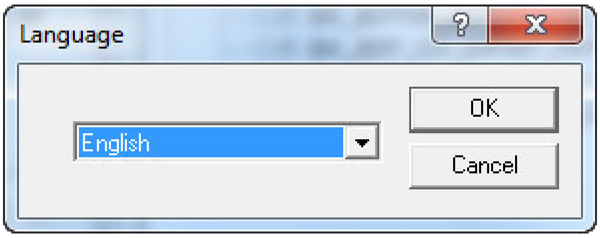


Figure 30: Language Dialog

10.11 Extra - Options

Change the saving directories for Project Files and RTU-Files. The path where Microsoft Access is installed and the path for MULTIPROG wt projects can be changed in this dialog, too.

Get the general options dialog from menu 'Extra/Options…'.

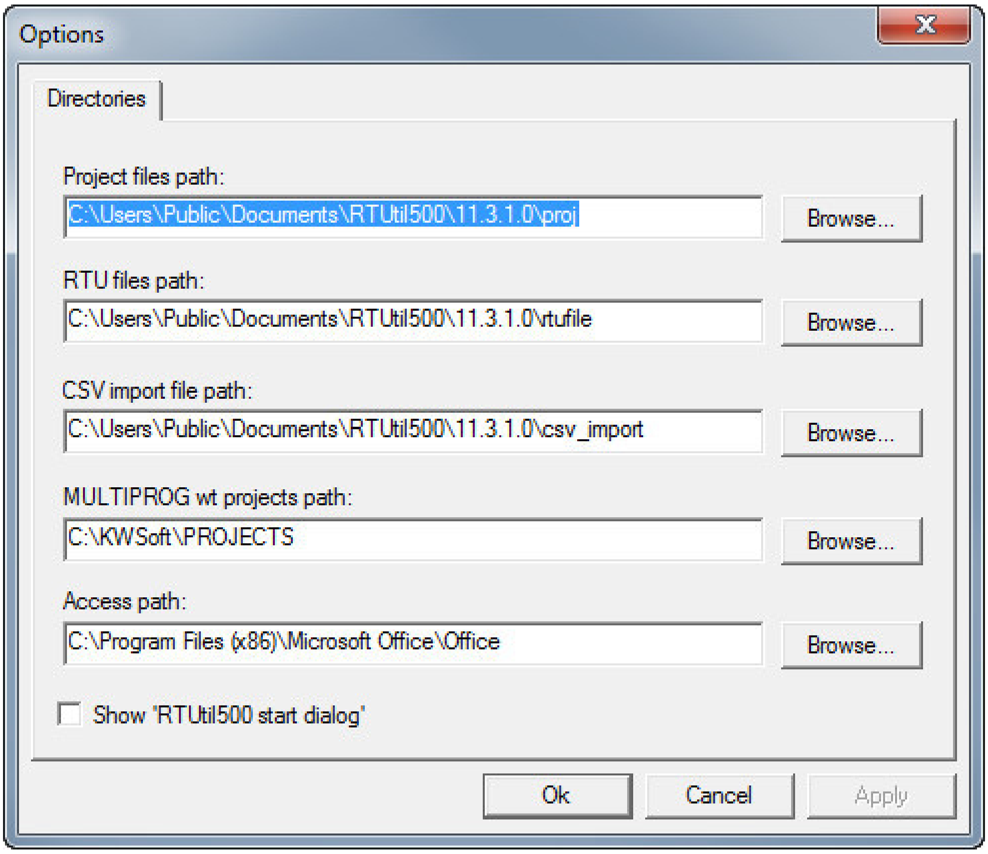


Figure 31: General options dialog

11 Excel Interface

11.1 Excel Import Introduction

Microsoft Excel is a suitable and often used tool for the definition and documentation of engineering data.

RTUtil500 provides an Excel interface to import the RTU relevant engineering data from Excel sheets.

Following chapters describe

- the structure of the "Excel Files and Sheets"
- steps in "Performing the Excel Import"

11.2 Excel Import Overview

The Excel import uses the Excel file(s) together with an RTUtil500 project file:

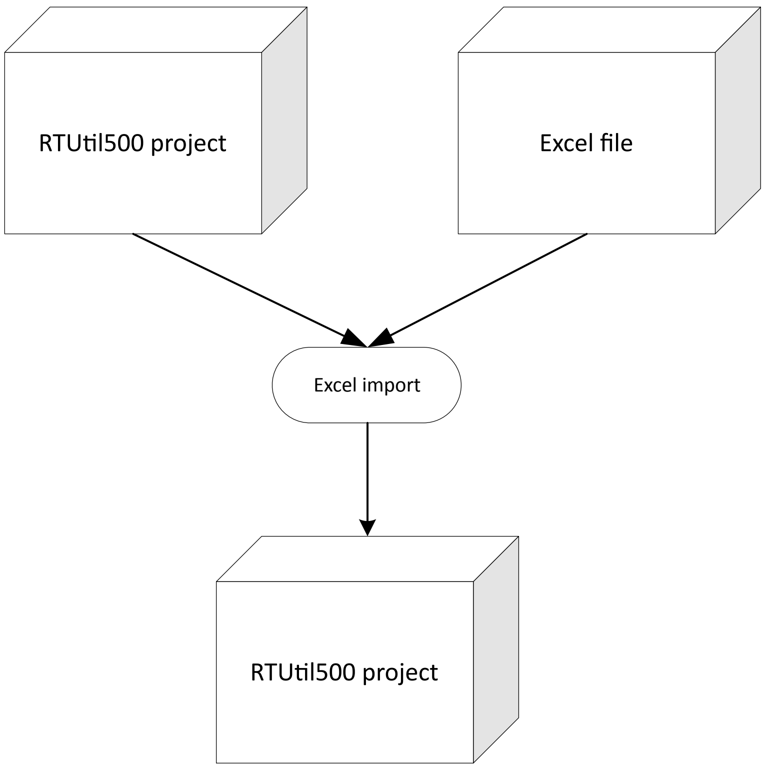


Figure 32: Sequence of an Excel import

ADVICE
As prerequisite for the Excel import an Excel import file is needed. Do an Excel export from your RTUtil500 project to create this Excel import file (Menu: 'Extra - Excel Export)

The Excel file(s) contains information for

- the project signal tree (signal names structured according to the project settings of the project pattern)
- the data object's hardware assignment to RTU I/O points or IED’s
- the data object's processing parameters
- communication protocol message addresses and parameters of each data object for all levels of the network tree

The project file has to contain

- the complete network tree (including all links to the hardware tree)
- the hardware tree down to RTU board level and IED level

Steps to perform an Excel import are:

1 build up the RTUtil500 project file, if it not exists
2 run Excel export to get an Excel import file
3 fill in all the information to be imported to the RTUtil500 project in the Excel sheet
4 perform the Excel import (RTUtil500 menu 'Extra – Excel Import')
5 for further modifications and extensions of data points step 3 and step 4 can be done again
6 For extensions in the hardware setup (e. g. an additional I/O module has to be added) the RTU-til500 project file has to be modified

11.3 RTUtil500 Project File

For the Excel Import, an RTUtil500 project file has to exist.

This file has to contain:

- the complete network tree
- the hardware tree down to RTU board level or IED level including all links to the network tree

This file can contain:

- data objects in the hardware tree (local I/O signals and I/O signals of the IED’s)
- data object related processing parameters
- communication protocol identifications/addresses and related data object parameters

11.4 Excel Files and Sheets

A draft for the Excel Import can be generated via the Excel Export.

ADVICE
Automatic conversion of Microsoft Excel 97-2003 Workbook (.xls) is not supported. Please open your Excel 97-2003 workbook in a current Microsoft Excel version (2007 - 2013) and save your workbook in the current file format (.xlsx).

11.4.1 Contents of Excel File and Sheets

For each RTU of an RTUtil500 project, a separate Excel file which contains at least one sheet has to be provided.

Sheets within the Excel File

For each subordinated IED line (a line that connects directly to IED's, not to lower level RTUs), one sheet has to be added. If the RTU has two subordinated IED lines, the excel file consists of three sheets: one sheet for the RTU's local I/O and one for each subordinated line.

Sheet Contents

Each data point with all its processing parameters and related communication addresses on all higher level lines of the network tree is specified in only one sheet of the set of Excel files and sheets belonging to a RTUtil500 project. This specification has to take place in the relevant Excel sheet (Local I/O or IED line) of the Excel file that belongs to the lowest level RTU. For each higher level line of the network tree, a protocol specific address/host parameter block has to be added to the sheet where the data point is defined.

11.4.2 Examples

Example 1: CS - RTU - Sub-RTU

The following network topology needs two Excel files:

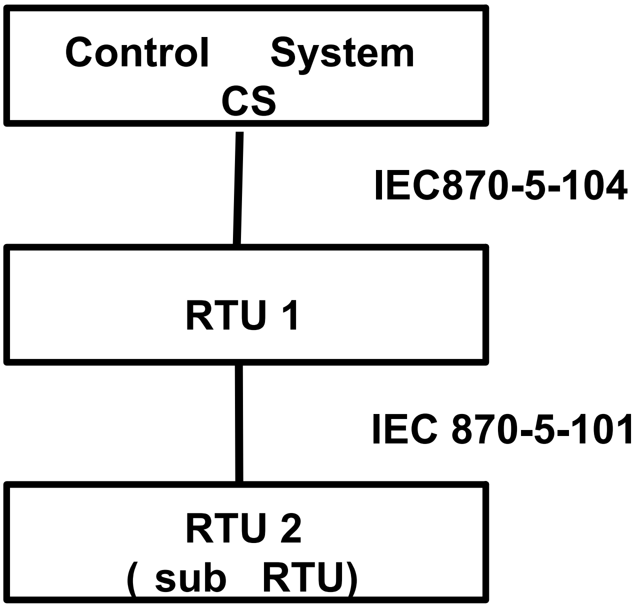


Figure 33: Network Topology with CS - RTU1 - SubRTU2

Excel files: two files - RTU1 and RTU2

RTU 1: one sheet

- Sheet 1:
  - IEC 870-5-104 block
  - Local I/O data point parameter block (RTU)

RTU 2: one sheet

- Sheet 1:
  - IEC 870-5-104 block
  - IEC 870-5-101 block
  - Local I/O data point parameter block (RTU)

Example 2: CS - RTU - IED

The following network topology needs one Excel file:

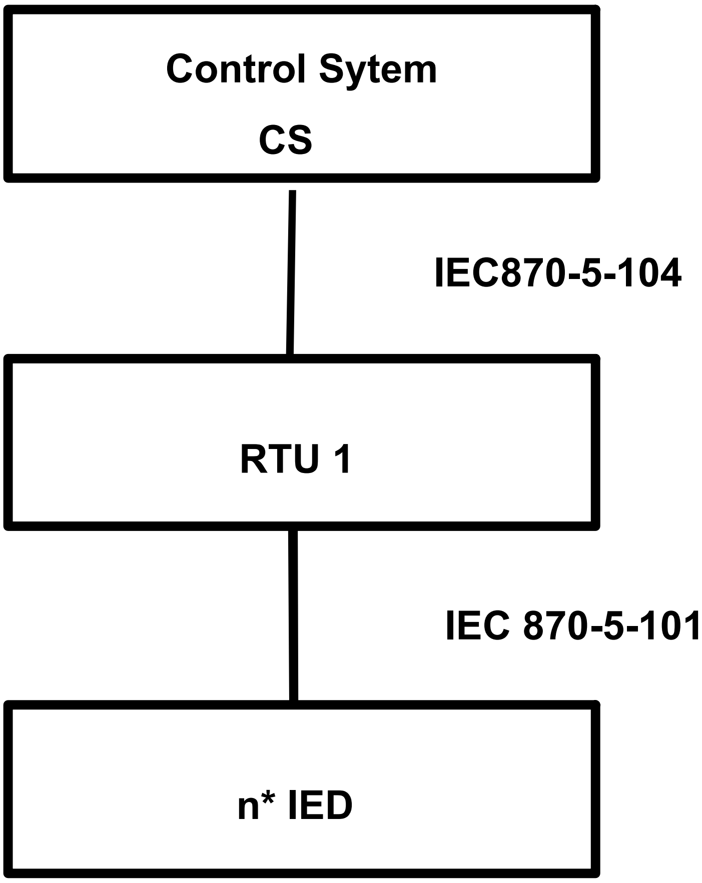


Figure 34: Network Topology with CS - RTU1 - n*IED

Excel files: one file – RTU 1

RTU 1: two sheets

- **Sheet 1:** with the RTU 1 local I/O data points
  - IEC 870-5-104 block
  - local I/O data point parameter block (RTU)

- **Sheet 2:** with the IED data points
  - IEC 870-5-104 block
  - IEC 870-5-101 block
  - IED data point parameter block (IEC 870-5-101)

Example 3: CS1 - CS2 - RTU - Sub-RTU - IED’s

The following network topology needs two Excel files:

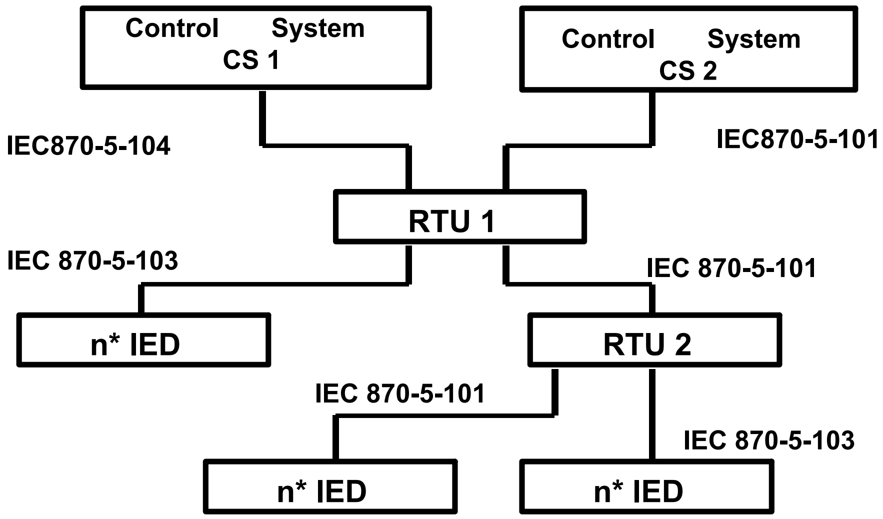


Figure 35: Network Topology with CS1, CS2 - RTU 1 - RTU2 - n*IED

Excel files: two files – RTU 1, RTU 2

RTU 1: two sheets

- Sheet 1: with the RTU 1 local I/O data points
  - IEC 870-5-104 block (CS 1)
  - IEC 870-5-101 block (CS 2)
  - local I/O data point parameter block (RTU)

- Sheet 2: with the IED data points
  - IEC 870-5-104 block (CS 1)
  - IEC 870-5-101 block (CS 2)
  - IEC 870-5-103 block (RTU 1)
  - IED data point parameter block (IEC 870-5-103)

RTU 2: three sheets

- Sheet 1: with the RTU 2 local I/O data points
  - IEC 870-5-104 block (CS 1)
  - IEC 870-5-101 block (CS 2)
  - IEC 870-5-101 block (RTU 1)
  - local I/O data point parameter block (RTU)

- Sheet 2: with IED data points from the IED’s connected to the IEC 870-5-101 line
  - IEC 870-5-104 block (CS 1)
  - IEC 870-5-101 block (CS 2)
  - IEC 870-5-101 block (RTU 1)
  - IEC 870-5-101 block (RTU 2)
  - IED Data point parameter block (IEC 870-5-101)

- Sheet 3: with IED data points from the IED’s connected with IEC 870-5-103 line
  - IEC 870-5-104 block (CS 1)
  - IEC 870-5-101 block (CS 2)
  - IEC 870-5-101 block (RTU 1)
  - IEC 870-5-103 block (RTU 2)
  - IED Data point parameter block (IEC 870-5-103)

11.4.3 Excel Sheet Types

To import data definitions related to an RTU's local I/O or related to subordinated IED lines, two types of Excel sheet column structures are used.

Excel Sheet for RTUs (Local I/O)

Column blocks:

- Signal (Signal type, System data type and RTUtil500 import flag)
- Process Object Identification
- RTU hardware address
- Address/host parameter blocks
- Data point parameter block (type specific)

Excel Sheet for Subordinated IED Lines

The IED data point parameter blocks depend on the protocol of the subordinated line the IED is connected to. Each protocol type has an individual data point parameter block structure.

The subordinated line sheet has no hardware address block. It consists of the following column blocks:

- Signal (Signal type, System data type and RTUtil500 import flag)
- Process Object Identification
- IED-Name
- Address/host parameter blocks
- IED line address/sub parameter block

11.4.4 General Hints for Columns and Rows

Unique Column Name

Each column from which data has to be imported to an RTUtil500 project file must have an unique identifier. All identifiers of the sheet must be in row number 5. With this identifier the import is managed.

ADVICE
It is not allowed to change the identifier names.

Rows per Data Point

The number of rows provided for each local process I/O signal can depend on the signal type and accords with the number of hardware I/O terminals occupied by the signal. The number of rows for each signal may be adapted to project specific requirements in the excel sheet (1 to n rows). It must be guaranteed that only one of the rows belonging to the same signal is allowed to get the 'Y' flag set in the STIM column which flags the signals as to be imported in the RTUtil500 project.

11.4.5 Excel Sheet Functions

The Excel sheet has some functions that facilitate working with the Excel sheet.

Drop-Down List of Predefined Values

Cells with limited entries provide a drop-down list.

Click on the arrow and then select an entry from the list.

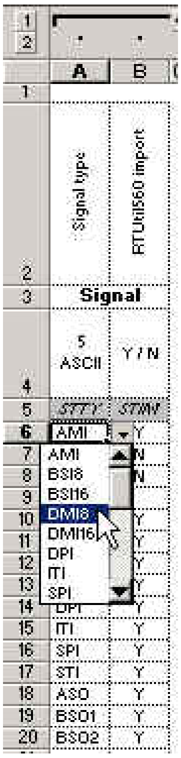


Figure 36: Excel cell drop-down list

Range Supervision

For cells of category number minimum and maximum limits are specified to prevent users from entering invalid data.

If invalid data is entered into the cell, an error messaged is displayed.

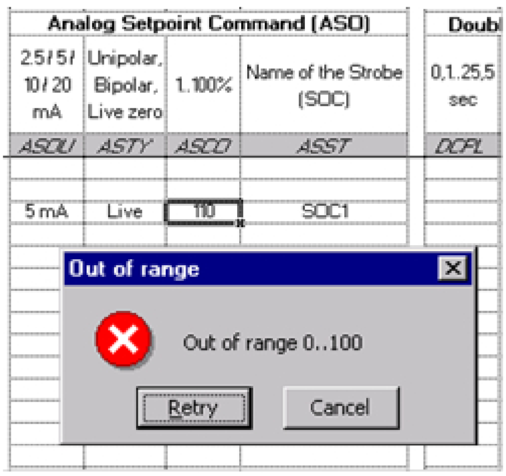


Figure 37: Excel cell range supervision

11.4.6 Excel Sheet Structure

Blocks of the Excel Sheet

The information of the sheets is arranged in logical blocks:

- Signal
- Process Object Identification
- RTU Hardware Address or IED name
- Address/Host parameter blocks
- Address/Sub parameter blocks (only in subordinated line sheets)
- Data Point Parameter blocks (only in the local I/O sheet)

The address/host parameter and address/sub parameter block is specific for each protocol type. The data point parameter block is point type specific.

Excel Sheet Block for Signal Types

Specifies the data point type and a flag which marks the data point for import to RTUtil500.

In case of data point type 'SEV' or 'SSC' the system data type have to be specified also.

Signal type	System data type	Type variant (AMI and EPI)	Process information (DCO and SCO)	RTUtil500 import
Signal				
5 ASCII	4 ASCII	ASCII	ASCII	Y/N
STTY	*STDT*	*STTV*	*STPI*	*STIM*
AMI		Standard		Y
DCO				Y
DPI				Y
DCO				Y
DPI				Y
DCO				Y
DPI				Y
DCO				Y
DPI				Y
DCO				Y
DPI				Y
SEV	#016			Y
SEV	#017			Y

Figure 38: Signal Block

Excel Sheet Block for Process Object Identification

The process object identification specifies the signal name of the data point. The length of the name is limited to 32 (128) characters. The structure of the signal name can be defined individually per project.

Station	Subnet	Bay	SCADA object
Process Object Identification			
8 ASCII	8 ASCII	8 ASCII	8 ASCII
OI01	OI02	OI03	OI04
RTUNAME	LADENBG	TEST007	AMI

Figure 39: Example of a Process Object Identification

In the example the name is structured in four elements (Station/Subnet/Bay/SCADA object) with a maximum length of eight characters for each of them. The name of the data point shown in the figure above that is imported to the RTUtil500 signal tree is “RTUNAME LADENBG TEST007 AMI“:

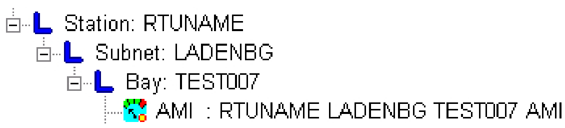


Figure 40: Example of a Process Object in a Signal Tree

A correct import of structured signal names to the RTUtil500 signal tree is achieved only if the signal tree structure elements defined after opening a new RTUtil500 project have exactly the same text length as the process object identification elements in the Excel import sheets.

ADVICE
It is strictly recommended that the object identification structure used in the Excel sheet is exactly the same as the signal tree structure defined for the RTUtil500 project file.

Excel Sheet Block for RTU Hardware Address

The hardware address of a local I/O data point is specified in the RTU hardware address block.

The hardware address consists of:

- I/O bus segment
- Rack address
- Slot address
- Channel within board

In addition, the data object qualifier "Blocked" is specified in the hardware address block.

I/O bus segment	Rack address	Slot address	Channel within board	Blocked
RTU Hardware Address				
1 .. 32	1 .. 7	1 .. 19	1 .. 64	Y / N
RUPB	*RURC*	*RUSL*	*RUCH*	*RUBL*
1	1	19	1	N

Figure 41: RTU Hardware Address

Excel Sheet Block for Line Address and Host Parameters

For each connected host line a protocol specific address/host parameter block is added to the Excel worksheet.

Following screenshot shows an example of an IEC 870-5-104 address/host parameter block. In this example, the rows contain the standard message address elements in a 1:1 manner – i.e. the input has to take place as one number for each standard address element.

ASDU address	Information object	Priority	Transmit with timestamp	In use	Spontaneous transmission (AMI, ITI, MFI)	Background cycle (AMI, MFI)	Transmission format (AMI, ASO, DSO)	Maximum value (100%) (AMI, ASO, DSO)	Minimum value (-100%) (AMI)	Interrogation group number (AMI, BSI, DMI, DPI, EPI, MFI, SEV, SPI, STI)	Include in sequence of event file (AMI, BSI, DMI, DPI, EPI, ITI, MFI, SEV, SPI, STI)
C 870-5-104-Host Addre		**IEC 870-5-104-Host Parameter**									
0 .. 65535	0 .. 16777215	1 / 2	Y / N	Y / N	Y / N	Off, 2, 4, 8, 10, 16, 30, 60 sec, 2, 5, 10 min	Normalized, Scaled	-32768 .. 32767	-32768 .. 32767	1 .. 16	Y / N
E301	*E302*	*E3PR*	*E3TI*	*E3IU*	*E3SP*	*E3BG*	*E3TF*	*E3MV*	*E3MNV*	*E3IGN*	*E3SEF*
90	3001	2	Y	Y	Y	Off	Normalized	1	-1		N

Figure 42: Example of an IEC 870-5-104 address/host parameter block

If the message address elements of communication protocols shall be engineered in a structured manner (possible with the IEC 60870-5-101 and –104 protocols), the structure elements and bit sizes should be defined in an early stage of a customer project, i.e. before creating the project pattern and the Excel sheets. The standard protocol message address elements may be structured in a free manner on bit limits. This means, that for example a 16 bit ASDU Address may be structured in two elements, occupying 7 and 9 bits. The address elements are then input as two numbers with value ranges of 0...127 respective 0...1023 (see following example).

Note, that restrictions in the possible ranges as with following example (e.g. 1...127) are not checked when the Excel import is performed. If those restrictions are specified in a customer project, the user has to take care for correct data entry.

ASDU address	ASDU address	Information object	Information object	Information object	Information object	Priority
IEC 870-5-104-Host Address						
1 .. 63	1 .. 1022	0 .. 7	1 .. 127	0 .. 63	0 .. 254	1 / 2
E3011	*E3012*	*E3021*	*E3022*	*E3023*	*E3024*	*E3PR*
2	21	5	26	7	3	2

Figure 43: Structured IEC 870-5-104 addresses

The Excel sheet has as many address/host parameter blocks as corresponds to the number of communication lines a message passes from the lowest level RTU up to a top level end node of the network tree (normally a Network Control System).

Excel Sheet Block for local Data Point Parameters

Each data point type has its own specific parameter block.

The following data point types are available in monitoring direction:

- Analog Measured value Input (AMI)
- Analog Measured value Floating input (MFI)
- Bit String Input (BSI8/BSI16)
- Digital Measured value Input (DMI8/DMI16)
- Double Point Input (DPI)
- Integrated Totals Input (ITI)
- Single Point Input (SPI)
- Step Position Input (STI)

Data points types in command direction:

- Analog Setpoint Output (ASO)
- Bit String Output (BSO1/BSO2/BSO8/BSO16)
- Digital Setpoint Output (DSO8/DSO16)
- Double Command Output (DCO)
- Floating Setpoint Command Output (FSO)
- Regulation step Command Output (RCO)
- Single Command Output (SCO)
- Strobe Output Channel (SOC)

As an example, the parameter block for analog measured value inputs (AMI) is shown in following section of an Excel sheet.

Input signal range	Input signal type	Live zero adjust (only live zero)	Conversion factor	Line frequency	Smoothing	Acquisition mode	Periodic update	Periodic update unit	Threshold	Zero range	Fast input scanning
Analog Measured value Input (AMI)											
2mA .. 10V	Unipolar, Bipolar, Live zero	Y / N	1 .. 100 (%)	16 2/3, 50, 60 Hz	0 .. 128	4 modes possible	1, 2, 4, 8, 30, 60 sec, 100, 200 mSec	Sec / mSec	0 .. 12 %	0.1 .. 5.0 %	Y / N
AMRA	*AMTY*	*AMLZA*	*AMCO*	*AMFR*	*AMSM*	*AMTR*	*AMSC*	*AMSCU*	*AMTH*	*AMZR*	*AMFIS*
5 mA	Bipolar	N	100	Off	200	0		Sec	12	0,5	N

Figure 44: Analog Measured Value Input (AMI) Parameter

11.5 Performing the Excel Import

The Excel import is started from the RTUtil500 **Extra** – **Excel Import** menu. A wizard guides through the dialogs and asks for the necessary settings. The wizard is described within chapter "Data Interface – Excel Import" of this document.

11.5.1 Excel Import Error Handling

During the Excel import a log file is created. This file gets automatically the same name as the imported Excel file, but file extensions '.log'. Errors and warnings detected during the Excel import are written to the log-file.

In addition error messages are displayed in Excel import wizard step 3:

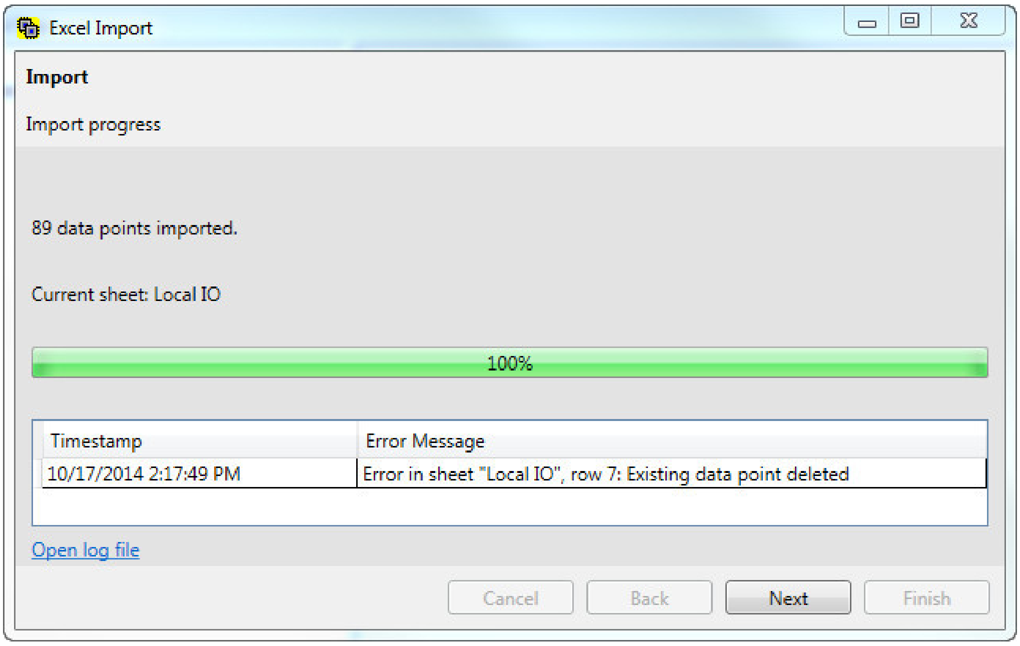


Figure 45: Excel import wizard step 3 with error message

12 CSV Interface

12.1 Introduction

The CSV interface allows importing data from a network control system into the RTUtil500 engineering tool. The data points and their properties are exported from the network control system in CSV format. The CSV data is imported via the RTUtil500 into an Excel sheet where additional information can be added. The last step is to run the Excel import to get the data into the tool.

The information goes into one direction: from the network control system to the RTU engineering tool RTUtil500.

These chapters describe

- how to use the 'CSV interface initialization wizard'
- steps 'Performing the CSV import'

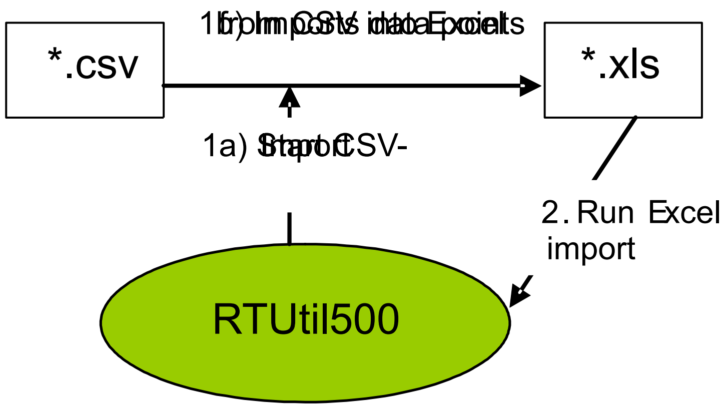


Figure 46: Workflow of CSV Interface

The following CSV records can be imported:

- VER - Version, (not imported, information only)
- RTU - RTU record, (not imported, information only)
- SEV - Systemevent
- FTR - Filetransfer

and IO data:

- AMI
- ASO
- BSI08, BSI16, BSI32
- BSO1, BSO2, BSO8, BSO16
- DCO
- DMI8, DMI16
- DPI
- DSO8, DSO16
- EPI
- FSO

- ITI
- MFI
- RCO
- SCO
- SPI
- SSC

12.2 CSV interface initialization wizard

With the CSV interface initialization wizard the CSV interface is adapted to the CSV file contents. The interface must be initialized for every host line for which data has to be imported.

12.2.1 Starting the wizard

Click in RTUtil500's menu bar on – **Project** - **Settings**

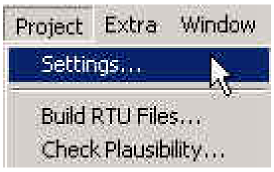


Figure 47: Menu Project settings

Choose the folder 'Initialize CSV Interface' in the project settings window:

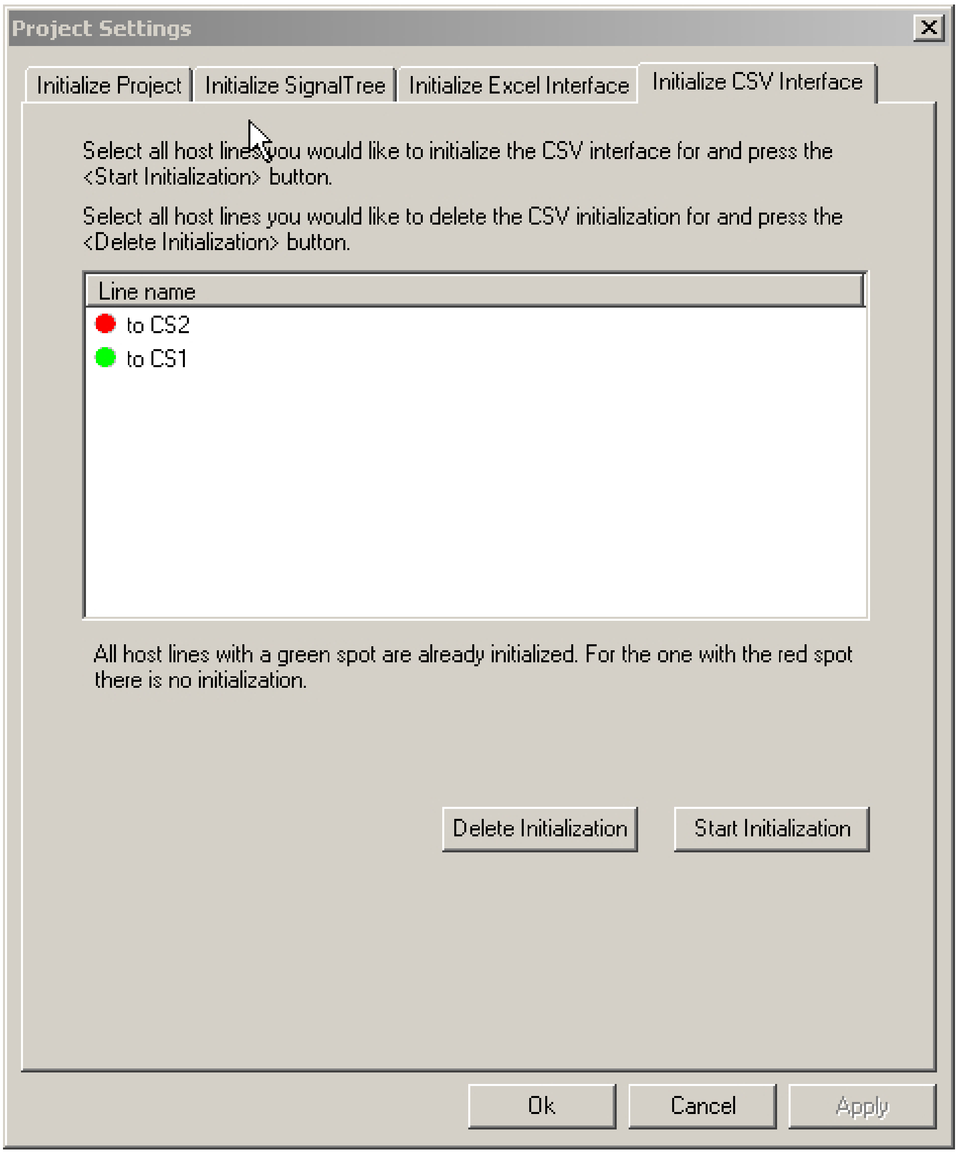


Figure 48: Project settings initialize CSV Interface

The status of the active RTUtil500 project's CSV interface initialization is shown.

In the dialog all host lines of the project are shown with a colored spot in front of them. A red spot identifies that the CSV interface is not yet initialized for the line. A green spot indicates that the CSV interface is initialized.

ADVICE
If a CSV interface is initialized for a host line the spot will always remain green. It will not change to red anymore even if the CSV file for this line is modified or changed.

Select the host line for which the CSV interface shall be initialized. The selection of multiple host lines is possible using the Windows standard mechanisms for multi selection (Shift resp. Ctrl key pressed + left Mouse click).

With the 'Start Initialization' button the initialization wizard of the CSV interface is started for the first of the selected host lines.

12.2.2 Entering data types

The standard data types which can be imported are listed here. The data type names have to be entered as they are used in the CSV file next to the standard data type. The names can be similar. If some of the predefined data types are not used in the import file, the colums "type Name" has to be left empty.

ADVICE
The data types 'RTU' and 'VER' must be set. Otherwise the CSV import will fail.

Select the CSV separator; default value is semicolon.

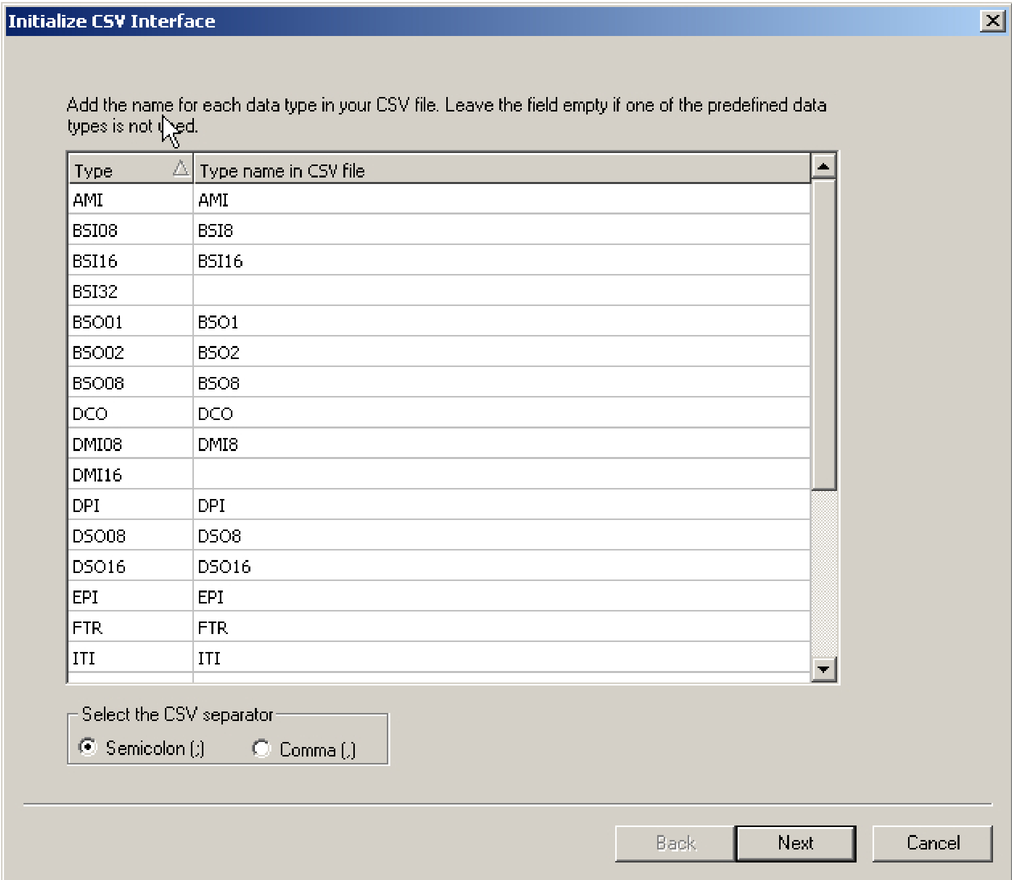


Figure 49: Enter data types

Press 'Next' to continue with the wizard and 'Cancel' to cancel.

12.2.3 Enter the positions of the parameters

For each used data type entered in the previous step this page is displayed.

The importable parameters are listed in the corresponding column. The parameter list depends on the data type and the type of the selected host line.

The position of the parameter in the CSV file has to be entered in 'Column number'. If a parameter is not used, the field for the column number has to remain empty.

'Column number' 1 is fix for the record type ('AMI', …) and can not be changed.

Sorting of the table is possible by selecting the 'Column number' header cell.

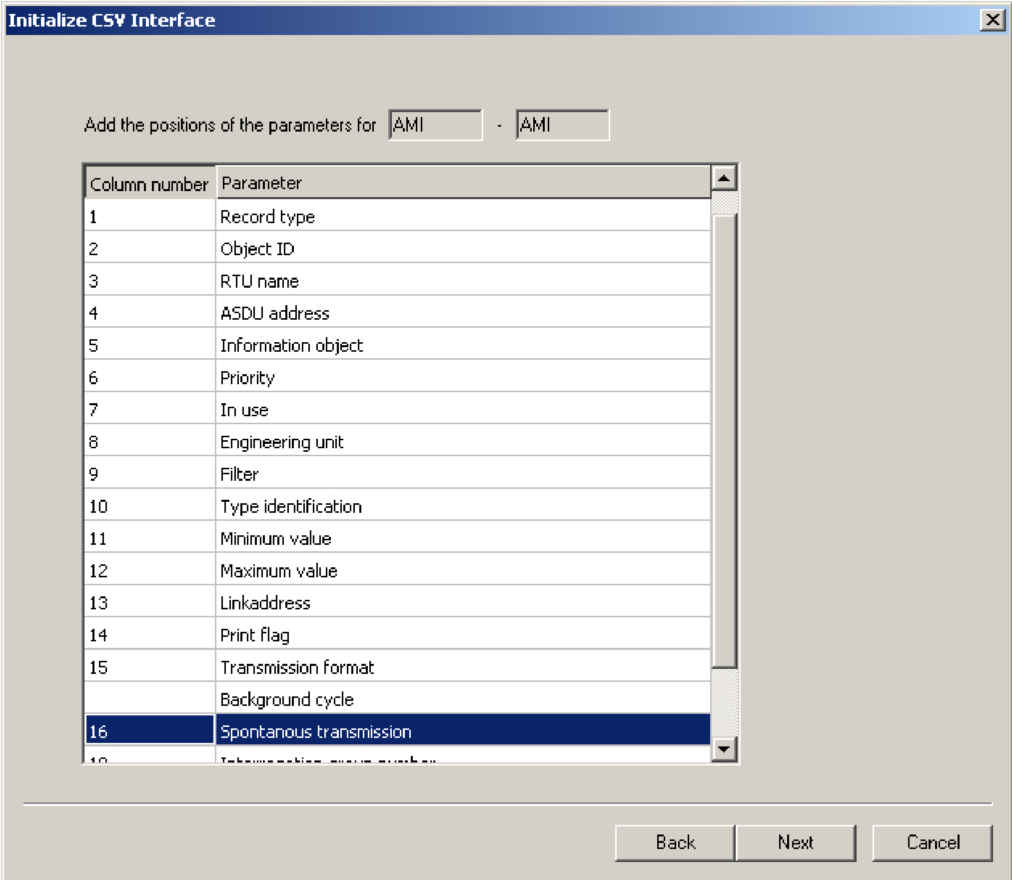


Figure 50: Enter positions

12.2.4 Finalizing the wizard

To complete the initialization press ‘Finish’. If the initialization was successful, the dot in front of the initialized host line should be green now. The settings are not stored yet. To store the settings for the CSV interface in the rtu file, the project has to be saved.

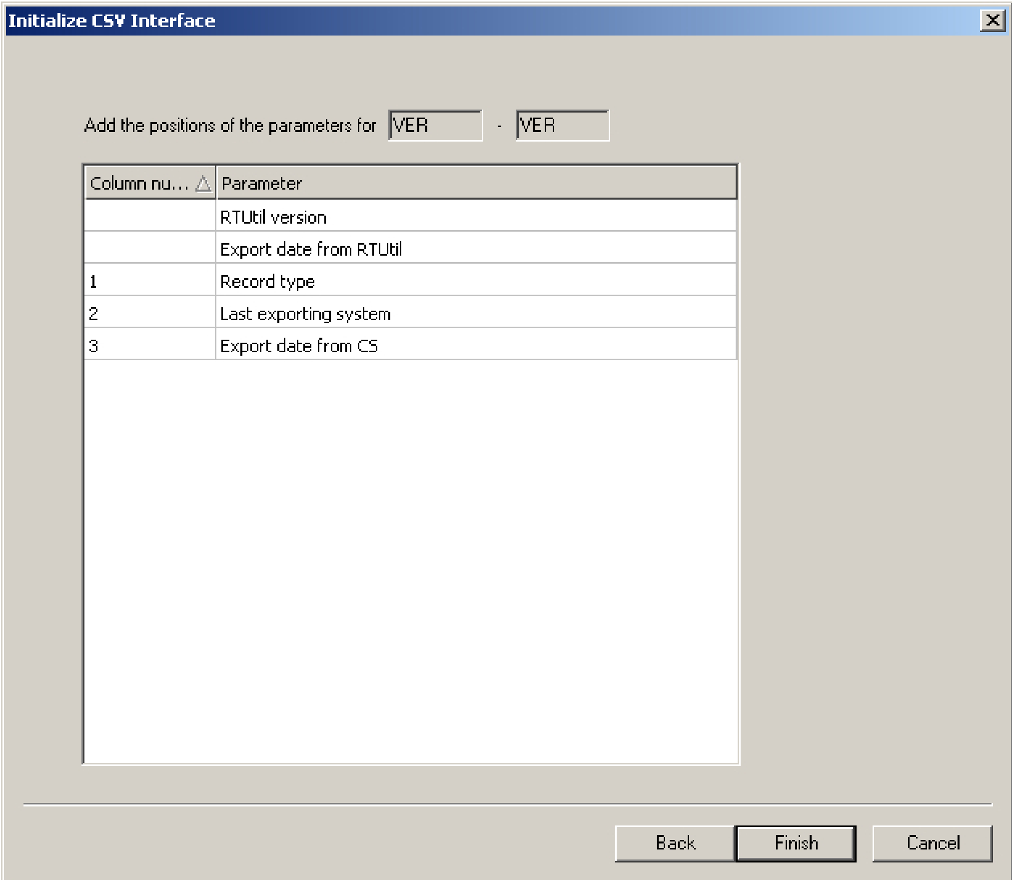


Figure 51: Last wizard page

12.3 Performing the CSV import

As a precondition both the CSV and the Excel interface has to be initialized as the CSV data is imported into an Excel file.

12.3.1 Start the CSV import

Start the CSV import:

- Menu: Extra/CSV Import …
- Toolbar: Click 

12.3.2 Select the line

Select the host line for which data has to be imported.

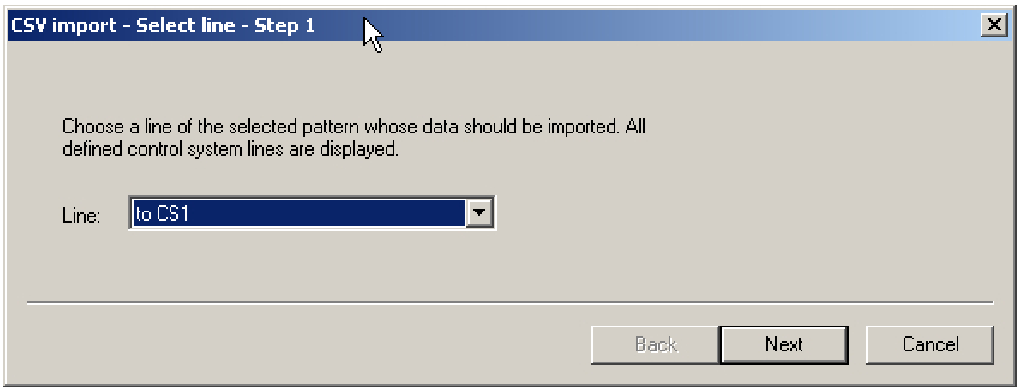


Figure 52: Choose host line

12.3.3 Select the CSV file

Select the CSV file you want to import.

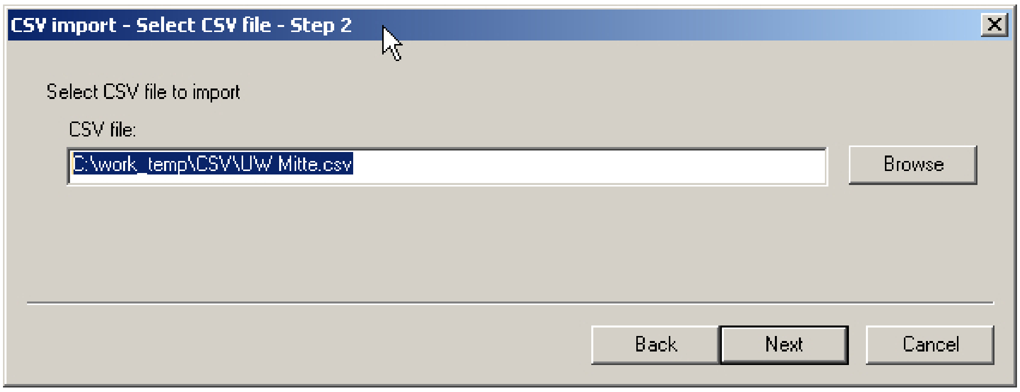


Figure 53: Select the CSV file

12.3.4 Select the Excel file and start importing

Select the Excel file into which the CSV data is imported.

The structure of the excel file must fit to the project.

ADVICE
The excel file must contain a column called 'Modified' and a ‘Filetype’ column. See chapter 'Check the CSV import result'.

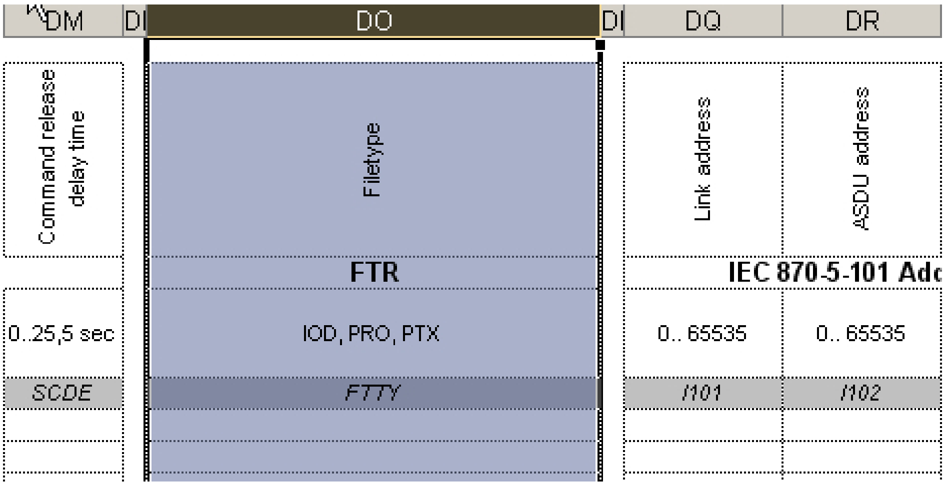


Figure 54: AddFfile Type column

The 'Modified' column contains information about the status of the record, if it is new, updated, deleted.

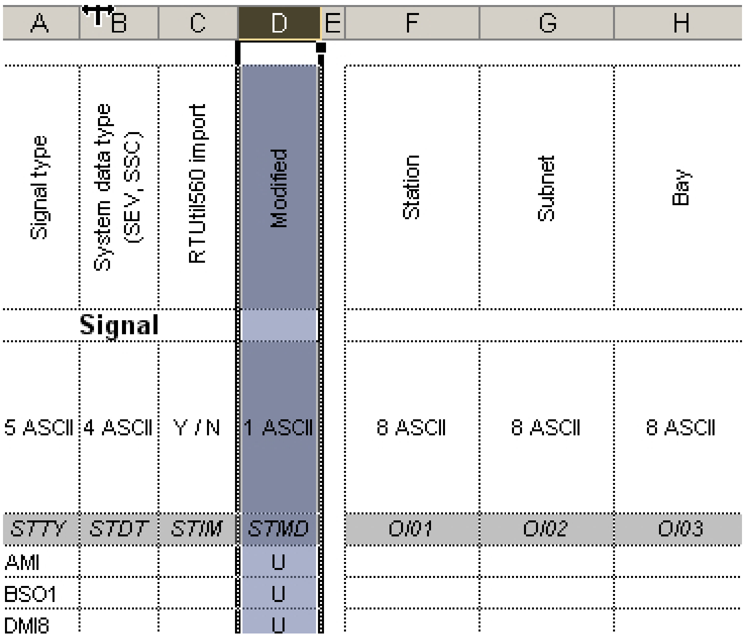


Figure 55: Add 'Modified' column

Only the properties set in the CSV file and the 'Modified' column will be written/overwritten in this Excel file.

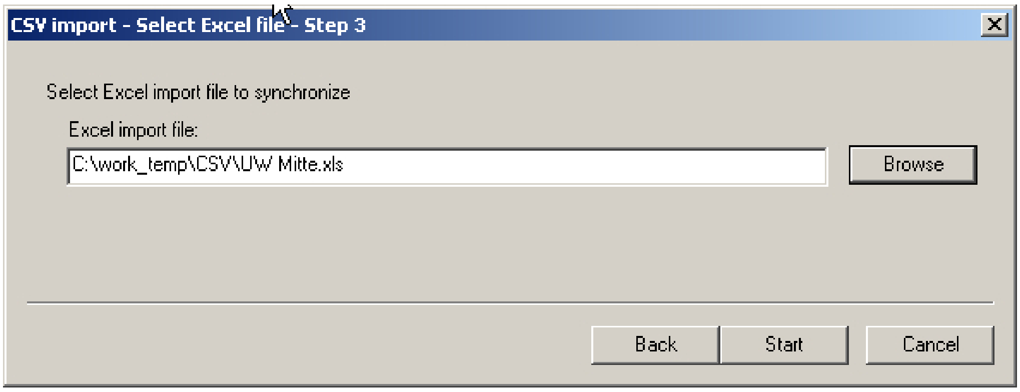


Figure 56: Select Excel file

ADVICE
Make sure that the CSV file and the Excel file are not opened in Microsoft Excel. Otherwise the import process will not run.

Select 'Start' to start the CSV import.

12.3.5 Check the CSV import result

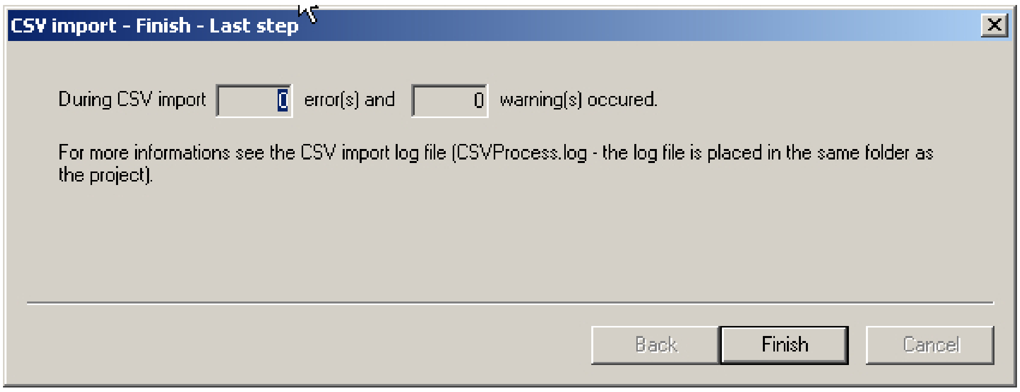


Figure 57: Import finished

The last page displays the error and warning counters. Select 'Finish' to close the wizard.

Check the log file 'CSVProcess.log' for information about the errors and warnings occurred during the import. The log file is placed in the same folder as the project (*.rtu). The following errors may occur:

Error	Description
100	CSV interface settings are not available. Run the CSV interface initialization, see chapter '"CSV interface initialization wizard"'
101	The imported data point is unknown.
102	Could not open CSV file. Check if the file exists and is not opened yet.
103	Error reading CSV file datasets

Error	Description
1000	The excel file cannot be opened. Close the file and try again.
1001	The excel file is read only. Deselect the read only attribute of the excel file
1002	Problem closing the Excel file.
1003	The sheet names of the Excel file cannot be read. Run the excel initialization again selecting the correct sheet names.
1004	The import sheet of the Excel file is missing. Run the Excel export.
1005	Error while reading the mapping. Run the Excel export.
1006	Error opening the database.
1007	Reading the initialization failed. Run the Excel export.
1008	Column 'Modified' not found. Add column 'Modified' to column 'D' of the Excel file.
1009	Column 'FTR' not found. Add column 'FTR' to the excel file.
1010	An excel column was not found. Run the Excel export.
1011	The excel sheet map was not found. Run the Excel export.
1012	Error getting data points from Excel file.
1013	Error applying data points to Excel file.
1014	Error updating data points

Check the Excel file into which the CSV data is imported. The 'Modified' column contains the status information for the data point:

- 'N' for New record: This data point was not part of the Excel file before.
- 'U' for Updated record: This data point has been in the Excel file before, all CSV properties are overwritten with the values of the CSV file.
  There is no check whether the CSV properties themselves have changed.
- ' D' for Deleted record: This data point is no longer contained in the CSV file.

Finally the Excel file can be imported into RTUtil500. The flags NEW, UPDATED and DELETED are not read by the Excel import. Rows that should not be imported, have to be deleted manually.

See chapter '"Excel Interface" for detailed Excel import description.

13 IEC61850 Engineering

13.1 RTU500 series in an IEC61850 System

The RTU provides as IEC61850 client NCC gateway functionality, connecting an IEC61850 station bus with NCC's. As IEC61850 server the RTU operates as IEC61850 IED providing data to an IEC61850 network from subordinated devices or direct connected signals. The figure below shows the RTU in an IEC61850 system.

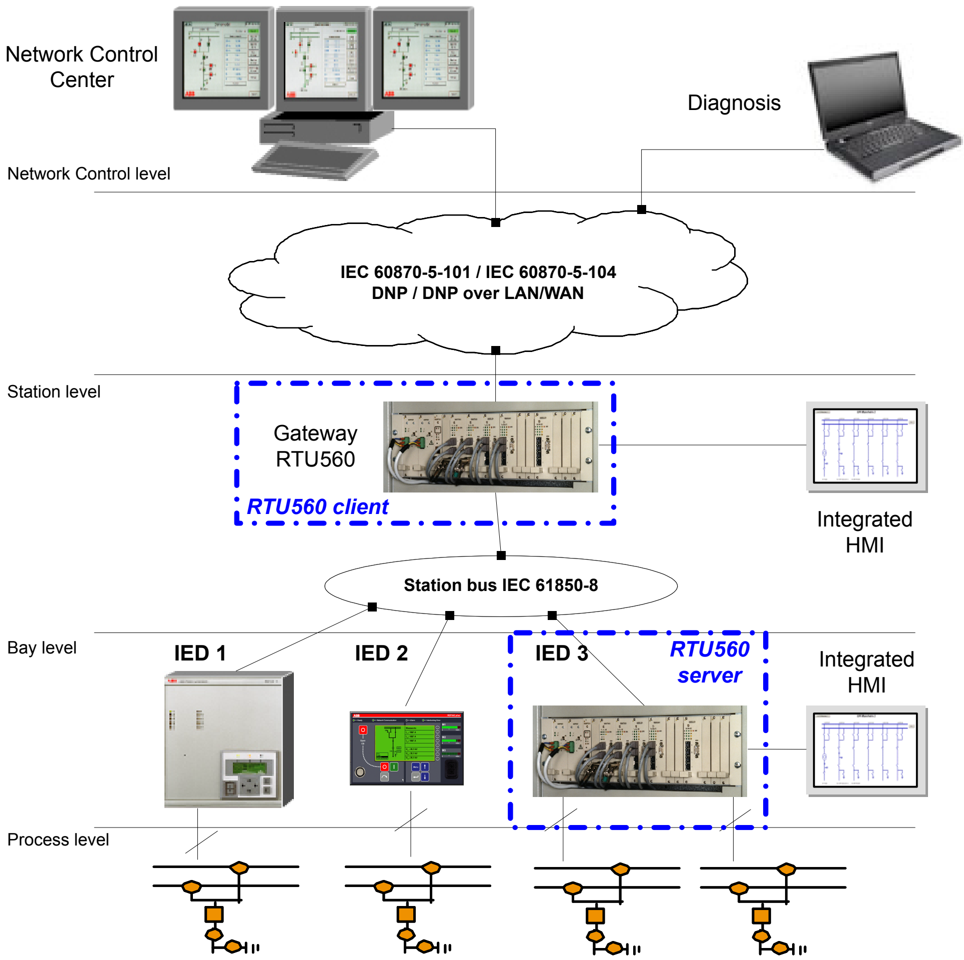


Figure 58: RTU in an IEC61850 system

The standard functionality of the RTU500 series like local I/Os and connections with legacy protocols are available in the client and the server configuration as well.

13.2 IEC61850 configurations

An RTU could be configured in RTUtil500 as IEC61850 client IED or IEC61850 server IED. The different IED types must be configured in separate projects. It is not possible to configure a whole IEC61850 network with several RTU clients or servers within one project. The following chapters show examples for RTU client and server configurations.

13.2.1 RTU500 series as IEC61850 client

As IEC61850 client the RTU connects NCC's with an IEC61850 network. Additional local I/Os and connections with legacy protocols are possible. In this configuration the RTU doesn't support any GOOSE communication. The following figure shows an RTU configured as IEC61850 client.

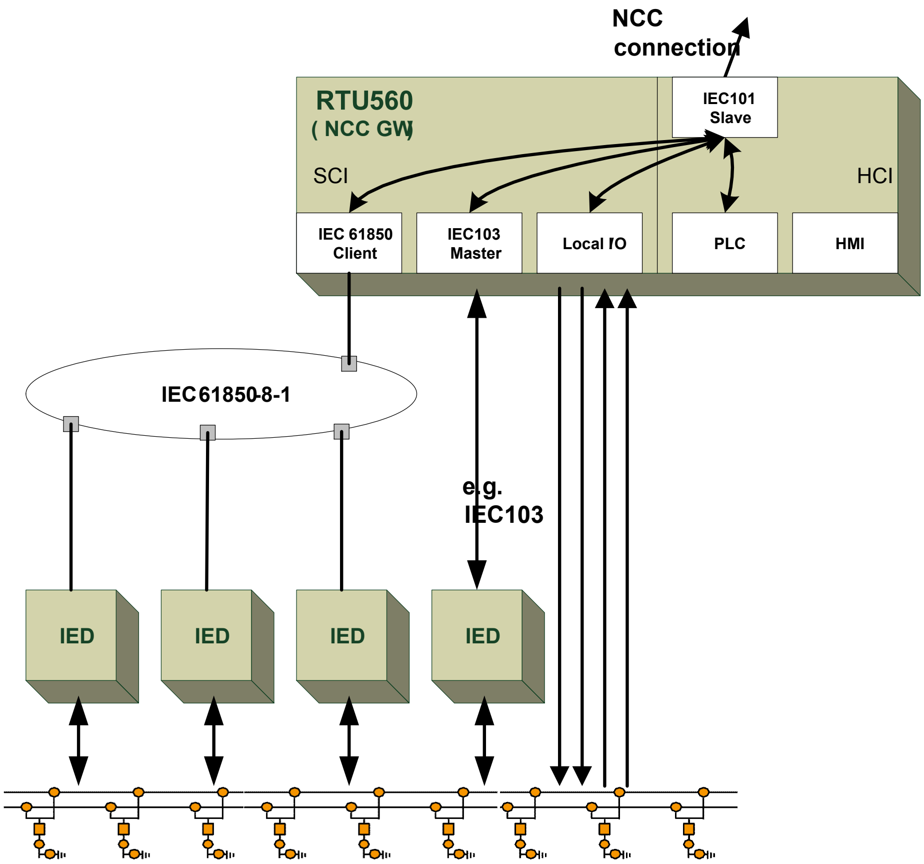


Figure 59: RTU as IEC61850 client

13.2.2 RTU500 series as IEC61850 server

As IEC61850 server the RTU provides data to an IEC61850 network. The data could be provided from IED's connected via legacy protocols, local I/Os or PLC applications.

ADVICE
In this configuration the RTU supports horizontal GOOSE communication with other IEC61850 IED's as well. The GOOSE data received from the IEC61850 network could be used in a PLC application only.

The following figure shows an RTU configured as IEC61850 server.

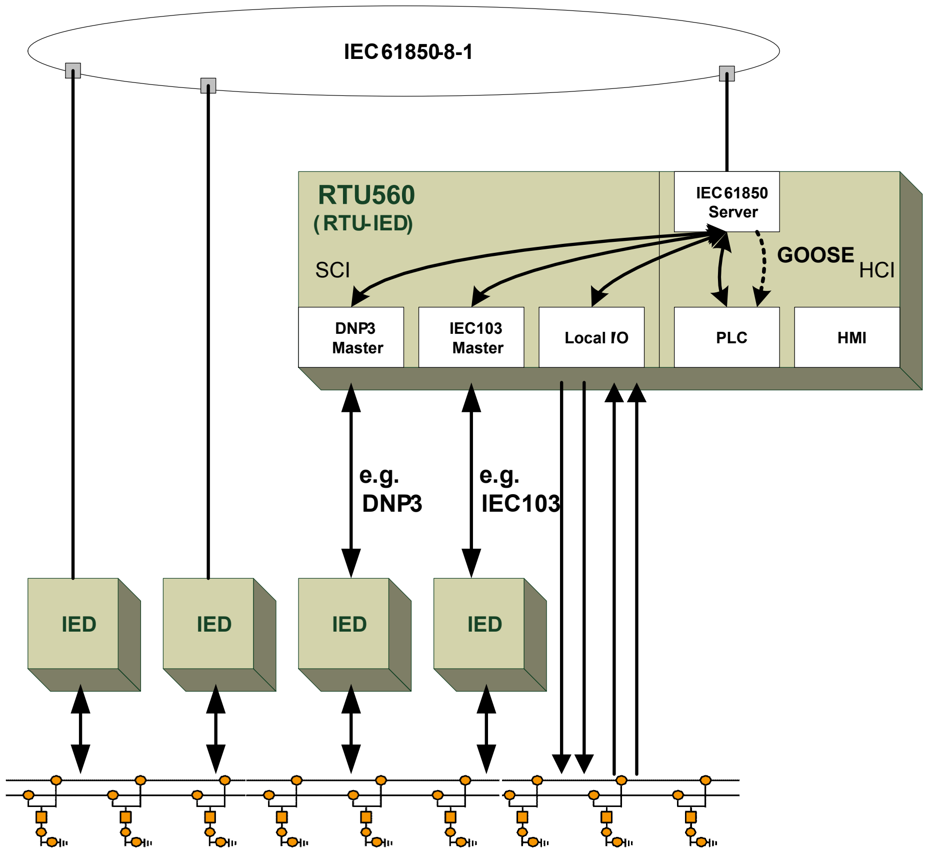


Figure 60: RTU as IEC61850 server

13.3 IEC61850 engineering process overview

The engineering process of an IEC61850 system is defined as top-down engineering in the IEC standard. In this process RTUtil500 is the IED configuration tool for the RTU. In case of the RTU the engineering process must be distinguished between IEC68150 client and IEC61850 server configurations. The following chapters provide an overview of the complete IEC61850 engineering process. The detailed engineering steps in RTUtil500 are described in further chapters.

13.3.1 IEC61850 client engineering

The following figure shows the different steps of the system engineering in an overview.

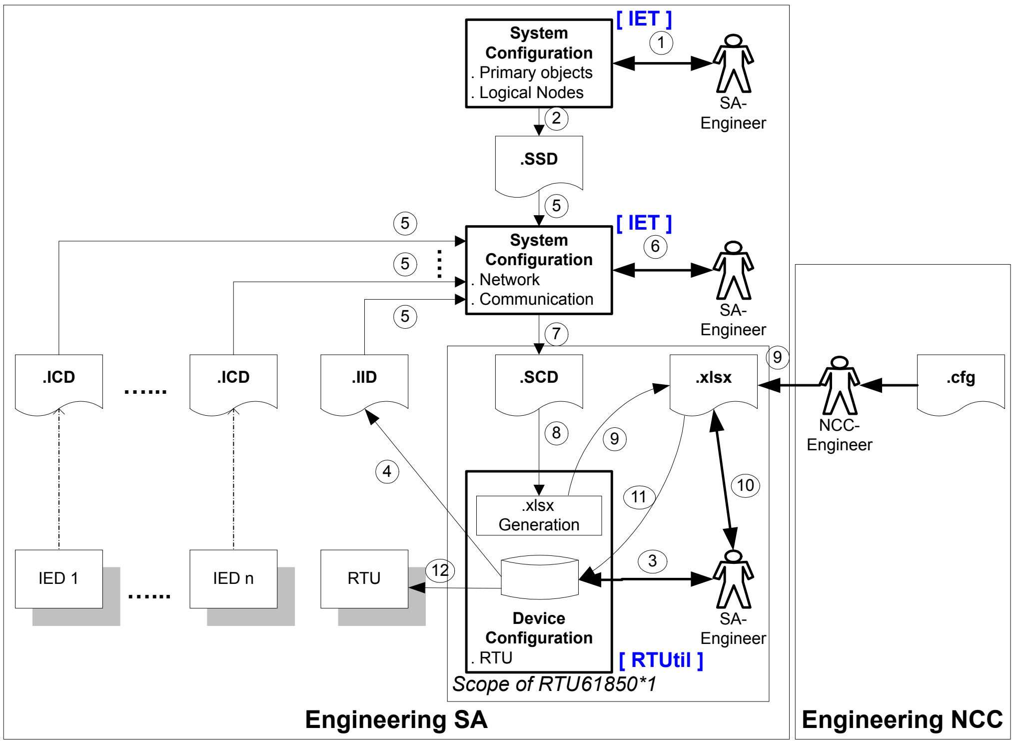


Figure 61: RTU client engineering process

The steps in detail are:

1 The Substation Automation (SA) engineer defines in the system configuration the primary objects, the structure of the substation and Logical Nodes of the substation.
2 The primary system configuration tool exports a System Specification Description (SSD) file including:
- primary objects
- substation structure
- Logical nodes
3 The SA engineer configures in RTUtil500 the network and hardware tree of the RTU. This includes the configuration for the IEC61850 client.
4 The IEC61850 model of the RTU client is exported in an IID file.
5 The communication system configuration tool imports following data.
- SSD-file of the primary system configuration
- ICD (IED Capability Description) files of the IED’s
- IID-file of the RTU (client)
6 The SA-Engineer configures the IEC61850 network, the relationship between Logical Nodes and IED’s and the communication between the IED’s and the gateway. This configuration is done in the communication system configuration tool.
7 It is not allowed to modify the RTU560 IED configuration during this step, i.e. appropriate nodes must not be added, changed or deleted.
8 The communication system configuration tool exports a System Configuration Description (SCD) file including:
- Primary objects
- Substation structure
- Logical nodes
- IEC61850 network
- Relationship between Logical Nodes and IED’s
- Communication between the IED’s and the gateway

9 RTUtil500 imports the SCD-file.
10 RTUtil500 extracts all available data points from the SCD-file. The data points are build from the IEC61850 data attributes. The following conditions are required for available status data attributes:
1. the data attribute is defined as a member of a data set.
2. A Report Control Block is defined for the data set.
3. The Report Control is directed to the RTU.
4. The server IED is assigned to the RTU in the network tree.
Controllable data attributes are extracted independent of the data set configuration. RTUtil500 writes these data points together with the substation structure information into the Excel import file.
11 The SA-Engineer or NCC-Engineer defines
- which data points have to be mapped to the internal data points of the RTU
- which data points has to be mapped to the NCC signal list
- additional information data like specific data point name
12 RTUtil500 imports the completed Excel import file. The configured IEC61850-data points (all or a subset) are mapped to internal data and to NCC-signal list as defined.
13 The data point configuration is generated by RTUtil500 and loaded to the RTU.

The primary system configuration tool and the communication system configuration tool are outside the scope of this document. These tools are not part of RTUtil500. RTUtil500 covers the steps 3, 4 and 8 to 12 only. The initial point for the engineering in RTUtil500 is the completed SCD file.

ADVICE
The ICD-file of the RTU used in former versions of RTUtil500 must not be used anymore. The ICD-file is replaced by step 4 with the generation of an individual IID file containing the IEC61850 model of the RTU client. The IID file is used for the IEC61850 system configuration.

The ICD file in the installation of RTUtil500 is for documentation purposes only. Do not use this file for the IEC61850 system configuration.

13.3.2 IEC61850 server engineering

The figure below shows the different steps of the system engineering in an overview.

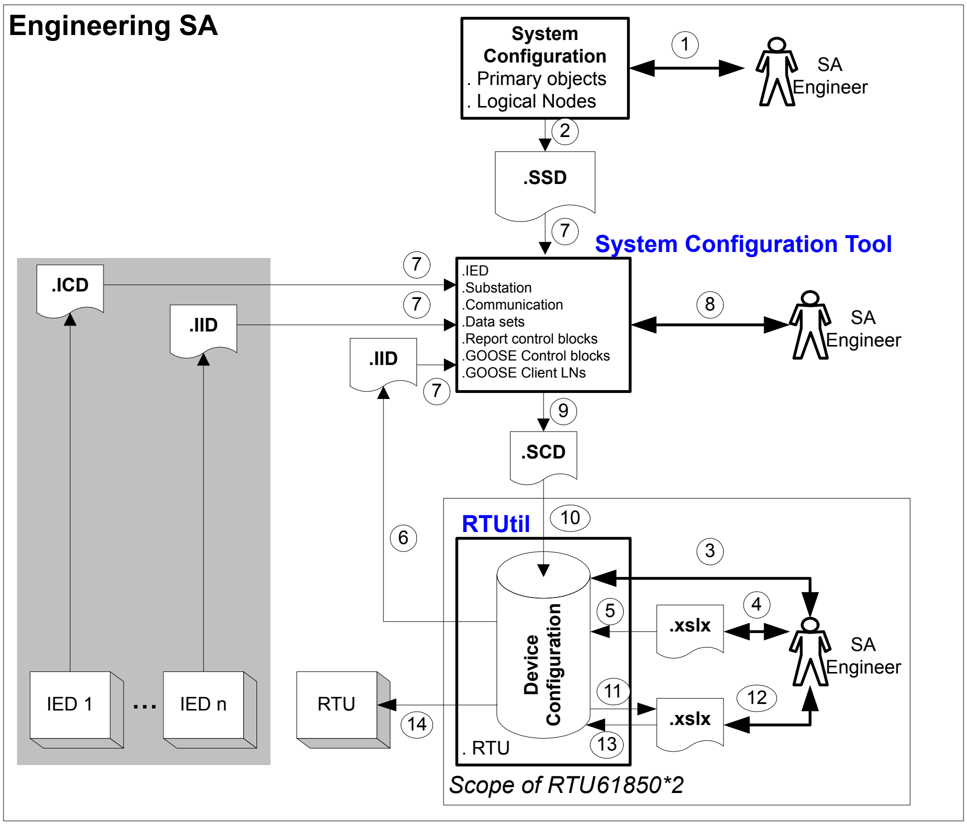


Figure 62: RTU server engineering process

The steps in detail are:

1 The Substation Automation (SA) engineer defines in the system configuration the primary objects, the structure of the substation and Logical Nodes of the substation.
2 The primary system configuration tool exports a System Specification Description (SSD) file including:
   - primary objects
   - substation structure
   - Logical nodes
3 The SA engineer configures in RTUtil500 the network and hardware tree of the RTU. This includes the configuration for the IEC61850 server. This template configuration of the RTU is exported to an Excel file.
4 In the Excel file the IEC61850 data model of the RTU server is configured. This includes:
   - Logical devices
   - Logical node instances
   - Data objects and attributes
   - MMS data sets
   - GOOSE data sets
   - Mapping to RTU data point types
5 RTUtil500 imports the Excel file with IEC61850 data model.
6 The IEC61850 model of the RTU server is exported in an IID file.
7 The communication system configuration tool imports the following data.
   - SSD-file of the primary system configuration
   - ICD (IED Capability Description) files of the IED’s
   - IID-file of the RTU (server)

8 The SA-Engineer configures the IEC61850 network, the relationship between Logical Nodes and IED's and the communication between the IED's and the gateway. This configuration is done in the communication system configuration tool.
9 It is not allowed to modify the RTU560 IED configuration during this step, i.e. appropriate nodes must not be added, changed or deleted.
10 The communication system configuration tool exports a System Configuration Description (SCD) file including:
- Primary objects
- Substation structure
- Logical nodes
- IEC61850 network
- Relationship between Logical Nodes and IED's
- Communication between the IED's and the gateway
11 RTUtil500 imports the SCD-file.
12 RTUtil500 extracts all available GOOSE data points send from other IED's from the SCD-file. The data points are build from the IEC61850 data attributes. For GOOSE data points only status data is possible. The following conditions are required for available status data attributes:
1. the data attribute is defined as a member of a GOOSE data set.
2. A GSE Control Block is defined for the data set.
3. The Control Block is directed to the RTU.
RTUtil500 writes these data points together with the substation structure information into the Excel import file.
13 The SA-Engineer or NCC-Engineer defines
- which GOOSE data points have to be mapped to internal data points of the RTU
- which GOOSE data points have to be processed in a PLC function
14 RTUtil500 imports the completed Excel import file (GOOSE sheet only). The configured GOOSE-data points (all or a subset) are mapped to internal data as defined.
15 The data point configuration is generated by RTUtil500 and loaded to the RTU.

The primary system configuration tool and the communication system configuration tool are outside the scope of this document. These tools are not part of RTUtil500. RTUtil500 covers the steps 3 to 6 and 10 to 14 only.

The installation of RTUtil500 contains an ICD file for the RTU server. This file is for documentation purposes only. Do not use this file for the IEC61850 system configuration.

13.3.3 RTUtil500 data model

There is vertical and as well horizontal communication on an IEC61850 system bus. The RTU has to support vertical communication as client and server. Furthermore the RTU server must provide horizontal communication. The horizontal communication is the GOOSE communication defined in IEC61850 (see IEC61850 standard for more information). To include the requirement for horizontal communication the RTUtil500 data model is extended for IEC61850 host communication interfaces (server), because in general host communication interface could not retrieve data from devices on the same level. The following figure shows the integration of the IEC61850 communication lines in the RTUtil500 data model.

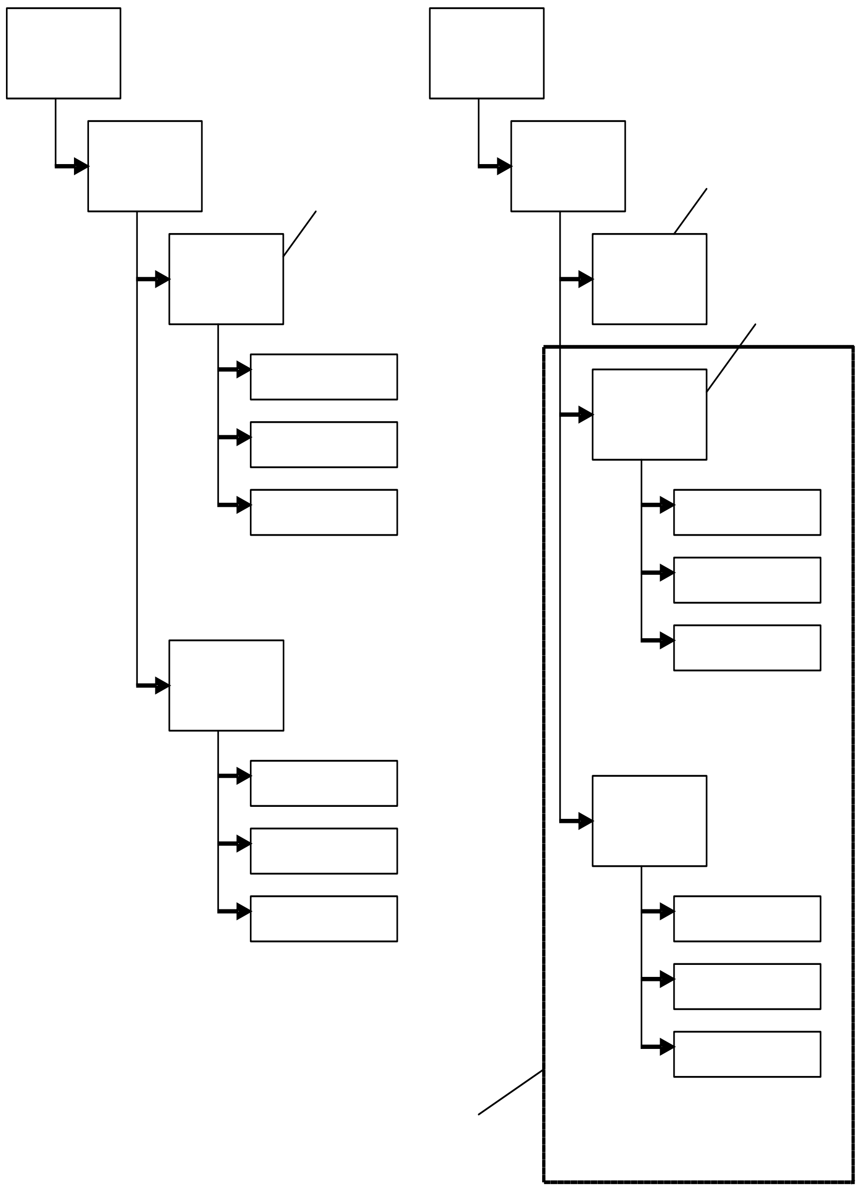


Figure 63: RTUtil500 data model

For RTU server it is possible to add IED's at an IEC61850 host communication line. These IED's represent the IEC61850 servers connected to the same station bus.

13.3.4 Horizontal GOOSE Communication

The RTU receives GOOSE data from other IED's on the same station bus. These data points are received in GOOSE data sets defined in the system engineering. The following restrictions apply to the GOOSE data points in RTUtil500:

- GOOSE data points could be monitoring data points only. Commands are not possible as GOOSE data points.
- GOOSE data points are available for PLC processing only. That means the data points could not be used in any other activities (e.g. it is not possible to send the data points to other host communication lines).
- GOOSE data points received by the RTU are configured in the system-engineering tool (via the GOOSE communication configuration). In RTUtil500 these data points are created with the SCD file import. It is not possible to create these data points in the GUI directly.
- The GOOSE data points send by the RTU are part of the data model in RTUtil500 and identified by GOOSE data set name set at the data point.

13.3.5 IEC61850 Excel Import Sheets

The Excel import files are organized as one sheet per subdevice communication line connected to a RTU. The host communication lines are represented by the protocol specific data point references in every sheet. This applies for IEC61850 lines as well. That means the RTU client is represented by an own Excel sheet and the RTU server by address references in every sheet.

Additionally, the GOOSE receive data points of the RTU server are included in a separate Excel sheet. This Excel sheet exists for each RTU server in the project and contains beside the IEC61850 address information the assignment to PLC functions only. The following overview summarizes the information sections included in the different Excel sheet types.

- IEC61850 sub device line sheet (RTU client)
  - Signal (with additional Modified column)
  - Process Object Identification
  - IED name
  - IEC61850 substation structure
  - Archive/Print
  - PLC function references
  - Host interface address and parameter (including RTU server)
  - IEC61850 data point address
  - IEC61850 data point parameter
- Other sub device line sheet
  - Signal
  - Process Object Identification
  - IED name
  - Archive/Print
  - PLC function references
  - Host interface address and parameter (including RTU server)
  - Sub device interface address and parameter
- GOOSE receive data sheet (RTU server)
  - Signal (with additional Modified column)
  - Process Object Identification
  - IED name
  - IEC61850 substation structure
  - PLC function references
  - IEC61850 data point address
  - IEC61850 data point parameter

The format of the different information sections with IEC61850 data are shown in the next chapters. Columns that should not be changed by the user are marked with a light grey background of the header cells.

13.4 Detailed RTUtil500 client engineering

13.4.1 Network and Hardware tree

The network defined in the SCD file must be mapped to an equivalent structure in the RTUtil500 network tree. The following figure shows the defined mapping.

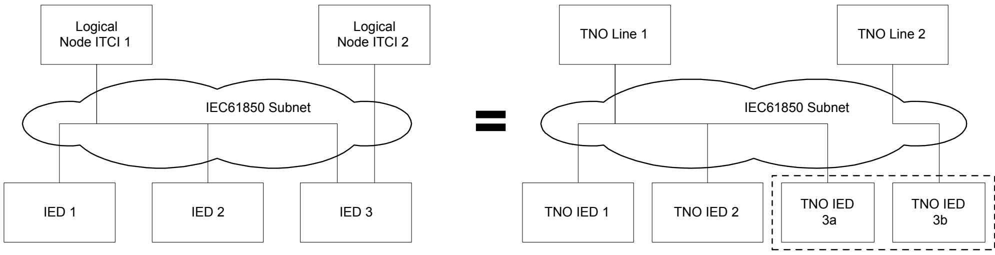
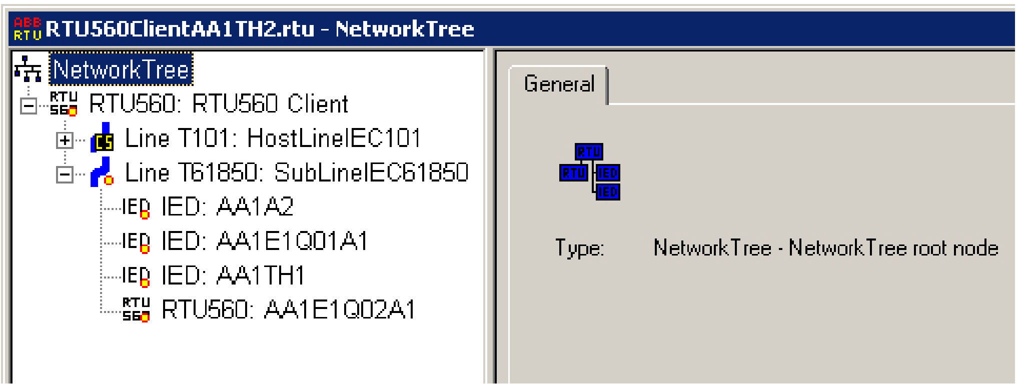


Figure 64: Mapping of client network

The logical node ITCI (Telecontrol interface) is mapped to an IEC61850 line in RTUtil500 and the IED's are mapped one to one. If an IED is part of two IEC61850 networks this IED is represented by two IED's in RTUtil500. An example configuration contains the following steps.

- In the network tree add an RTU. Then add an IEC61850 line at the RTU.
- Add substation IED's and/or external configured subordinated RTU as server (see figure below).

ADVICE
Be sure to use as name for the IED's the same as defined in the SCD file. This is important for the correct mapping during the SCD file import described below.



- At the IEC61850 line set the IED name and the access point name to the defined names from the SCD file. These are the names of the RTU client. If an RTU has connections to different IEC61850 networks each connection is represented by a separate line. The IED name at these lines could be the same but the access point name must be different in this case. For a description about the possibility to use multiple RTU clients in one IEC61850 network see chapter "Extensive RTU client configurations". The figure below shows the setting of IED and access point name.

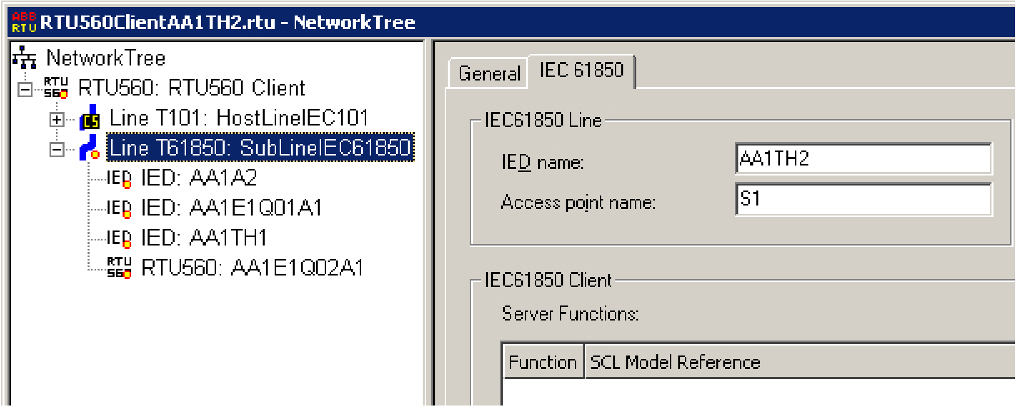


Figure 65: Setting IED and access point name

- In the next steps all RTUs and IED's are linked to the hardware tree. Then build your RTU hardware and link the IEC61850 line to an Ethernet interface at a CMU module. This is shown in the figure below.

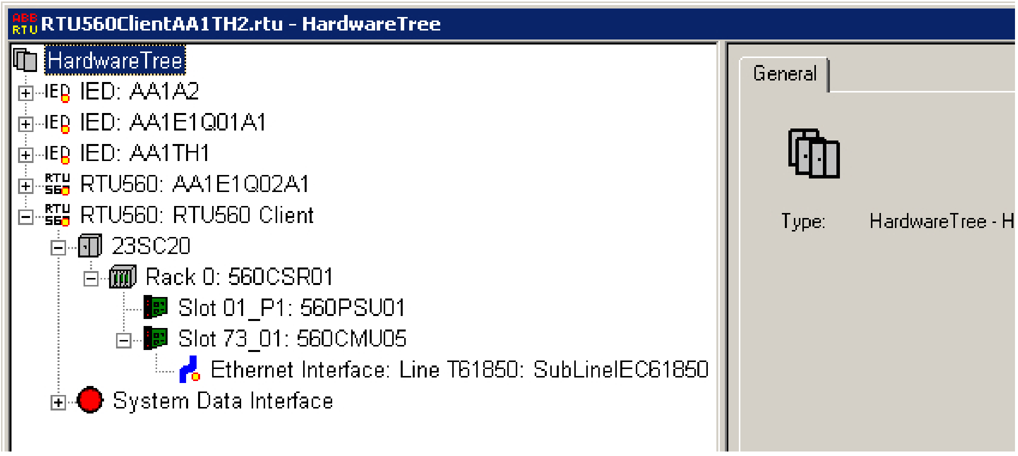


Figure 66: RTU client example hardware tree

- Finish the basis configuration with needed parameter at RTU (e.g. time administration) and at CMU module (e.g. IP addresses). Be sure to save your configuration as pattern project
- As prerequisite for the further IEC61850 configuration an Excel import file is needed. Do an Excel export from your pattern project to create this Excel import file (Menu: 'Extra - Excel Export')

13.4.2 Export of IID file

With the pattern project created before, the IEC61850 data model of the client is exported to an IID file. The IID file is used in the system configuration to define the IEC61850 network, the relationship between Logical Nodes and IED's and the communication between the IED's and the RTU client.

The IID file export is started in the menu via 'Extra – IID Export'. When selected, a dialog opens to choose the IID file name and the IEC61850 IED to export. The dialog is shown below.

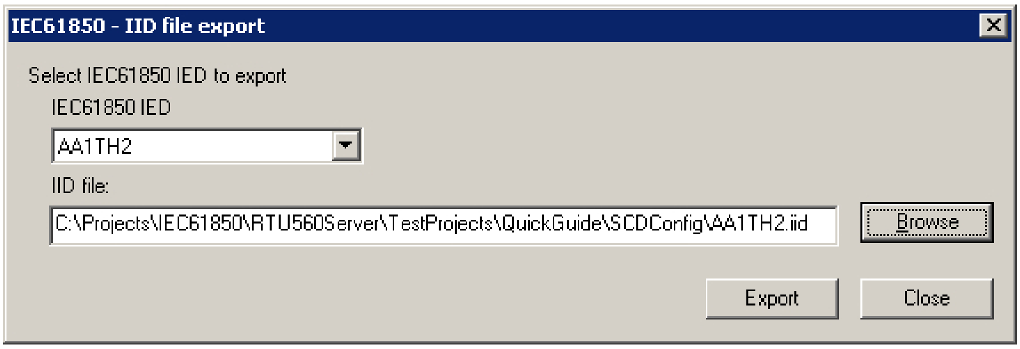


Figure 67: IID file export dialog

In the pull down list the IEC61850 IED to export is chosen. The IED name listed here is the name defined at the IEC61850 line (see chapter above). The IID file name could be typed in or selected in the file system with the ‘Browse’ button.

The export is started by pressing the ‘Export’ button. The result of the export is shown in a message box and erroneous results are written to a log file. There are failures in the IEC61850 configuration if an export of an IED is not possible (‘Export’ button not selectable). These are shown in the consistency check. For more information about the IEC61850 consistency check see chapter ‘Detailed RTU server engineering’.

The dialog remains open to export IID files for all IEC61850 IED’s defined in the project. The dialog can be left via the ‘Close’ button.

ADVICE

The IEC61850 data model of the client, exported in the IID file, must not be changed during system configuration. It is not allowed to add, change or delete any data nodes like logical devices, logical nodes or data attributes. Especially the private nodes from the data model must remain in the resulting SCD file. Please make sure that the system configuration tool doesn't remove the private nodes unintentional.

13.4.3 SCD File Import

For the RTU client the SCD file import is used to configure the data points received from and send to server IED's. The data points identified during the SCD file import are written to the Excel import data sheet (see chapter "Excel Import File" for more information).

ADVICE

The configuration tool RTUtil500 supports SCD files with the namespace of IEC 61850 edition 1 only (SCL2003). In case the SCD file contains certain extended attributes or data types of IEC 61850 edition 2, parsing of the SCD file in RTUtil500 may fail.

The SCD file import and Excel import file synchronization could be started by the user from the RTUtil500 menu if a project is active. The menu item is located in ‘Extra – SCD Import’. When selected, a wizard starts that guides the user through the import and synchronization process. The steps are:

1 The user has to select the SCD file to import. The user can type the complete path of the file or browse to the file in the file system. If the option 'Evaluate Logical Nodes Inputs' is activated, the RTUtil500 import flag is automatically set for data points that are listed in the input section. If the 'Next' button is pressed the SCD file is imported and parsed. After that the dialog continues with the next step.

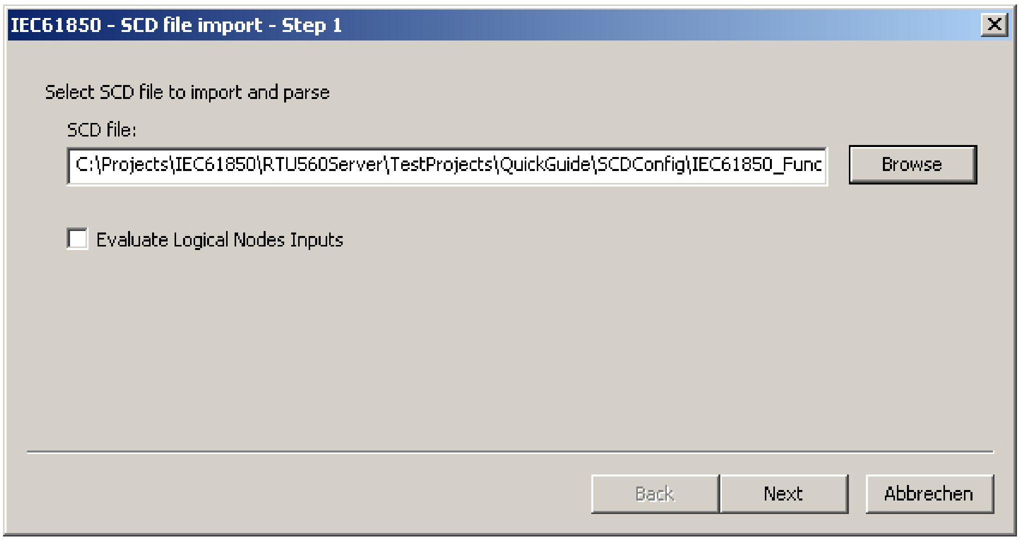


Figure 68: Select SCD file

2 The user has to select the Excel import file that should be synchronized with the SCD file. The user can type the complete path of the file or browse to the file in the file system. If the 'Next' button is pressed the Excel import file is read and the dialog continues with the next step.

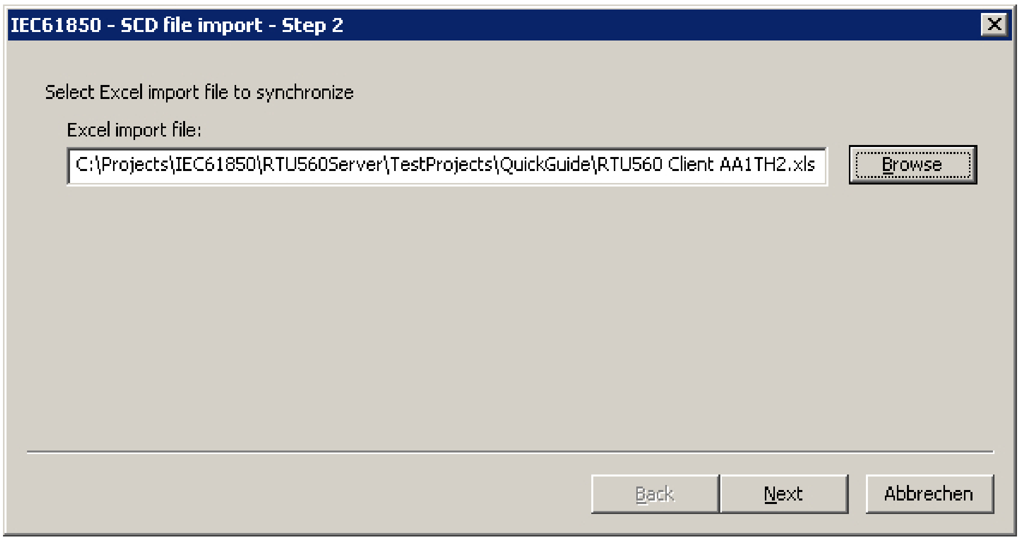


Figure 69: Select Excel import file

3 In the last dialog the user specifies the mapping between the RTU IEC61850 IED found in the SCD file and the sheets from the Excel import file. For the RTU client each sheet in the Excel import file represents an IEC61850 line in RTUtil500.

In the list of found IED's the type of the IED is shown (Client or server). Be sure that the client configured in the project is mapped only. Not linked IED will not be handled during synchronization.

The defined mapping is stored in the Excel import file. So, if the Excel import file is used a second time the user must not specify the mapping again.

If the 'Finish' button is pressed the processing starts. Result of the synchronization process is an Excel import file filled with data points that should be received from IED's in the network.

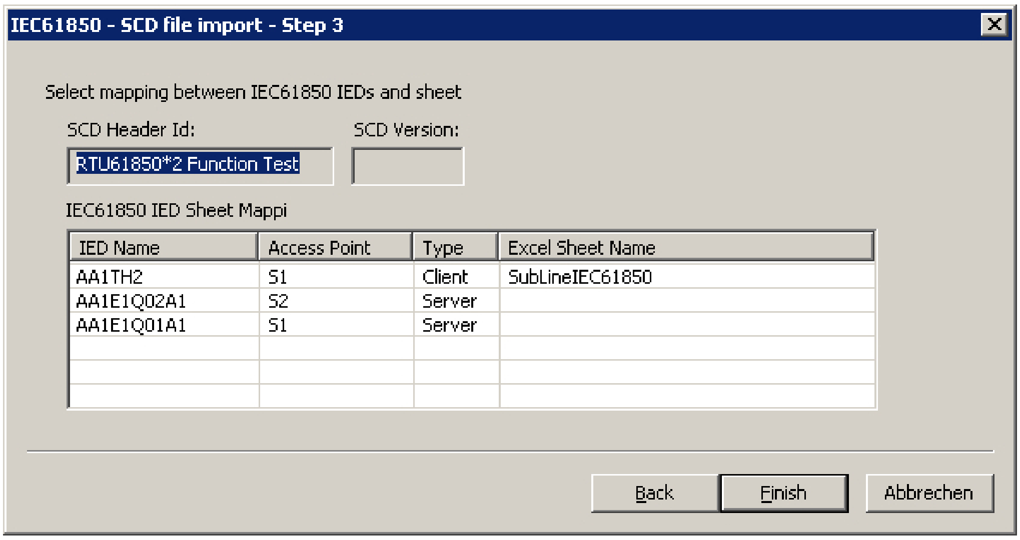


Figure 70: Select mapping between client IED's and sheet names

As written in the engineering overview the SCD import extracts data points from server IED's only, if these server are assigned to the RTU client. The assignment is done within the network tree by adding IED's to an IEC61850 sub device line (see description in chapter 'Network and Hardware tree' above). During the SCD import this assignment is considered in the following way:

- A server IED found in the IEC61850 sub network of the RTU client that is not assigned in the network tree is ignored.
- A server IED that is assigned in the network tree but is not found in the IEC61850 sub network of the RTU client is ignored as well.

As result the Excel import file contains for a specific RTU client data points of server IED's only, if these IED's are assigned to the client and exists in the IEC61850 sub network. Information and warnings about not assigned server IED's can be found in the log file created during the SCD file import. This log file can be found in the project directory with the name 'SCLProcess.log'.

The Excel import file generated in the synchronization is used during the normal Excel import process described in chapter "Excel Interface".

13.4.4 Excel Import File

In the engineering process RTUtil500 imports the SCD file, extracts all data points and writes these data points to the Excel import file.

The Excel import sheet for the IEC61850 sub device communication interface contains in one section the IEC61850 process object references and in another section additional parameters like for example scaling limits. The process object references are filled during the import of the SCD file. The user may not modify the data in this section. The object references are for information purposes only. See figure below.

IED name	Logical Device Instance Name	Logical Node Prefix	Logical Node Class	Logical Node Instance	Signal Data Object Name	Signal Common Data Class	Signal Data Attribute Name	Signal Data Type	Signal Function Code	Signal Support select before operate	Signal Support enhanced security
IEC61850 Address											
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	Y / N	Y / N
I8IN	*I8LD*	*I8LNP*	*I8LNC*	*I8LNI*	*I8SDN*	*I8CDC*	*I8SAN*	*I8SDT*	*I8SFC*	*I8SSO*	*I8SES*
AA1A2	LD0	CC	RPLD	1	Beh	INS	stVal	Enum	ST	N	N
AA1A2	LD0	CC	RPLD	1	Mod	INC	Oper.ctlVal	Enum	CO	N	N
AA1A2	LD0	CC	RPLD	1	Op	ACT	general	BOOLEAN	ST	N	N
AA1A2	LD0	C	MMXU	1	A.phsA	CMV	cVal.mag.f	FLOAT32	MX	N	N
AA1A2	LD0	C	MMXU	1	A.phsB	CMV	cVal.mag.f	FLOAT32	MX	N	N
AA1A2	LD0	C	MMXU	1	A.phsC	CMV	cVal.mag.f	FLOAT32	MX	N	N
AA1A2	LD0	C	MMXU	1	Beh	INS	stVal	Enum	ST	N	N
AA1A2	LD0	C	MMXU	1	Mod	INC	Oper.ctlVal	Enum	CO	N	N
AA1A2	LD0	CV	MMXU	1	ALd	SPS	stVal	BOOLEAN	ST	N	N
AA1A2	LD0	CV	MMXU	1	ALg	SPS	stVal	BOOLEAN	ST	N	N
AA1A2	LD0	CV	MMXU	1	Amp	MV	mag.f	FLOAT32	MX	N	N
AA1A2	LD0	CV	MMXU	1	Beh	INS	stVal	Enum	ST	N	N

Figure 71: IEC61850 process object reference

The additional parameters are the scaling limits and the definition of a default command qualifier. The scaling limits are relevant for AMI, ASO, DMI and DSO data points. The default command qualifier is set for SCO, DCO and RCO commands. The parameters could be edited by the user. See figure below.

Maximum value (100%) (AMI/ASO/DMI/DSO)	Minimum value (-100%) (AMI/ASO/DMI/DSO)	Default command qualifier (DCO, RCO, SCO)
IEC61850 Sub Parameter		
-2147483648 .. 2147483647	-2147483648 .. 2147483647	run interl., run synchro. run synchrocheck run interlocking no check
I8MXV	*I8MIV*	*I8DCQ*
32767	-32767	
		run interl., run synchro.
200000	-100000	
		no check
32767	-32768	

Figure 72: IEC61850 additional parameter

The default command qualifier defines the value of the IEC61850 ‘check’ data attribute in case the qualifier of command is not set to predefined value. The following values are possible for the qualifier of commands:

Qualifier of command (QU)	Command check attribute on IEC61850 set to
0 .. 8	default command qualifier
9	no interlocking, run synchrocheck
10	run interlocking, no synchrocheck
11	no interlocking, no synchrocheck
12	run interlocking, run synchrocheck
13 .. 31	default command qualifier

The IEC61850 process object reference is not sufficient for the identification of the data point. To be able to identify the IEC61850 data points the information from the substation section are written to the Excel import sheet as reference. The substation references are for information purposes only and may not be changed by the user. The following figure shows the substation information in the Excel sheet.

Substation	Voltage level	Bay	Conducting equipment	Subequipment
Substation				
ASCII	ASCII	ASCII	ASCII	ASCII
SUST	*SUVL*	*SUBA*	*SUEQ*	*SUSEQ*
AA1	E1	Q03	QB2	
AA1	E1	Q03		
AA1	E1	Q03	QC2	
AA1	E1	Q05	T1	
AA1	E1	Q05		
AA1	K1	Q21	QB1	
AA1	K1	Q21	BI1	
AA1	K1	QBB	QA1	

Figure 73: IEC61850 substation reference

The extraction of data points from the SCD file could be done with an empty Excel import file or with a file filled from a previous SCD file import. In this case the Excel import file is synchronized with latest SCD file. During synchronization three cases are considered:

- New data point: The data point is part of SCD file but not part of Excel import file.
- Updated data point: At least one parameter of the data point is changed in SCD File compared to the Excel import file.
- Deleted data point: The data point is part of Excel import file but not part of SCD file.

The result of the synchronization is written to the first section of the Excel import file. A new column called 'Modified' contains a character for each data point. The meaning of the characters is:

- N New data point
- U Updated data point
- D Deleted data point

The following figure shows an example from the Excel import file.

Signal type	System data type (SEV, SSC)	RTUtil560 import	Modified
Signal			
3 ASCII	4 ASCII	Y / N	1 ASCII
STTY	*STDT*	*STIM*	*STMD*
SPI		Y	U
SPI		Y	U
SPI		Y	D
SPI		Y	U
SPI		Y	N
SPI		Y	N
SPI		Y	N

Figure 74: Modified column

13.4.5 IEC61850 server functionality

The RTU IEC61850 client additionally provides an IEC61850 server with restricted functionality. The provided IEC61850 server functions are:

- Authority control for the whole substation
- Logical node health identification

The IEC61850 server functions are defined as private attributes in the IED capability file (ICD file) of the RTU. There is no need to configure the server functionality in RTUtil500. Not the ICD file but the generated IID file should be used (See chapter ‘"IEC61850 client engineering")

During the SCD file import the IEC61850 server functions are identified by the private attributes at the IEC61850 data object of the RTU IED section. The found server functions are written to the AODM data model in RTUtil500 (see next chapter).

For detailed information about the SCL data model of the IEC61850 server see document "Subdevice Communication Interface IEC61850". This document contains the description of the provided logical nodes and data objects and there relation to the RTU firmware functions.

13.4.6 User Interface

According to other subdevice communication interfaces the IEC61850 parameters are shown in separate property pages. There are parameters for IEC61850 lines and for IED data points. The IED itself has no IEC61850 parameters.

The IEC61850 subdevice line parameters are the IED name, the access point name and the IEC61850 server functions. The server functions are updated during the SCD file import described above. The server function parameters are not editable. See figure below.

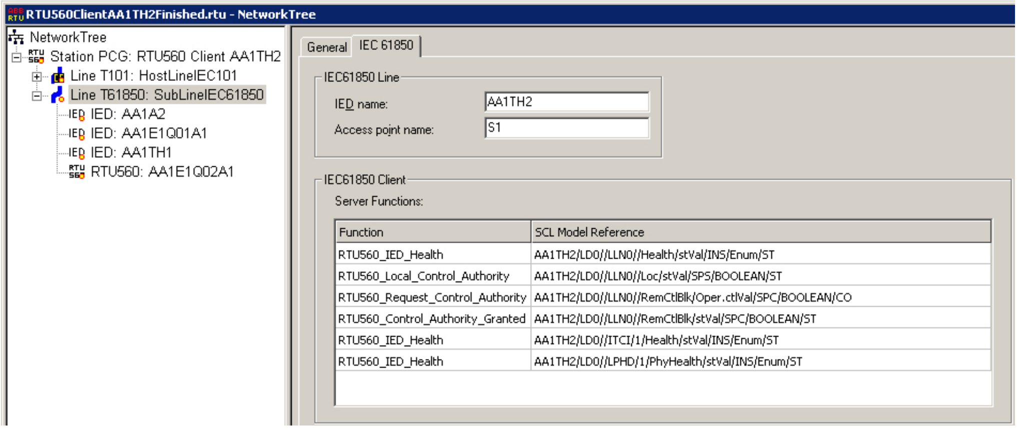


Figure 75: IEC61850 line parameter

Addresses and parameters of IEC61850 data points will be displayed in the properties page of a selected data point. Two sections are grouped in this page.

a) Address group:

All information imported from the process object reference section (see chapter 'Excel Import File') are as shown in the page. All fields within this group are not editable. The address element ‘logical node instance name’ is a concatenation of logical node prefix, logical node class and logical node instance number. The signal attribute is a concatenation of signal data object name and signal data object attribute separated by a dot character.

b) Sub parameter:

Processing parameters that are configuration specific will be displayed in this editable group. Only parameters referring to the selected data point type are shown. If no parameter is needed for the selected data point, the group is not displayed.

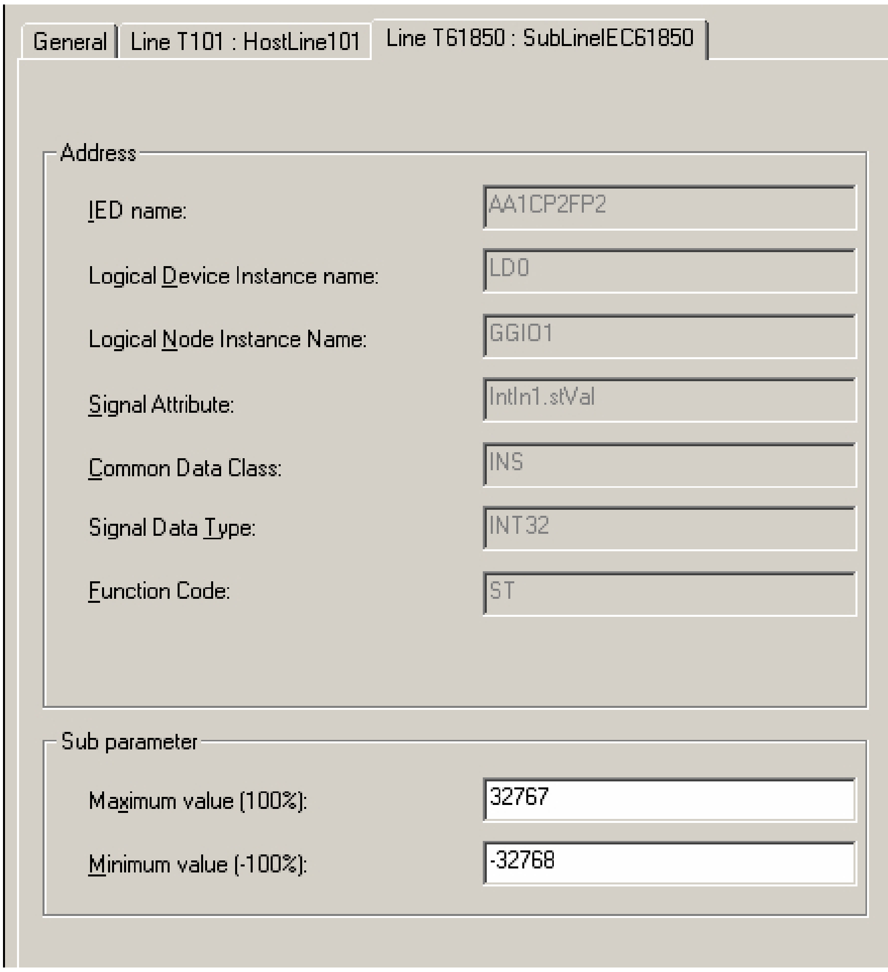


Figure 76: IEC61850 client data point parameter

13.4.7 Extensive RTU client configurations

The maximum number of server IED's per RTU client is limited. For the concrete limit see document "Subdevice Communication Interface IEC61850". In case a configuration includes more server IED's than the limit, the IED's must be subdivided over several RTU clients. This configuration must be considered in the structure of the IEC61850 network. There are two possible structures:

1 The RTU clients are connected to separate IEC61850 sub networks, or
2 The RTU clients are connected to the same IEC61850 sub network.

In the second network structure all IED's are in the same sub network and a communication between all IED's via GOOSE is possible. The next figure shows examples for both possibilities.

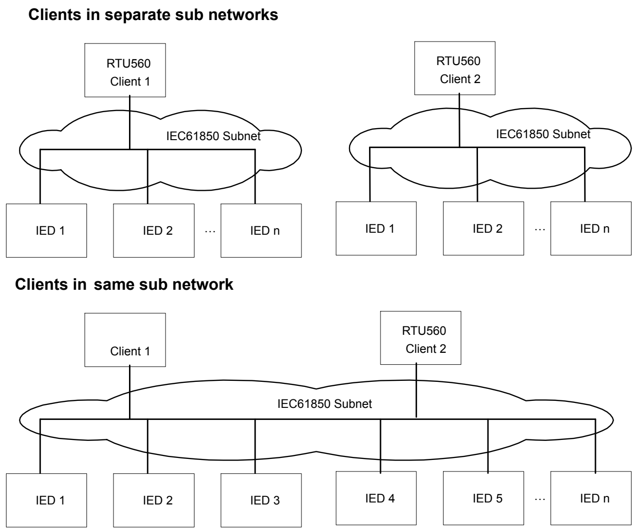


Figure 77: Extensive client configurations

In both configurations it is important that the limited number of server IED's are assigned to the different clients. This assignment is done in the RTUtil500 network tree as described in the chapters above. That means for the second network structure that only a part of the server IED's available in the sub network are assigned to an RTU client.

For the RTU client IED naming conventions the following rules apply for extensive configurations:

- In case the RTU clients are in different sub networks the name of the client can be same or different. If the client name is the same the access point names must be different. That means in the SCD file the RTU clients can be modeled as one or separate client IED's.
- In case the RTU clients are in the same sub network the name of the client IED's must be different. It is not possible to use the same IED name with different access point names. This requires different client IED's in the SCD file.

Communication between the different IEC61850 clients is not possible in both cases. That means it is not possible to exchange datasets or commands between the IEC61850 clients. A corresponding configuration in the SCD file is ignored during the import.

13.5 Detailed RTU server engineering

13.5.1 Network and Hardware tree

As for the RTU client the IEC61850 server network defined in the system configuration must be mapped to an equivalent structure in the RTUtil500 network tree. The following figure shows the defined mapping.

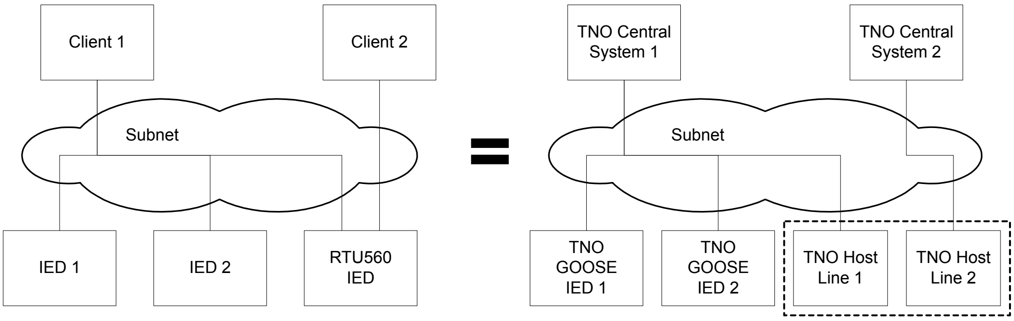


Figure 78: Mapping of server network

Each connection of the RTU to a network is mapped to an IEC61850 line in RTUtil500. The clients are represented as central stations and the IED's providing GOOSE data are mapped as GOOSE IED's to RTUtil500. If an RTU server has connections to different IEC61850 networks the modeling in RTUtil500 contains host interface lines for each connection. An example configuration contains the following steps.

- In the network tree add an RTU. Then add an IEC61850 line at the RTU.
- Add control systems and/or external configured Host RTU as client (see figure below).
- Add all GOOSE IED's that provide data for the RTU server and set the name in the General tab to the IED name from the SCD file (see figure below).

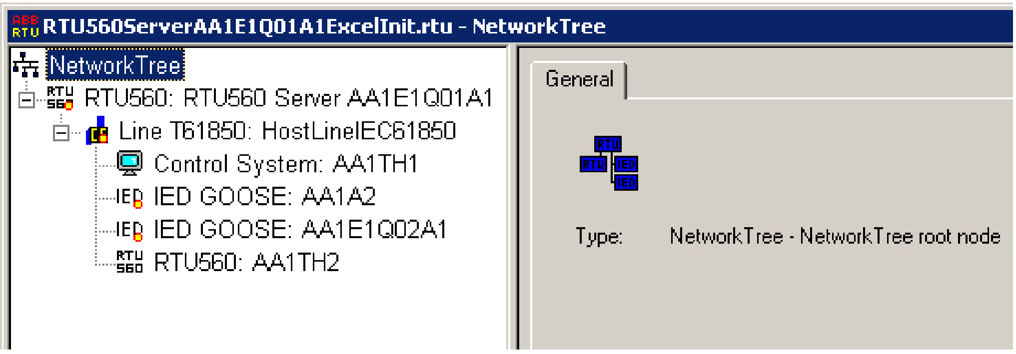


Figure 79: RTU client example network tree

- At the IEC61850 line set the own IED and access point name to the defined names for the system configuration. These are the names of the RTU server. If an RTU has connections to different IEC61850 networks each connection is represented by a separate line. The IED name at these lines could be the same but the access point name must be different in this case. The figure below shows the setting of IED and access point name.

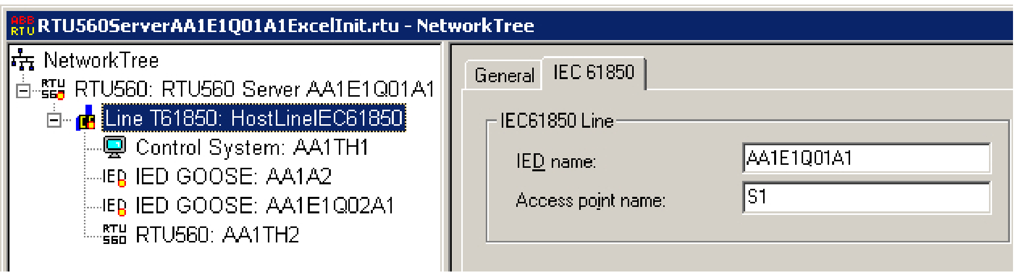


Figure 80: Setting own IED and access point name (Server)

- At the defined control systems and external configured Host RTU the IEC61850 IED name of the client and the IP address must bet set. These are the definitions from the system configuration.

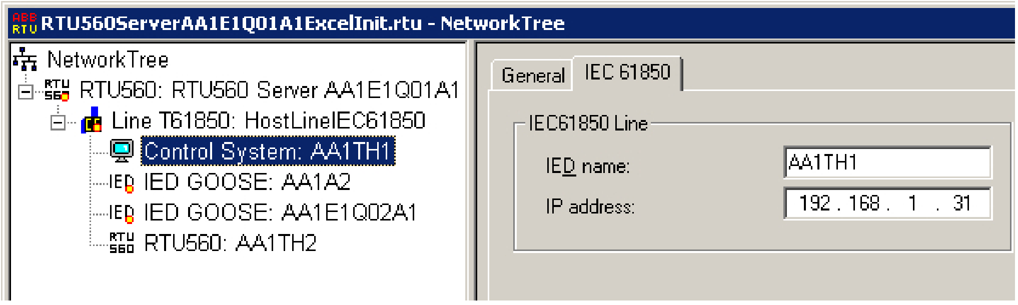


Figure 81: Setting IED name and IP address for control system

- In the next steps the RTU server and all GOOSE IED's are linked to the hardware tree. After that, the RTU hardware has to be built and the IEC61850 line has to be linked to an Ethernet interface at a CMU module.
- When working with GOOSE data points, a PLC activity has to be added to the CMU module and at least one PLC task must be configured. An example of the hardware tree is shown in the next figure.

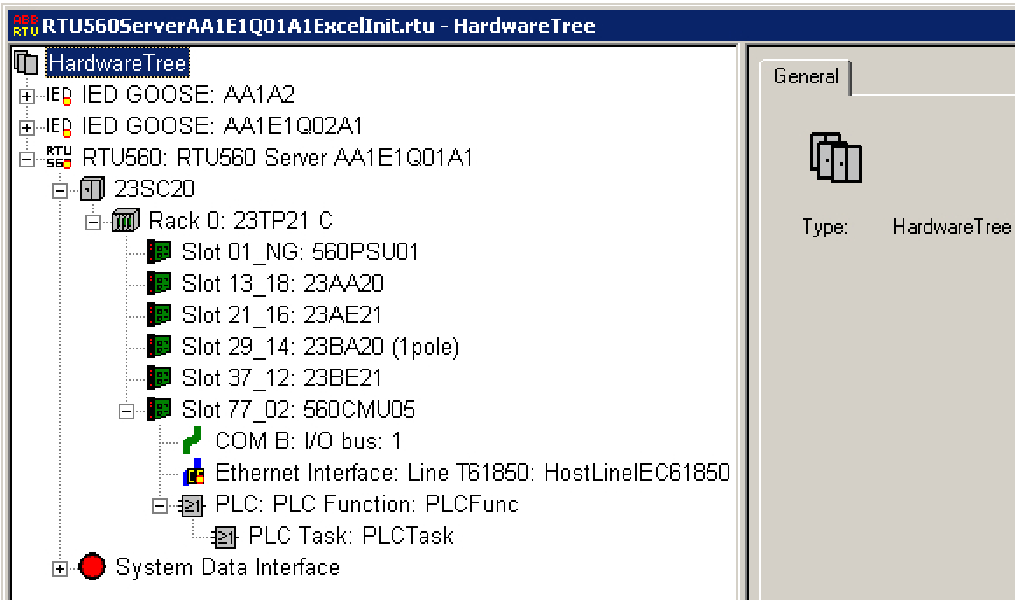


Figure 82: RTU server example hardware tree

- Continue the configuration with needed parameter at RTU (e.g. time administration) and at CMU module (e.g. IP addresses).
- At the Ethernet interface configure a unique host number for all IEC61850 clients (Control systems or external configured Host RTU). Decide whether the IEC61850 server should consider local control authority or not (could be set for the whole server only, not per connected client). The figure below shows the setting at the Ethernet Interface.

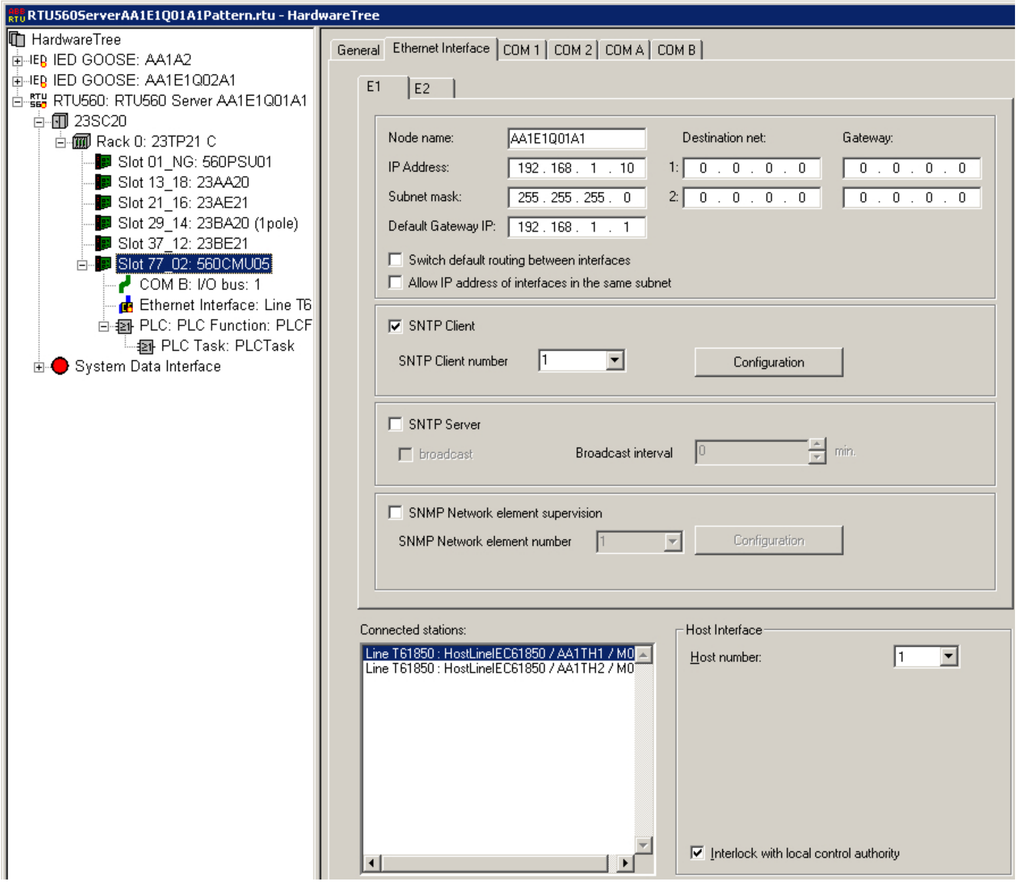


Figure 83: Setting host number for IEC61850 clients

- When finished be sure to save the configuration as pattern project.
- As basis for the further IEC61850 configuration an Excel import file is needed. Do an Excel export from your pattern project to create this Excel import file (Menu: 'Extra - Excel Export'). The Excel import file contains an additional sheet for the GOOSE data points. This sheet includes for document purposes the system events of the GOOSE IED's. These system events could not be imported (generates a warning during the import) and should be set to 'not import'.

13.5.2 IEC61850 data modeling

The IEC61850 data modeling of the RTU is done in the Excel import sheet generated from the pattern project (Excel export). At first in the modeling an IEC61850 object reference is set for an RTU data point. The columns used here are grouped in the section IEC61850 address shown below. The modeling goes from left to right with the following steps.

1. Type a logical device instance name in the column 'LD'.
2. Choose from a group of logical node types. This column is used to make the modeling clearer and is not imported.
3. Choose a logical node prefix in the column 'LNP'. The predefined prefixes in the pull down list are the mandatory prefixes for the logical node class 'GGIO'. To configure no or any other prefix delete the data validation rule for the cell. This is done by selecting a cell, open the data validation dialog via 'Data – Validation' and set the validation criteria to 'Any value'.
4. Choose a logical node class in the column 'LNC'. The choices in the pull down list of this cell depend on the selected logical node group.
5. Type a logical node instance in the column 'LNI'.

6 Choose a data object name in the column ‘SDN’. The choices in the pull down list of this cell depend on the selected logical node class.
7 Choose a data attribute name in the column ‘SAN’. The choices in the pull down list of this cell depend on the selected data object name.
8 For system events (SEV) choose whether these should be used or not in the column ‘IU’. The default value of this column is ‘Yes’.

The columns ‘CDC’ (Common Data Class), ‘SDT’ (Data Type) and ‘SFC’ (Function Code) are filled by predefined formulas. These cells must not be set by the user. Be sure that these formulas are copied when creating new rows.

Logical Device Instance Name	Logical Node Grroup	Logical Node Prefix	Logical Node Class	Logical Node Instance	Signal Data Object Name	Signal Common Data Class	Signal Data Attribute Name	Signal Data Type	Signal Function Code	In use (SEV)
IEC61850 Address										
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	Y / N
E8LD		*E8LNP*	*E8LNC*	*E8LNI*	*E8SDN*	*E8CDC*	*E8SAN*	*E8SDT*	*E8SFC*	*E8IU*
LD0	System_LN		LLN0		Loc	SPS	stVal	BOOLEAN	ST	
LD0	LN_Control	A	CSWI	1	Loc	SPS	stVal	BOOLEAN	ST	
LD0	LN_Control	A	CSWI	1	Pos	DPC	Oper.ctlVal	BOOLEAN	CO	
LD0	LN_Control	A	CSWI	1	Pos	DPC	stVal	Dbpos	ST	
LD0	LN_Control	A	CILO	1	EnaOpn	SPS	stVal	BOOLEAN	ST	
LD0	LN_Control	A	CILO	1	EnaCls	SPS	stVal	BOOLEAN	ST	
LD0	LN_Control	A	CALH	1	GrAlm	SPS	stVal	BOOLEAN	ST	
LD0	LN_Generic_IO	SP	GGIO	1	Ind	SPS	stVal	BOOLEAN	ST	
LD0	LN_Generic_IO	SP8	GGIO	1	Ind1	SPS	stVal	BOOLEAN	ST	
LD0	LN_Generic_IO	SP16	GGIO	1	Ind2	SPS	stVal	BOOLEAN	ST	
LD0	LN_Generic_IO	DP	GGIO	1	DPI	DPS	stVal	Dbpos	ST	
LD0	LN_Generic_IO	DP8	GGIO	1	DPI3	DPS	stVal	Dbpos	ST	

Figure 84: RTU IEC61850 object reference modeling

The mapping of IEC61850 object references to RTU data point is described in detail in the document “Host Communication Interface IEC61850” (1KGT 150 702). As reference the following table summarizes the supported common data classes, their data attributes and the possible mappings to RTU data point types.

Common Data Class	Attribute Name	Default RTU data point type	Other RTU data point type
ACD	dirGeneral	DPI	BSI
ACD	dirNeut	DPI	BSI
ACD	dirPhsA	DPI	BSI
ACD	dirPhsB	DPI	BSI
ACD	dirPhsC	DPI	BSI
ACD	general	SPI	DPI, SEV
ACD	neut	SPI	DPI, SEV
ACD	phsA	SPI	DPI, SEV
ACD	phsB	SPI	DPI, SEV
ACD	phsC	SPI	DPI, SEV

Table 2: RTU data point mapping

Common Data Class	Attribute Name	Default RTU data point type	Other RTU data point type
ACT	general	SPI	DPI, SEV
ACT	neut	SPI	DPI, SEV
ACT	phsA	SPI	DPI, SEV
ACT	phsB	SPI	DPI, SEV
ACT	phsC	SPI	DPI, SEV
BCR	actVal	ITI	-
BCR	frVal	ITI	-
BSC	Oper.ctlVal	RCO	-
BSC	valWTr.posVal	STI	-
CMV	cVal.mag.f	MFI	AMI
DPC	Oper.ctlVal	DCO	-
DPC	stVal	DPI	BSI
DPS	stVal	DPI	BSI
INC	Oper.ctlVal	ASO	BSO, DSO
INC	stVal	AMI	BSI, DMI, DPI, MFI
INS	stVal	AMI	BSI, DMI, DPI, MFI
ISC	Oper.ctlVal	BSO	-
ISC	valWTr.posVal	STI	-
MV	mag.f	MFI	AMI
SPC	Oper.ctlVal	SCO	-
SPC	stVal	SPI	DPI, SEV
SPS	stVal	SPI	DPI, SEV

Table 2: RTU data point mapping

To finish the IEC61850 data modeling the additional parameters for datasets and scaling of analog values must be set in the section ‘IEC61850 Host Parameter’ shown below.

Client Data Set Name	Client Data Set Suffix	GOOSE Data Set Name	GOOSE Data Set Suffix	Maximum value (100%) (AMI/ASO/DMI/DSO)	Minimum value (-100%) (AMI/ASO/DMI/DSO)
IEC61850 Host Parameter					
ASCII (choose from list)	A, B, C or D	Interlocking, Protection	A, B, C or D	-2147483648 .. 2147483647	-2147483648 .. 2147483647
E8CDN	*E8CDS*	*E8GDN*	*E8GDS*	*E8MXV*	*E8MIV*
StatUrg	A				
StatNrml	B				
		Interlocking	B		
StatIed	C				
StatIed	C				
StatIed	C				

Figure 85: RTU IEC61850 host parameter

In the columns for the data sets the user chooses which data points should be send in a client data set (MMS data set) and which data points should be send in a GOOSE data set. It is possible to send the same data point in a client data set and as well in a GOOSE data set. For a correct configuration a predefined data set name and suffix must selected.

ADVICE

The data set configuration of the RTU must not be changed in the system configuration tool. For the RTU the destination of the data sets is defined in the system configuration tool only.

The maximum and minimum values for scaling of analog values are set in the columns ‘MXV’ and ‘MIV’. For the measurement information AMI and DMI the scaling limits are -2^{31} to $2^{31} - 1$. For the setpointcommands ASO and DSO the scaling limits are -32767 to 32768

In the Excel sheet for the GOOSE data points the system events should be set to ‘not import’ (like stated before). Do not fill any further data in the GOOSE data point sheet. This sheet is filled by the SCD file import.

When the IEC61850 data modeling is completed the Excel sheet is imported in RTUtil500 (Menu: ‘Extra – Excel Import’). Make sure that the GOOSE data point sheet is not imported in this step.

13.5.3 Export of IID file

The IEC61850 data model of the RTU server created before is now exported to an IID file. The IID file is used in the system configuration to define the IEC61850 network, the relationship between Logical Nodes and IED’s and the communication between the IED’s and the IEC61850 clients.

To be able to export an IID file the data model must pass the consistency check. Before exporting inspect the consistency check for IEC61850 errors (Menu: Project – Check Consistency). The following rules are checked:

- IED and access point name must be set at IEC61850 line.
- In one IED the access point names must be all different.
- Every access point of an IED must be from the same type (Server or client).
- Client data set name and suffix must be set both.
- GOOSE data set name and suffix must be set both.
- A data object is allowed once in a data set only.
- The mandatory IEC61850 parameter must not be empty. These parameter are 'IED Name', 'Logical Device Instance', ' Logical Node Class', 'Data Object', 'Common Data Class', 'Data Attribute', 'Data Type' and 'Function Code'.
- The IEC61850 object reference must be unique within an IED. That means the reference must be unique over all access points of the IED.
- The host numbers defined for the IEC61850 clients must be unique per client.
- IEC61850 client name and IP address must be set.
- A mapping to an RTU data point must be possible for the defined IEC61850 object reference. That means the IEC61850 object reference is not supported by the RTU.
- The mapping of RTU data point type and IEC61850 object reference must be allowed. See reference table in the last chapter.
- At logical node class LLLN0 no prefix or instance is allowed.
- At logical node class GGIO one predefined prefix must be set.

The IID file export is started in the menu via 'Extra – IID Export'. When selected a dialog opens to choose the IID file name and the IEC61850 IED to export. The dialog is same like shown in the chapter 'Detailed RTUtil500 client engineering'.

In the pull down list the IEC61850 IED to export is chosen. The IED name listed here is the name defined at the IEC61850 line (see chapter above). The IID file name could be typed in or selected in the file system with the 'Browse' button.

The export is started by pressing the 'Export' button. The result of the export is shown in a message box and erroneous results are written to a log file. The dialog remains open to export IID files for all IEC61850 IED's defined in the project. Leave the dialog with the 'Close' button.

ADVICE
The IEC61850 data model of the server, exported in the IID file, must not be changed during system configuration. It is not allowed to add, change or delete any data nodes like logical devices, logical nodes or data attributes. Especially the private nodes from the data model must remain in the resulting SCD file. Please make sure that the system configuration tool doesn't remove the private nodes unintentional.

13.5.4 SCD file import

For the RTU server the SCD file import is used to configure the GOOSE data points received from other IED's. The GOOSE data points identified during the SCD file import are written to the Excel GOOSE receive data sheet (see chapter "IEC61850 Excel Import Sheets" for more information).

ADVICE
The SCD file import is necessary in every RTU server project. Even if no GOOSE communication is used the SCD file import must be done to store the SCD file in the RTUtil500 project. Without the SCD file no RTU configuration file could be generated. The configuration tool RTUtil500 supports SCD files with the namespace of IEC 61850 edition 1 only (SCL2003). In case the SCD file contains certain extended attributes or data types of IEC 61850 edition 2, parsing of the SCD file in RTUtil500 may fail.

The menu item to start the SCD file import is located in ‘Extra – SCD Import’. When selected a wizard starts that guides the user through the import and Excel synchronization process. The process here is the same like for the RTU client.

In the first step of the wizard the user has to select the SCD file to import. The user can type the complete path of the file or browse to the file in the file system. If the ‘Next’ button is pressed the SCD file is imported and parsed. See chapter ‘Detailed RTUtil500 client engineering’ for a figure of the first step.

In the second step the user has to select the Excel import file with the GOOSE receive data sheet. The user can type the complete path of the file or browse to the file in the file system. If the ‘Next’ button is pressed the Excel import file is read and the dialog continues with the next step. See chapter ‘Detailed RTUtil500 client engineering’ for a figure of the second step.

In the last step the user specifies the mapping between the RTU IEC61850 IED found in the SCD file and the GOOSE receive data sheet from the Excel import file. In the list of found IED’s the type of the IED is shown (Client or server). Be sure that the server configured in the project is mapped only (see figure below). Not linked IED’s will not be handled during synchronization.

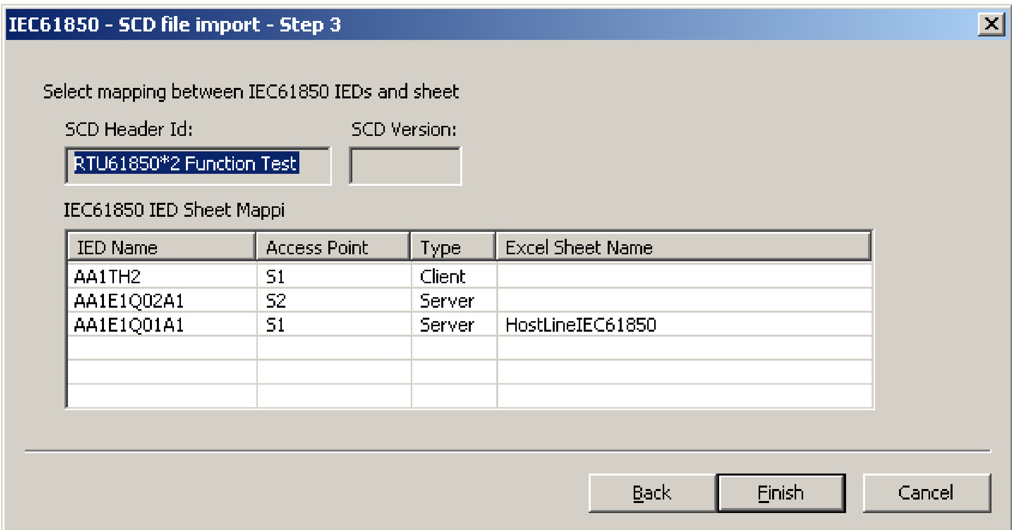


Figure 86: Select mapping between server IED’s and sheet names

The defined mapping is stored in the Excel import file. So, if the Excel import file is used a second time the user must not specify the mapping again. The SCD import could be done on an empty Excel sheet and as well on an Excel sheet filled by a former SCD file import (see next chapter).

The processing of the GOOSE receive data sheet starts if the 'Finish' button is pressed. Result of the processing is an Excel sheet filled with GOOSE data points received from other IED's. The Excel sheet generated is used afterwards in a further Excel import process to configure the GOOSE data points in RTUtil500 (see chapter 'Excel Interface'). Be sure to import the GOOSE sheet only during this step. Do not import the Excel sheets with the IEC61850 server data modeling again.

13.5.5 Excel GOOSE receive data sheet

During the SCD file import processing described in the chapter above RTUtil500 extracts all GOOSE receive data points and writes these to an Excel sheet. The GOOSE receive data sheet contains in one section the IEC61850 object references and in another section the additional parameters for scaling limits. The IEC61850 object references are filled during the import of the SCD file. The user may not modify the data in this section. The object references are for information purposes only. See figure below.

IED name	Logical Device Instance Name	Logical Node Prefix	Logical Node Class	Logical Node Instance	Signal Data Object Name	Signal Common Data Class	Signal Data Attribute Name	Signal Data Type	Signal Function Code	Signal Support select before operate	Signal Support enhanced security
IEC61850 Address											
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	Y / N	Y / N
I8IN	I8LD	I8LNP	I8LNC	I8LNI	I8SDN	I8CDC	I8SAN	I8SDT	I8SFC	I8SSO	I8SES
AA1E1Q01A1	LD0	Q0	CSWI	1	Pos	DPC	stVal	Dbpos	ST	N	N
AA1E1Q01A1	LD0	SP16	GGIO	3	Ind12	SPS	stVal	BOOLEAN	ST	N	N

Figure 87: GOOSE IEC61850 object reference

The additional parameters (shown below) relevant for GOOSE data points are the scaling limits. The column for the default command qualifier is not used for GOOSE data points because commands are not allowed as GOOSE data points. The scaling limits are relevant for AMI, ASO, DMI and DSO data points.

Maximum value (100%) (AMI/ASO/DMI/DSO)	Minimum value (-100%) (AMI/ASO/DMI/DSO)	Default command qualifier (DCO, RCO, SCO)
IEC61850 Sub Parameter		
-2147483648 .. 2147483647	-2147483648 .. 2147483647	run interl., run synchro. run synchrocheck run interlocking no check
I8MXV	I8MIV	I8DCQ

Figure 88: GOOSE IEC61850 additional parameter

As described for the RTU client the IEC61850 object reference is not sufficient for the identification of the data point. To be able to identify the IEC61850 data points the information from the substation section are written to the GOOSE data sheet as reference. The substation references are for information purposes only and may not be changed by the user. A figure of the substation information could be found in chapter ‘Detailed RTUtil500 client engineering’.

The extraction of GOOSE data points from the SCD file could be done with an empty Excel sheet or with a sheet filled from a previous SCD file import. In this case the Excel GOOSE receives data sheet is synchronized with the latest SCD file. During synchronization three cases are considered:

- New data point: The GOOSE data point is part of SCD file but not part of Excel sheet.
- Updated data point: At least one parameter of the GOOSE data point is changed in SCD File compared to the Excel sheet.
- Deleted data point: The GOOSE data point is part of Excel sheet but not part of SCD file.

The result of the SCD file import is written to the first section of the GOOSE receive data sheet. The column called ‘Modified’ contains a character for each data point. The meaning of the characters is:

- N New data point
- U Updated data point
- D Deleted data point

An example of the ‘Modified’ column could be found in chapter "Excel Import File".

13.5.6 User interface

According to other host communication interfaces the IEC61850 parameters are shown in separate property pages. For the RTU server there are parameters at the IEC61850 lines, at the IEC61850 central systems (clients), at the server data points and at the GOOSE IED data points. The GOOSE IED itself has no IEC61850 parameters.

The IEC61850 parameters of lines and central systems are described in the chapter "Network and Hardware tree". At the server data points IEC61850 addresses and parameters are displayed in the properties page of a selected data point. Two sections are grouped in this page (see figure below).

a) Address group:

All information imported from the IEC61850 object reference section (see chapter "IEC61850 data modeling") are shown in the page. All fields within this group are not editable. The address element ‘logical node instance name’ is a concatenation of logical node prefix, logical node class and logical node instance number. The signal attribute is a concatenation of signal data object name and signal data object attribute separated by a dot character.

b) Host parameter:

Processing parameters that are configuration specific will be displayed in this editable group. Possible parameters are the data set settings and the scaling limits. Only parameters referring to the selected data point type are shown. If no parameter is needed for the selected data point, the group is not displayed.

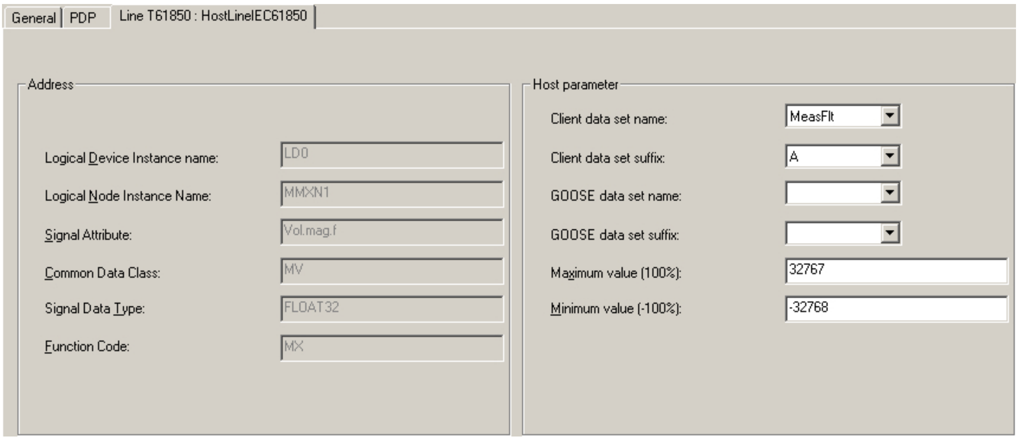


Figure 89: IEC61850 server data point parameter

At the GOOSE data points the property page contains as well IEC61850 addresses and parameters. The two sections here are:

a) Address group:

All information imported from the IEC61850 object reference section (see chapter "Excel GOOSE receive data sheet") are shown in the page. All fields within this group are not editable. The elements are concatenated like described before.

b) Sub parameter:

Processing parameters that are configuration specific will be displayed in this editable group. Possible parameters are scaling limits. Only parameters referring to the selected data point type are shown. If no parameter is needed for the selected data point, the group is not displayed.

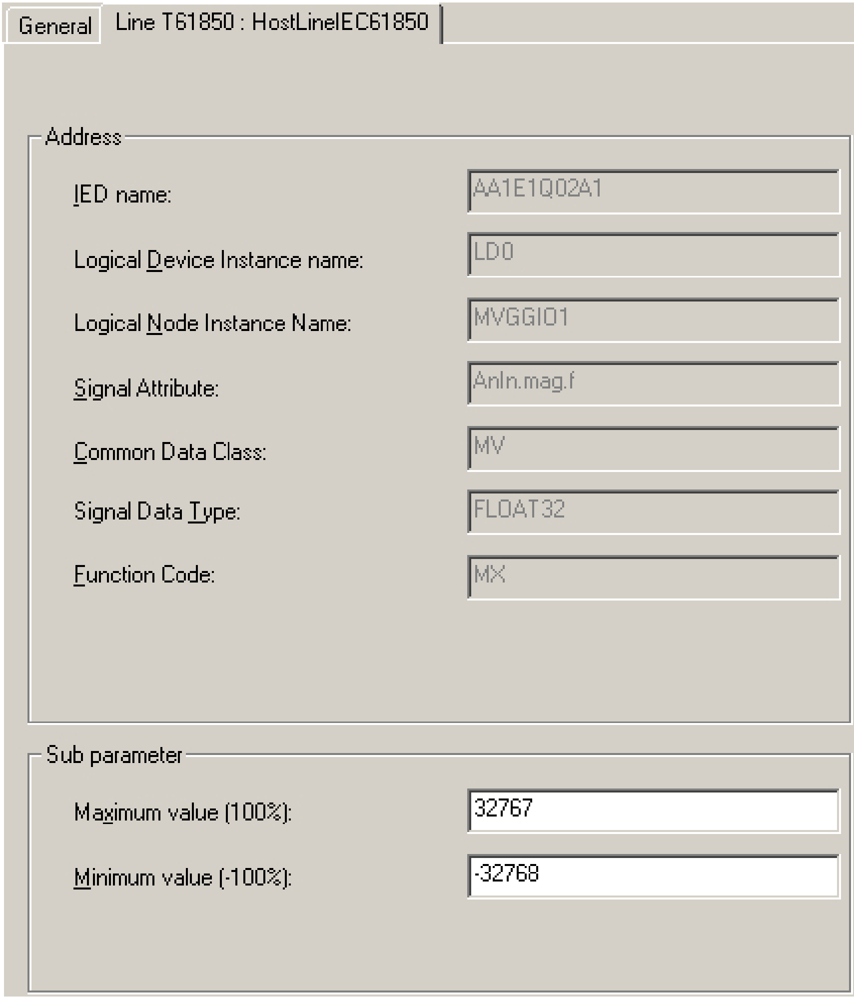


Figure 90: IEC61850 GOOSE data point parameter

14 PLC Engineering

14.1 Programmable Logic Control (PLC)

The RTU500 series can be equipped with a PLC software. To configure this functionality the PLC programming system MULTIPROG wt is used. MULTIPROG wt is a standard programming system for PLCs and it is based on the standard IEC 1131-3. Inside an RTUtil500 project using the PLC functionality there are some things to be configured.

Insert a PLC function:

The RTUtil500 PLC function is equivalent to a resource inside a MULTIPROG wt project. In a resource global variables can be declared, which are only valid within this resource. In a resource one or several tasks can be executed.

- Select a CMU and select "Add item ..." inside context menu. Following dialog occurs:

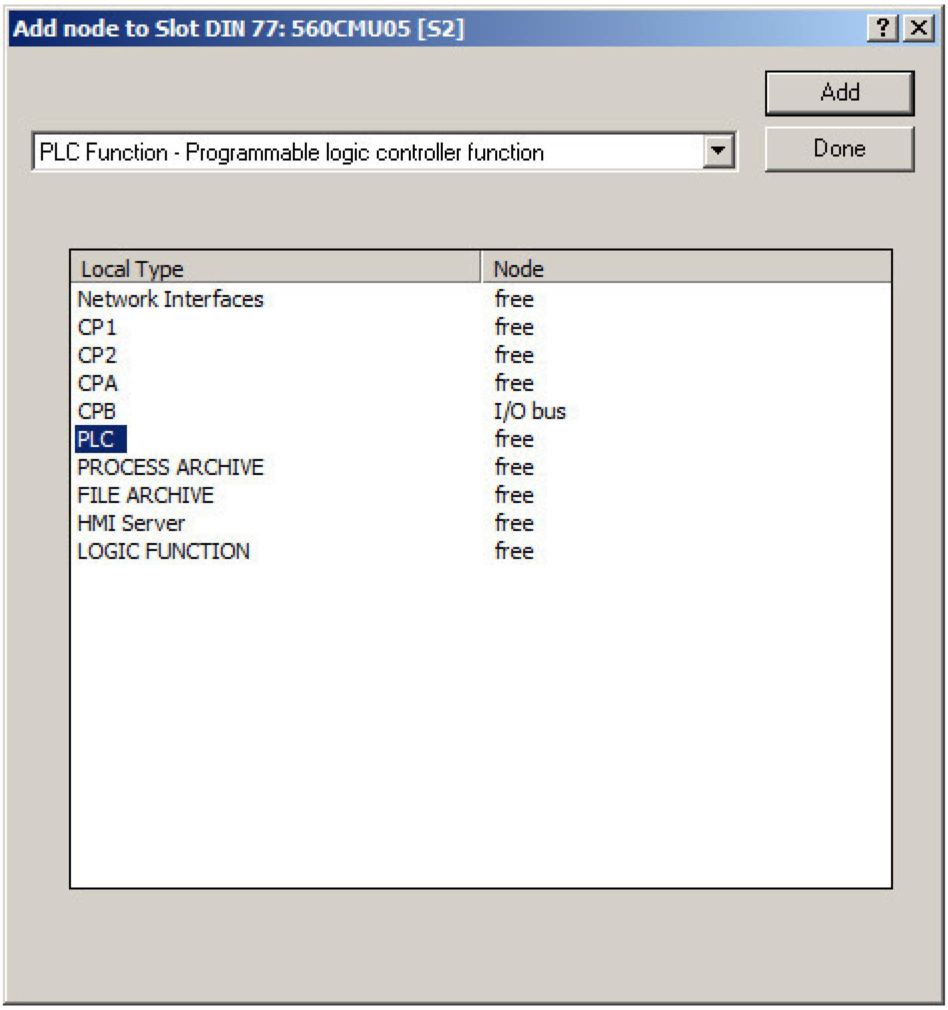


Figure 91: Add PLC function

- Select local type "PLC". When pressing "Add" the PLC function is added.

- Enter the function name on the “General” tab of the PLC function

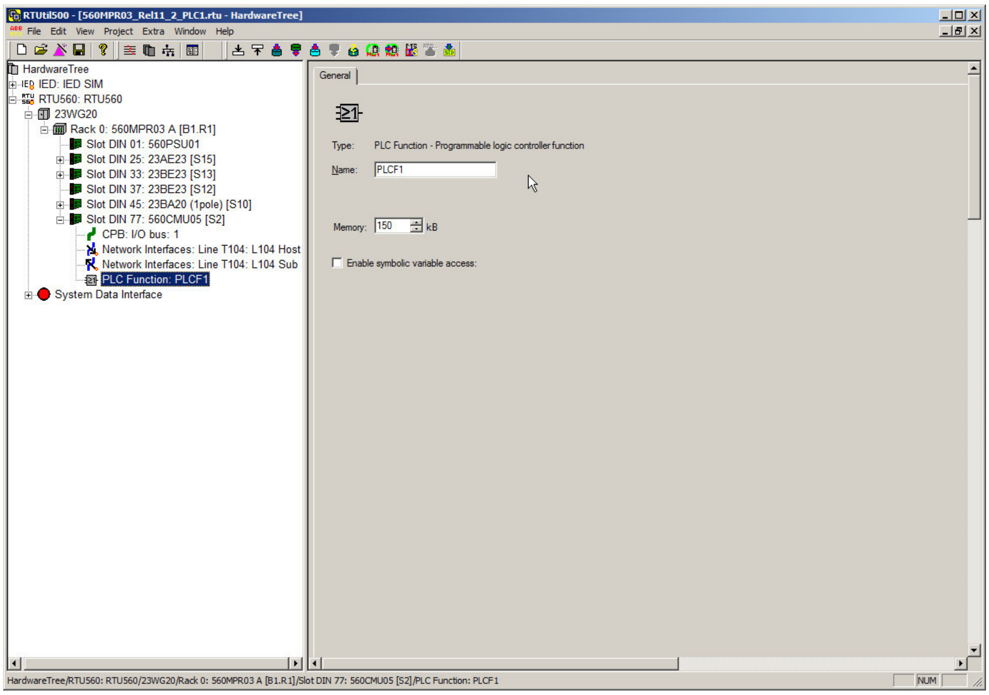


Figure 92: Configure PLC function

- Memory could be adapted in the range of 10 ... 8192kB (default 150kB) , if needed.

Insert a PLC task:

- Select the PLC function inside the HW tree and select "Add item..." inside context menu. Following dialog occurs:

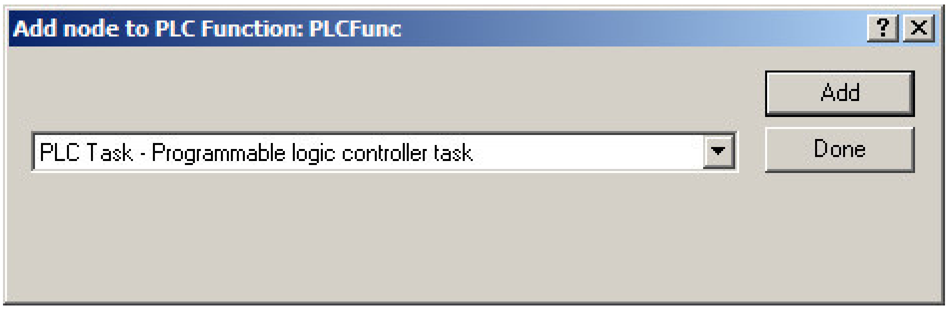


Figure 93: Add PLC task

- Up to 15 PLC tasks could be added to a PLC function
- Enter a task name on the “General” tab of the PLC task

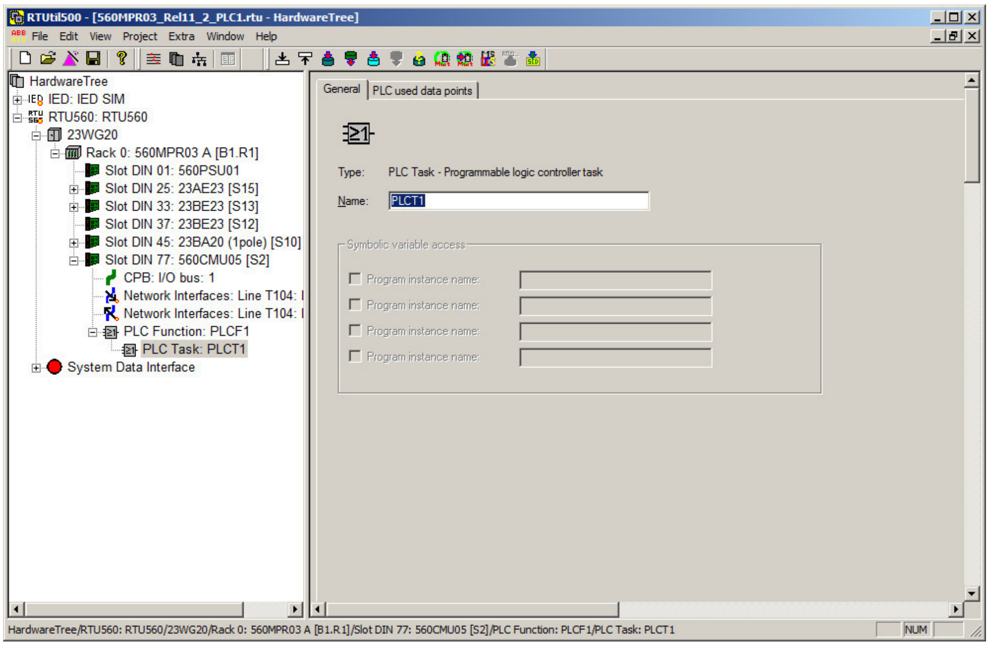


Figure 94: Configure PLC task: tab 'General'

- Select the data points to be used in the PLC function

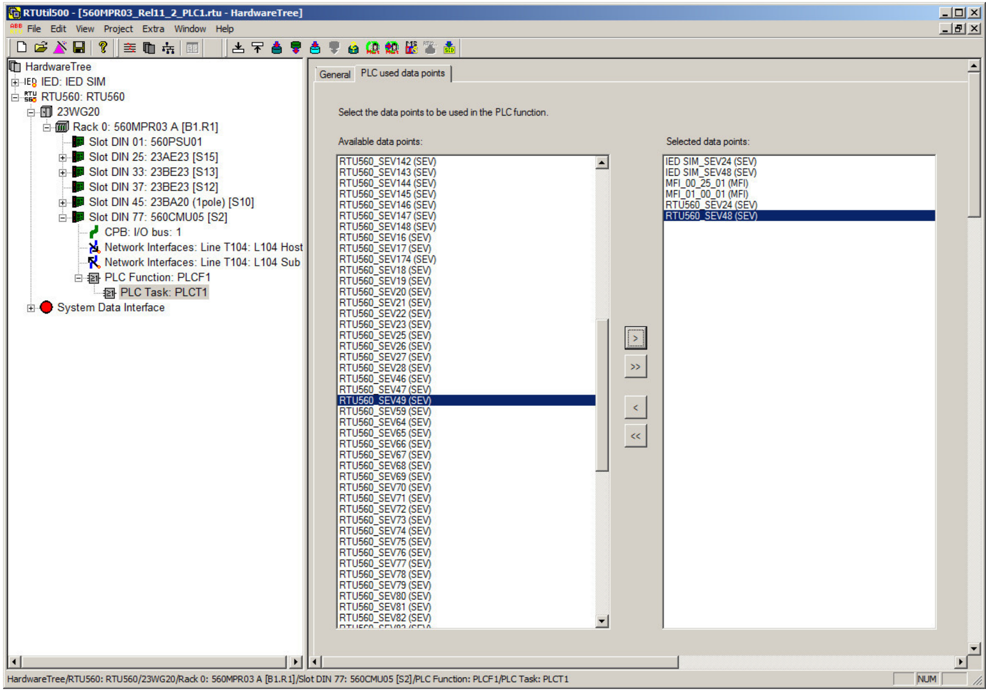


Figure 95: Configure PLC task: tab 'PLC used data points'

ADVICE

After finishing configuration of PLC function, task and used datapoints, select the menu entry **'MULTIPROG wt Export...'** of the menu bar element **'Extras'** in RTUtil500. When working with the multiprog wt project select **'File/Import...'** to run the **'RTUil NT Import'**. After that the configured functions, tasks and datapoints are available inside the multiprog wt project.

14.2 Symbolic access to PLC variables

Inside an RTUtil500 project using the PLC functionality there are some things that have to be done in addition for the usage of symbolic access to PLC variables.

Enable usage of symbolic variable access:

- Select the checkbox “Enable symbolic variable access” on the “General” tab of the PLC Function.

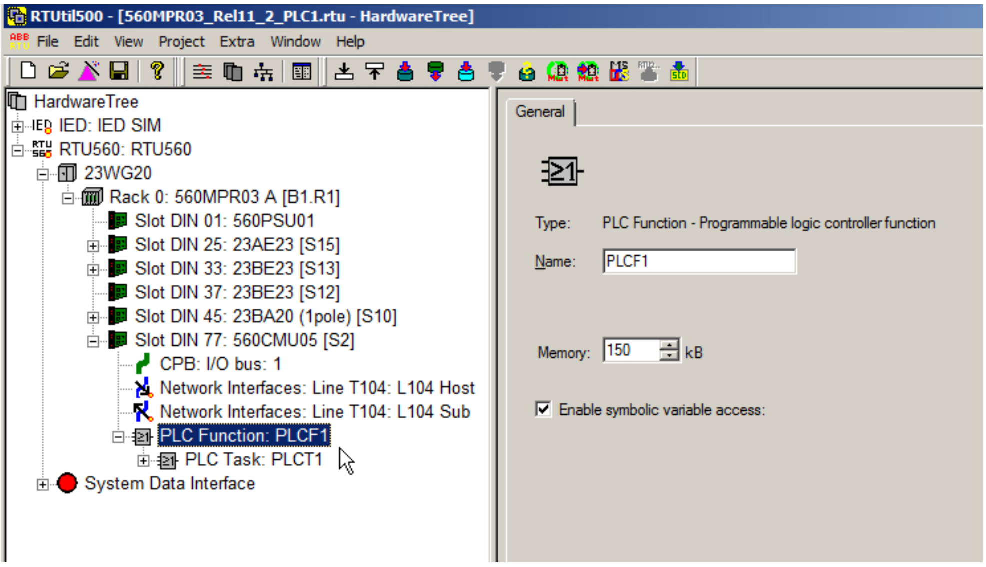


Figure 96: Enable symbolic variable access

- Enter the task name and up to 4 program instance names on the “General” tab of the PLC Task.

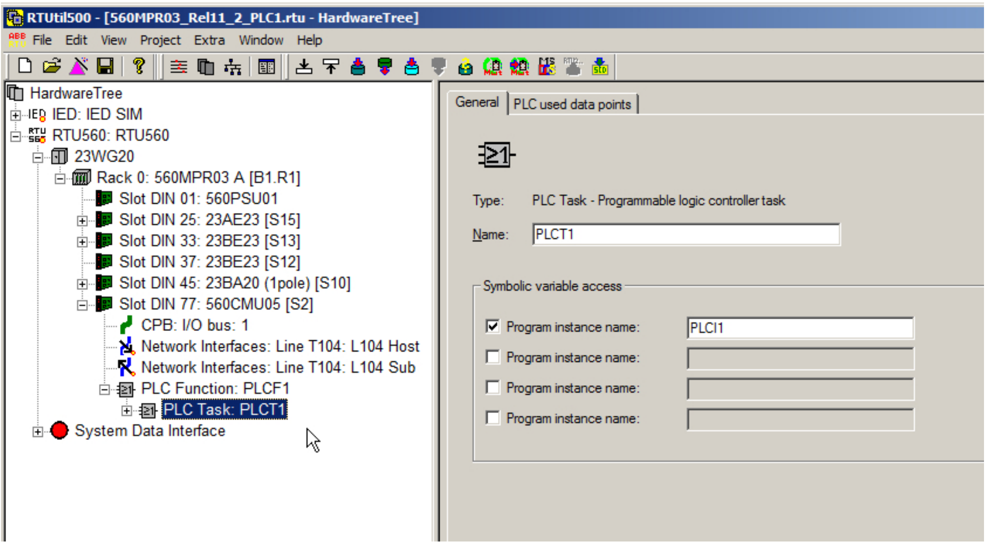


Figure 97: Configuration for PLC task

Mapping of symbolic variables to datapoint:

- Select the checkbox “Enable symbolic variable” on the “PLC:<PLC function>” tab of the data point.
- Enter a program instance name (one of the possible four instances mapped to the PLC task).

- Map the data point to a symbolic variable that is already declared inside the MULTIPROG wt project. Mapping should be done for data points of the correct type.

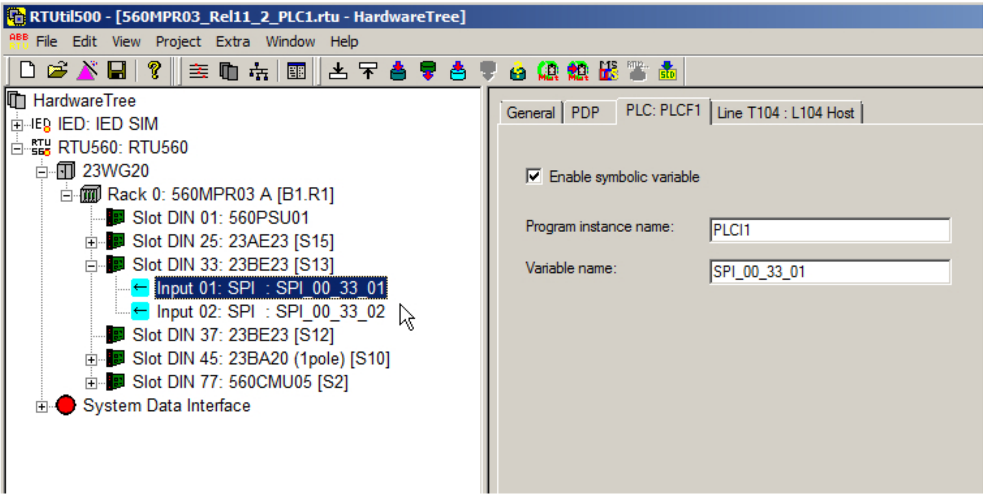


Figure 98: Map symbolic variable to datapoint

ADVICE

The selection could only be done for data points not already be chosen at the “PLC used data points” tab of the PLC task.

15 Directory Structure

15.1 Relative Directory

The relative directory structure after installation is described in the following table. The structure is user dependent and suggested by the installation program as

C:\Program Files (x86)\ABB\RTUtil500_n_m_b_d

n: Major release number

m: Minor release number

b: Build number

d: Development version number

15.2 Sub Directories

The following table contains the sub directories of RTUtil500, which will be build during the installation process.

Sub Directory	Description
\batch	Contains batch files that can be started from RTUtil500
\bin	The binary file directory for executables and dynamic link library files.
\csv_import	CSV based files for data import.
\db	Directory for function definition files and help database.
\patterns	Pattern project file. Basic project files for start.
\patterns_mwt	Pattern file of an empty MULTIPROG wt project.
\proj	Project directory for RTUtil500 projects.
\rtufile	The loadable files for the several RTU's and projects.
\usb	Contains USB RNDIS driver information file for RTU hardware

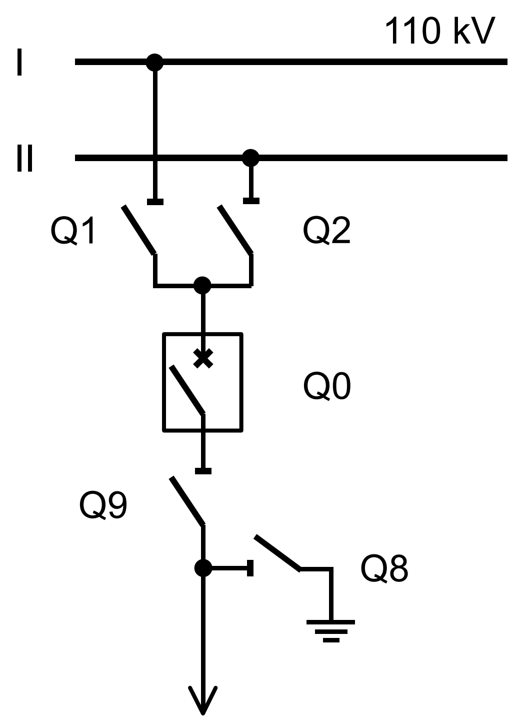
Table 3: Sub directories of RTUtil500

As default directory for MULTIPROG wt projects (*.mwt) the MULTIPROG wt project directory is taken.

16 Engineering Example

16.1 Process Control System

As process control system the following busbar with an outgoing feeder bay is given:



Q0 – circuit breaker

Q1, Q2 – busbar disconnector

Q9 – outgoing feeder disconnector

Q8 – outgoing feeder earthing switch

I, II – busbar

This bay is at two locations, in Mannheim and in Ladenburg.

16.2 RTU Configuration

16.2.1 Initialize Project

At the start of a new project some project information is set.

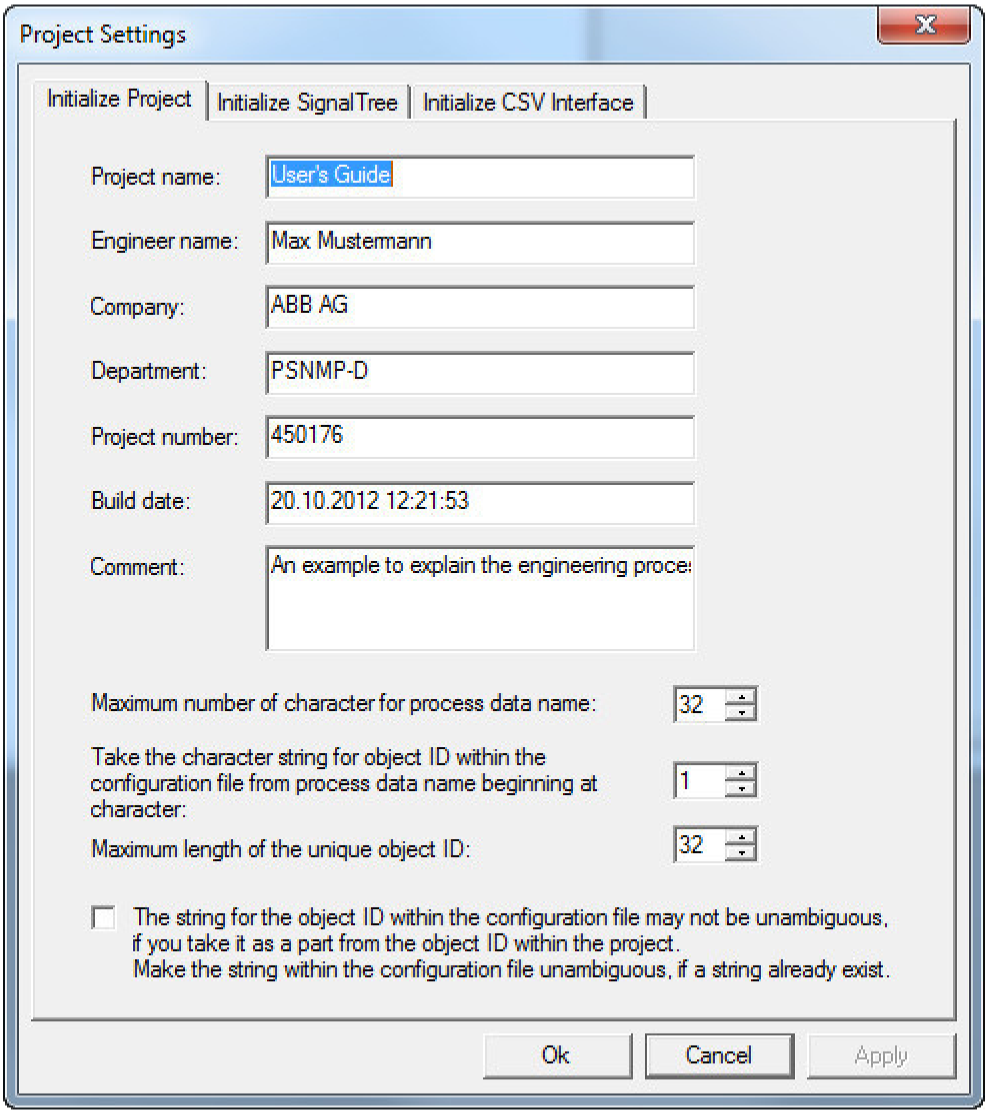


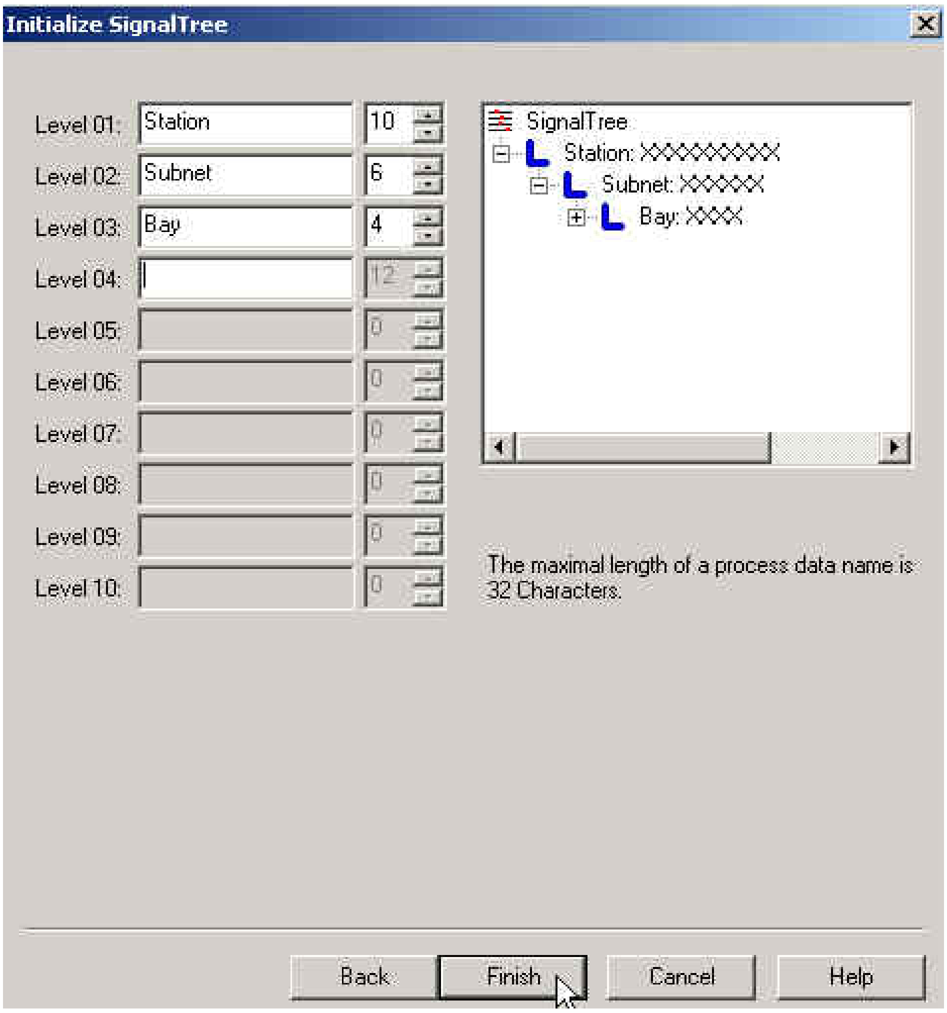
Figure 99: Initialize project

16.2.2 Initialize Signal Tree

The signal tree structure is defined. The name and the length of the levels have to be given.

Four levels are defined:

- Station with 10 character length
- Subnet with 6 character length
- Bay with 4 character length
- SCADA Object with 12 character length (the last level never has a name)



Now the general project settings are made. With 'Finish' they are accepted.

ADVICE
These settings can't be changed, if the signal tree contains an object.

16.2.3 Build the Network Tree

The two outgoing feeder bays are supervised by the telecontrol. The one in Mannheim is supervised by one RTU and the one in Ladenburg by another RTU. The RTU in Mannheim is a router RTU and connected to the control system. The RTU in Ladenburg is a sub RTU, connected to the router RTU in Mannheim.

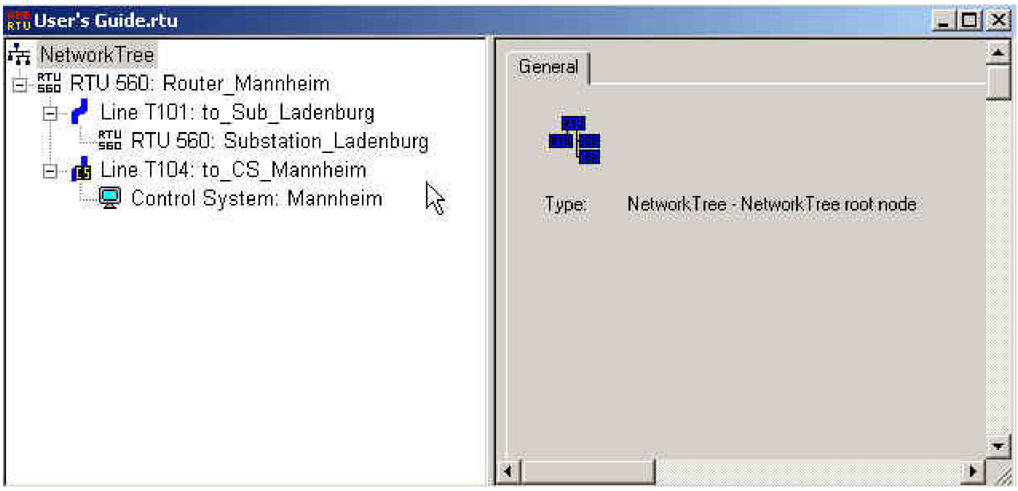


Figure 100: Network Tree

16.2.4 Build the Signal Tree

First the outgoing feeder in Mannheim is built. The outgoing feeder in Ladenburg is exactly the same. It can be copied. When the station Mannheim is copied to the signal tree the name of the station ('Mannheim') is replaced by 'COPY'. The only thing that has to be done is to replace 'COPY' by 'Ladenburg' and the signal tree is finished.

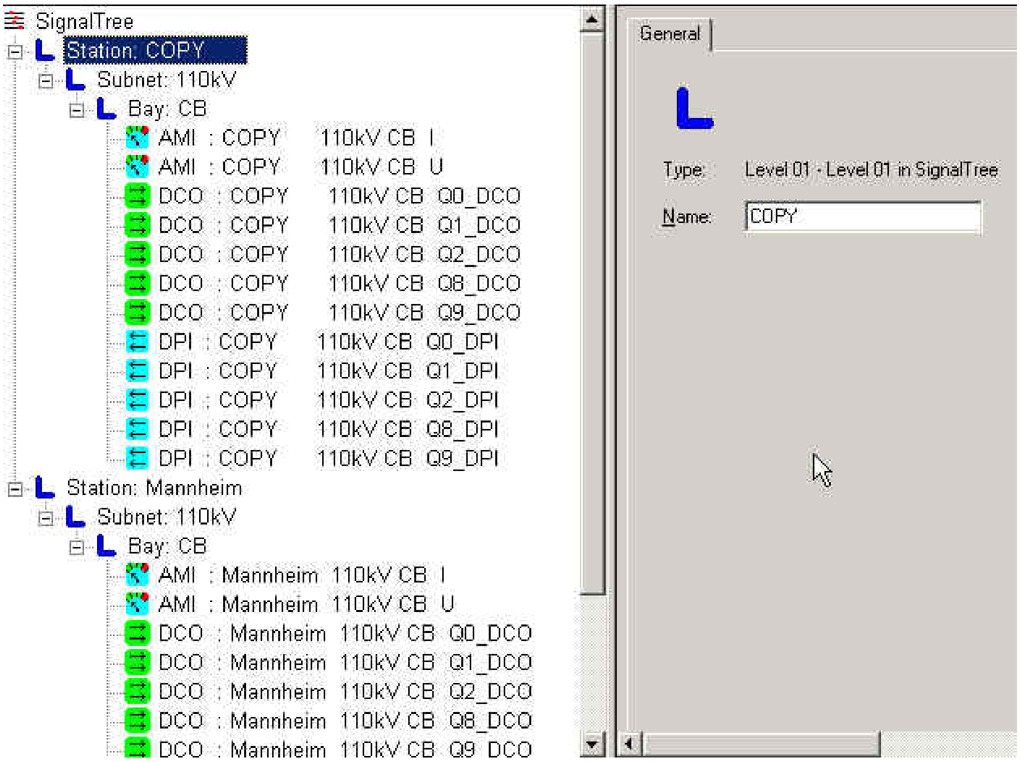


Figure 101: Signal Tree

16.2.5 Build the Hardware Tree

Communication Part

Every RTU a configuration file should be built for has to be configured in the hardware tree.

The nodes RTU, line and data point are already defined in the other trees so they have to be linked in to the hardware tree. The RTUs and the lines are in the network tree and the data point in the signal tree.

In the first step the hardware is built up to the communication rack.

The time master settings have to be done for the RTU. The parameters 'Initial redundancy mode' and 'Time administration mode' of the communication boards and the 'node name', 'IP address', 'Subnet mask', 'Default gateway IP' of the Ethernet interface have to be set.

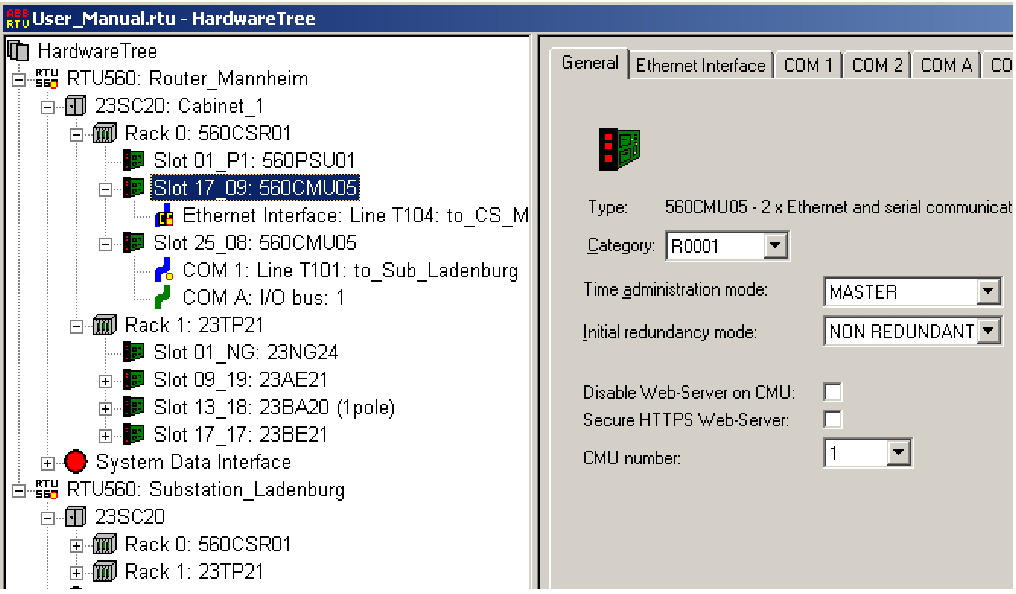


Figure 102: Parameters of the communication board

I/O Part

Then the I/O racks and I/O boards are added. They can be added only if I/O bus is defined.

Configure system data

System data are used for system signalization and system controlling. The default system data is created to a RTU or IED by RTUtil500.

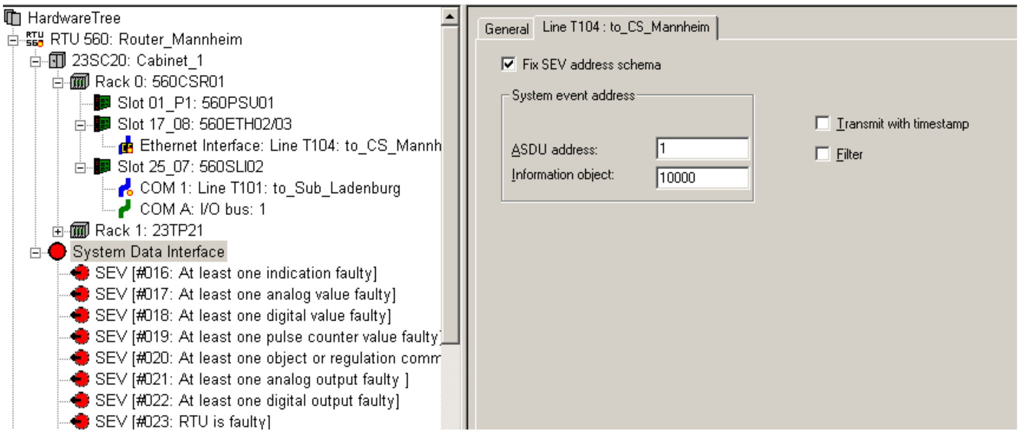


Figure 103: Address parameter of RTUs system data

Linking of Data Points

Data points added in signal tree have to be assigned to the hardware. This is done by linking them to the hardware tree.

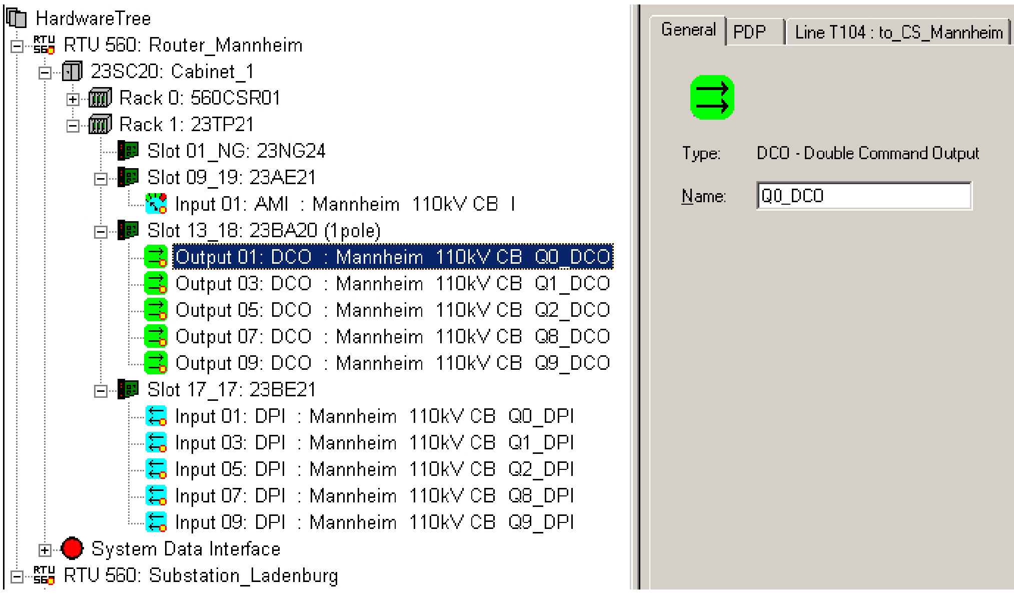


Figure 104: Hardware Tree

Now the RTUs have been engineered with the RTUtil500.

16.2.6 Configuration Files

Before the configuration files are build the project should be saved.

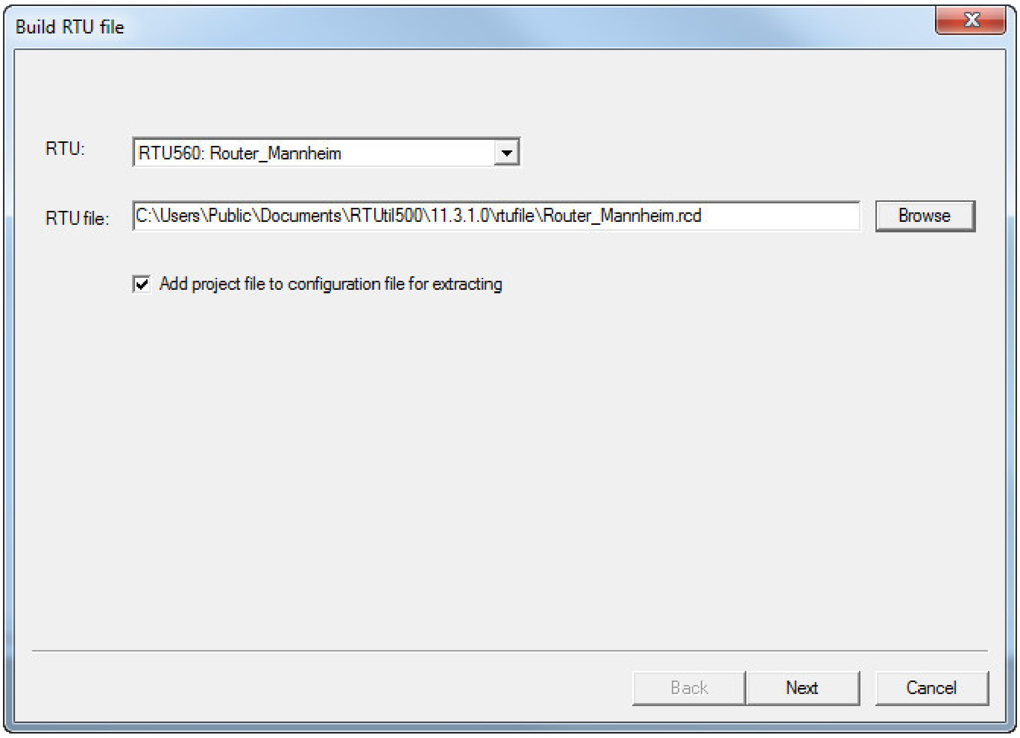


Figure 105: Generate download file

The last part (after the last „\“) of the „Filename“ is the configuration file name.

16.2.7 Configfile Download

How you can upload the configuration file to the RTU is described in the documentation: "RTU500 series Web-Server User’s Guide".

17 Wizard for DIN rail mounted RTU

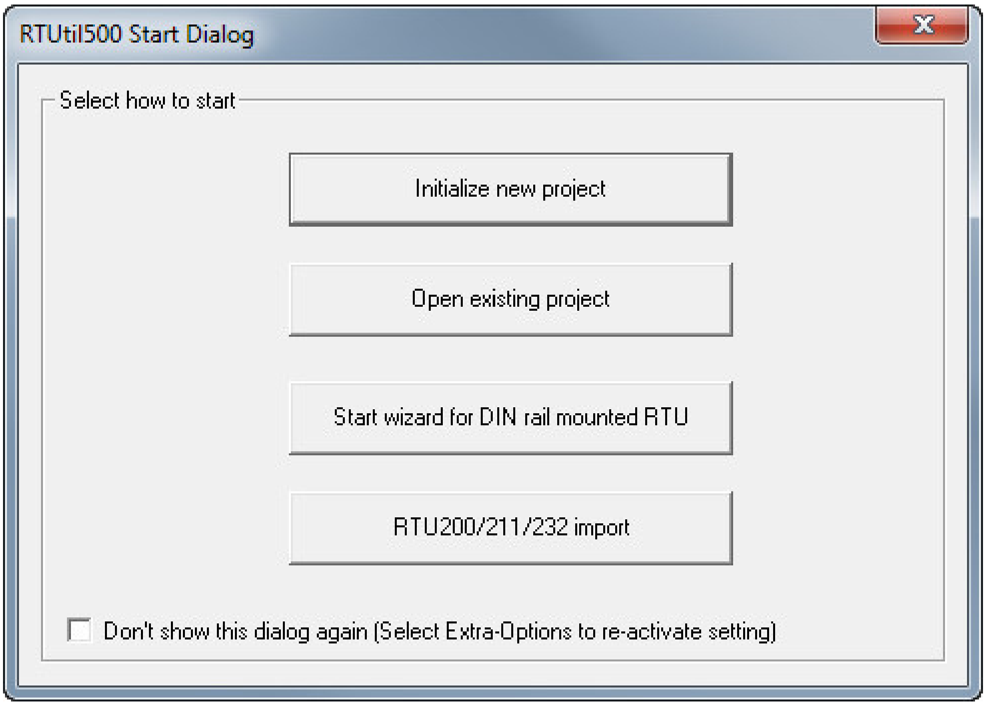
The wizard supports users in setting up DIN rail RTUs.

It offers optimized configuration for DIN rail base modules and DIN rail mounted devices. A subset of host protocols and parameters is available.

If your application requires complex configurations they can be added via RTUtil500 after the wizard is completed.

17.1 Starting the wizard

The wizard starts within RTUtil500.



It can either be started from the RTUtil500 start dialog by selecting ‘Start wizard for DIN rail - RTU’ or by selecting the icon from the toolbar.

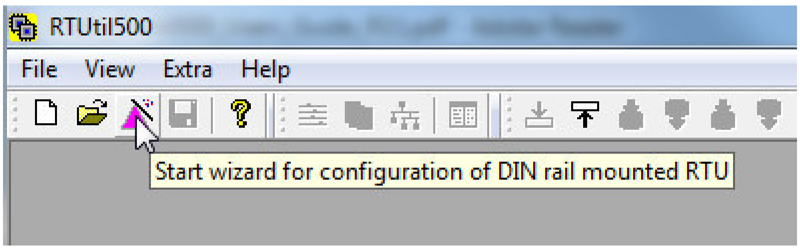
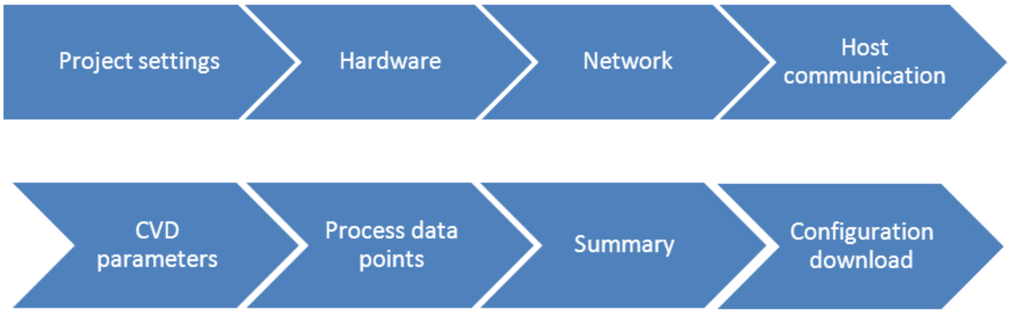


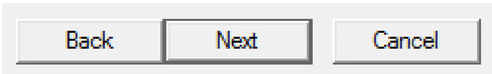
Figure 106: Toolbar icon for starting the wizard

17.2 Step by step through the wizard

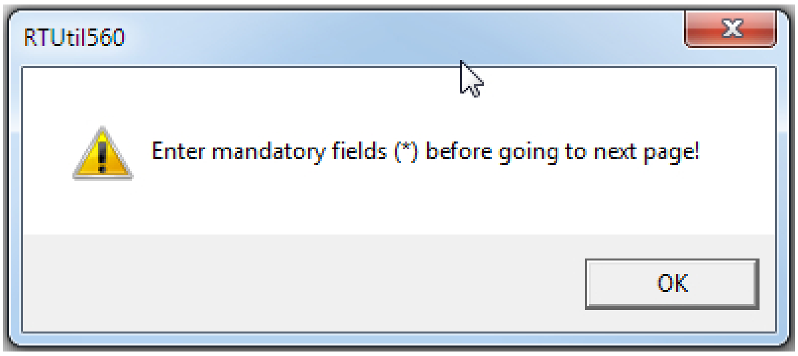
Run through the wizard by entering your configuration data: Enter hardware devices, network settings, host protocol and configure the data points as needed.



You can navigate through the wizard by clicking “next” and “back” at the right bottom of each page.



A consistency check at each page prevents you from forgetting necessary inputs.

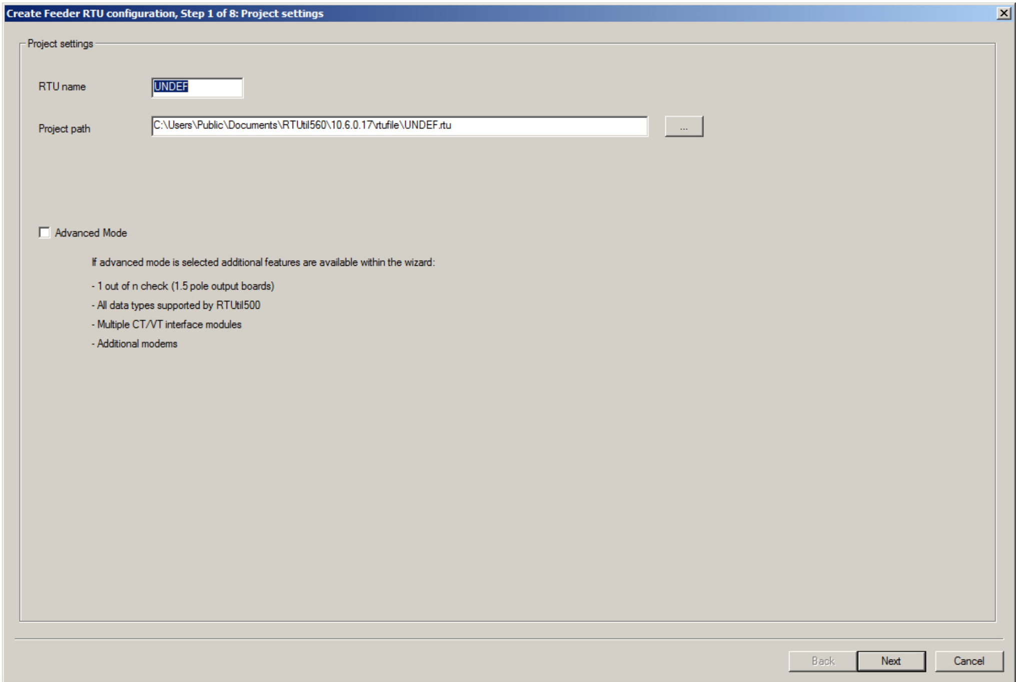


Step one is to enter a name for the RTU and choose whether the advanced mode is needed.

To enable the advanced mode, tick the check box on the start page of the wizard.

If the advanced mode is selected the following features can be configured:

- 1 out of n check (1,5 pole output boards)
- All data types supported by RTUtil500
- Multiple CT/VT interface modules
- Additional modems



In step two you can choose the used hardware devices.

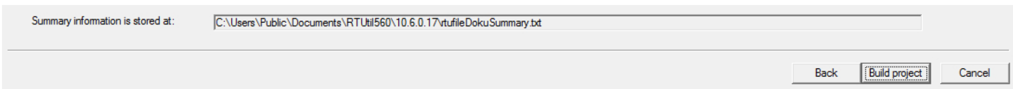
The network settings can be configured in step three.

Step four lets you select the host protocol and required parameters.

In step five the CVD parameters can be entered.

Step six requires the selection and addressing of data points.

Step seven gives you an overview about the project settings and the project will be saved. Until this point you can go back in the wizard and change your settings, clicking on “build project” means that you cannot go back anymore.



The summary information shown in this step is saved as a text file. The location where the file is saved can be found at the bottom of the page.

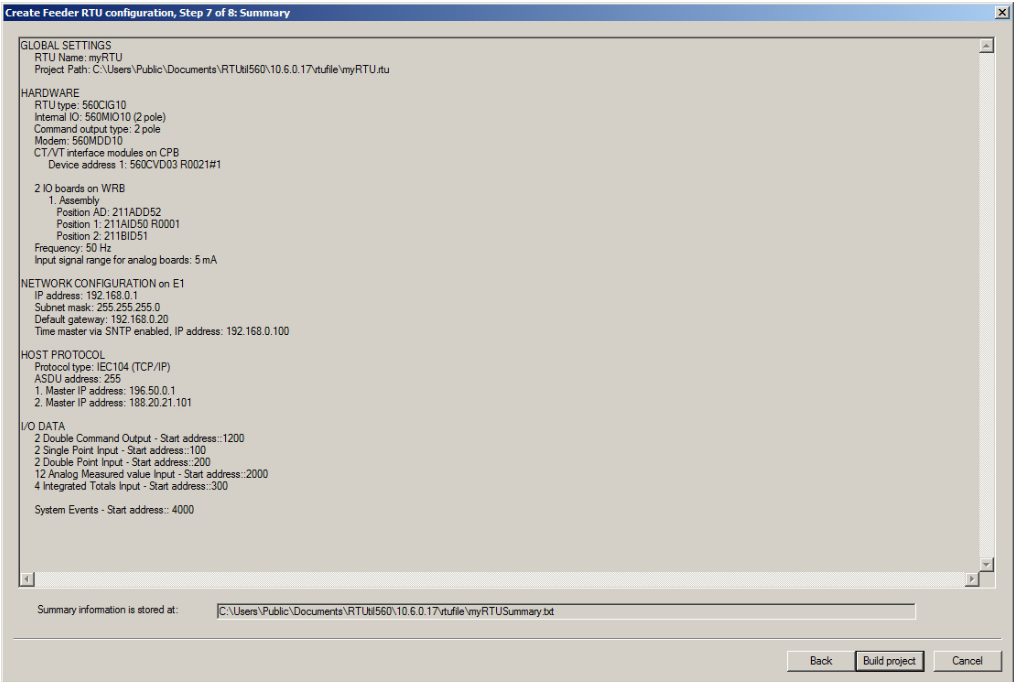


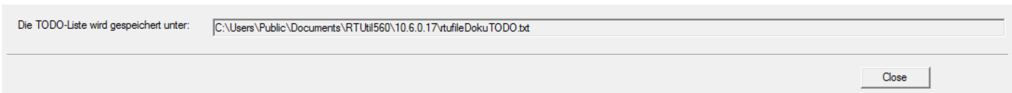
Figure 107: Summary page of the wizard

Your configuration is now ready to generate load files and download them into your RTU.

17.3 Finalizing the wizard

The final page displays a to-do list.

The to-do list will be saved as a text file and the location is displayed at the bottom of the page.



Follow the instructions step by step to configure your RTU.

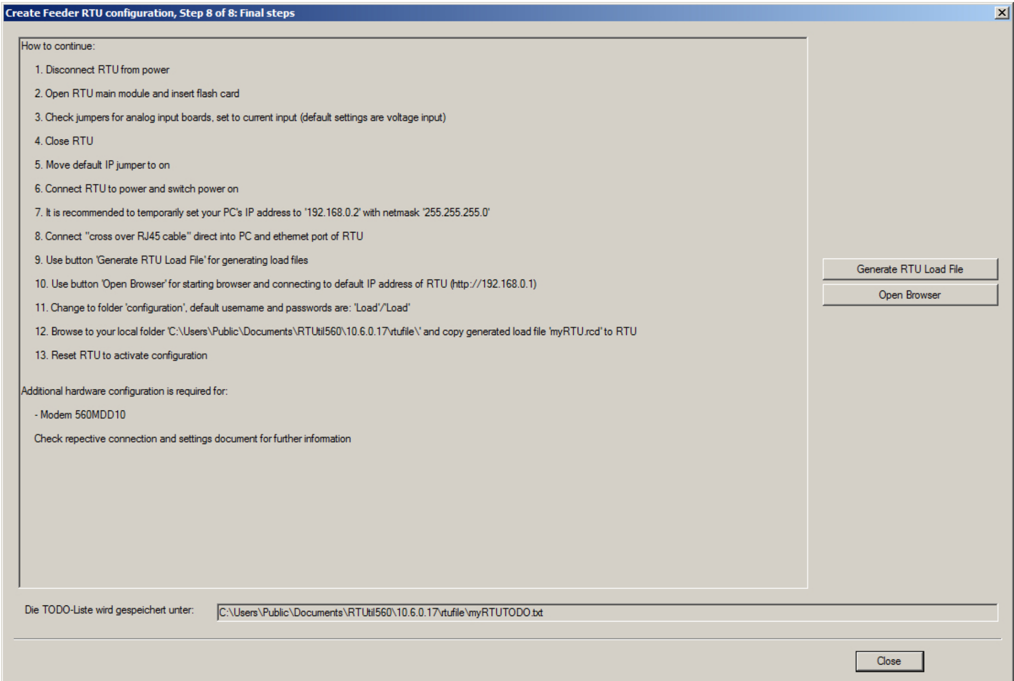


Figure 108: Figure : Final page of the wizard with todo list

Clicking on “close” closes the wizard and the configuration is opened in RTUtil500 and ready to enter more complex configurations. If you don’t need them, you can simply close the program.

Run an Excel export to get an overview of configured hardware and data points in form of an Excel table.

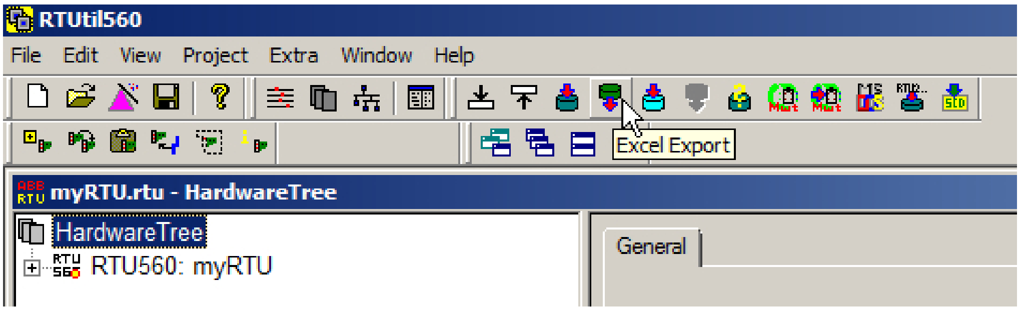


Figure 109: Run Excel export

The project cannot be re-opened in wizard mode. If you want to change your configuration or if you need additional setup you have to do it within RTUtil500.

18 Glossary

AMI	Analog Measured value Input
ASO	Analog Setpoint command Output
BSI	Bit String Input
BSO	Bit String Output
CMU	Communication and Data Processing Unit
CS	Control System
DCO	**D**ouble **C**ommand **O**utput
DMI	Digital Measured value Input (8, 16 bit)
DPI	Double Point Input
DSO	Digital Setpoint command Output (8, 16 bit)
EPI	Event of Protection equipment Input (1 bit)
FSO	Floating Setpoint Command Output
GOOSE	Generic Object Oriented Substation Event
IED	Intelligent Electronic Device
ITI	Integrated Totals Input
MFI	Analog Measured value Floating Input
NCC	Network Control Center
PDP	Process Data Processing
PLC	Programmable Logic Control
RCD	RTU Configuration Data
RCO	Regulation step Command Output
RTU	Remote Terminal Unit
SAN	singly attached PRP nodes
SCADA	Supervision, Control and Data Acquisition
SCD	Substation Configuration Description
SCL	Substation Configuration description Language
SCO	Single Command Output
SEV	System Event
SOC	Strobe Output Channel
SPI	Single Point Input or Single point information
SSC	System Single Command
STI	Step position Input

Note: