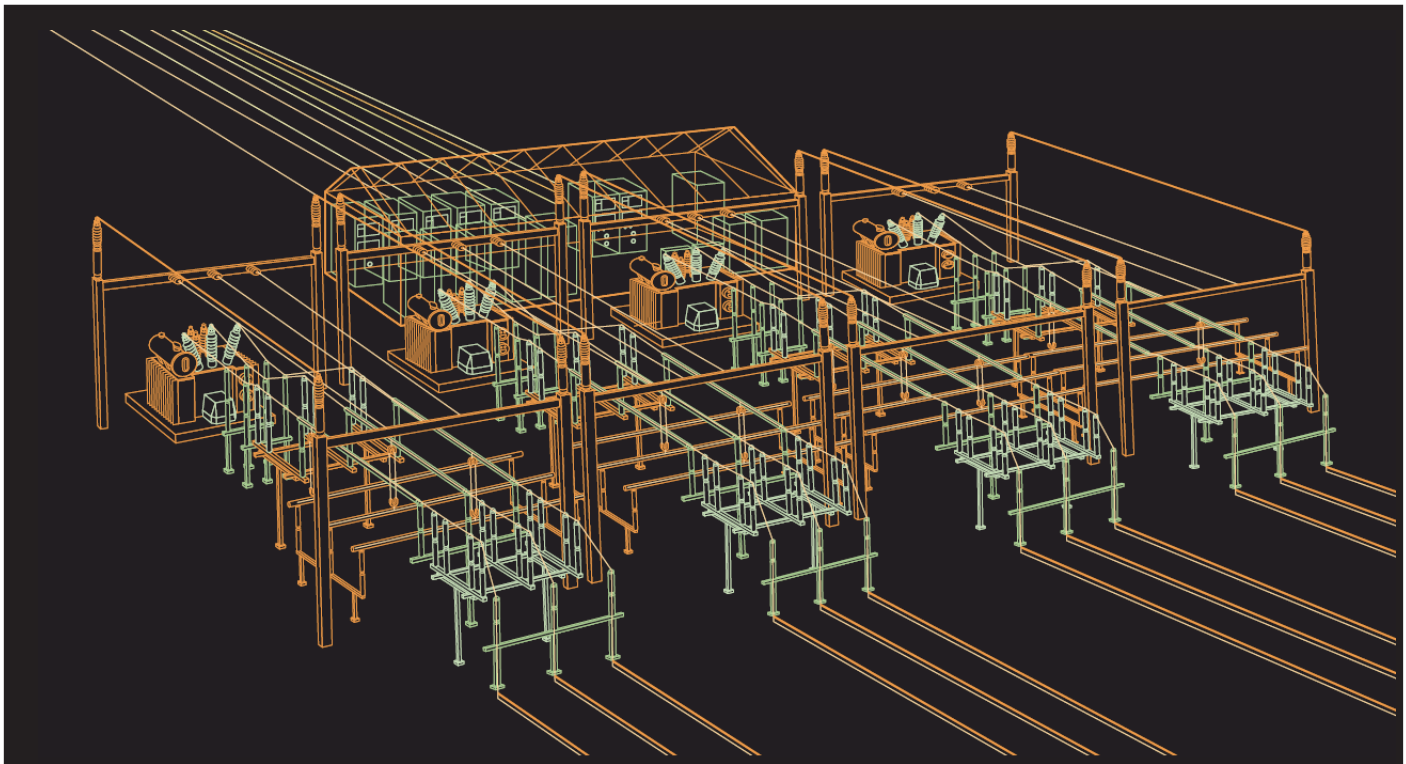
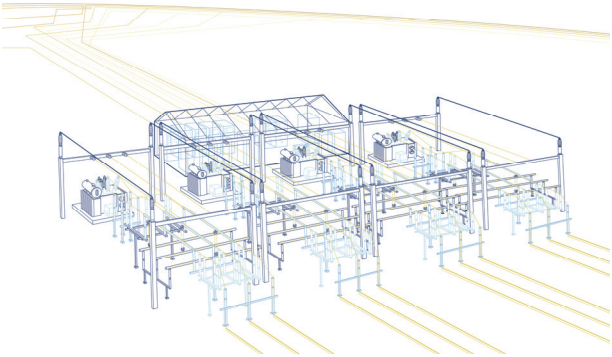


Substation Automation Products

# IET600 Integrated engineering tool

## User manual

Power and productivity
for a better world™



Document ID: 1MRK500097-UEN

Issued: October 2012

Revision: A

Version: IET600 v.5.2



**IET600 Integrated engineering tool**

User manual

# Copyright



**Guarantee**
Please inquire about the terms of guarantee from your nearest ABB representative.

**Important notice**
Experience has shown that reliable operation of our products is assured, providing the information and recommendations contained in these Operating Instructions are adhered to.

It is scarcely possible for the instructions to cover every eventuality that can occur when using technical devices and systems. We would therefore request the user to notify us directly or our agent of any unusual observations or instances, in which these instructions provide no or insufficient information.

In addition to these instructions, any applicable local regulations and safety procedures must always be strictly observed both when connecting up and commissioning this equipment.

Any work such as insertion or removal of soldered jumpers or setting resistors, which may be necessary, may only be performed by appropriately qualified personnel.

We expressly accept no responsibility or liability for any damage, which may result from operation of this equipment, even if no reference is made to the particular situation in the Operating Instructions.

ABB AB

Substation Automation Products

SE-721 59 Västerås

Sweden

Telephone: +46 (0) 21 34 20 00

Facsimile: +46 (0) 21 14 69 18

http://www.abb.com/substationautomation

# Disclaimer

This data, examples and diagrams in this manual are included solely for the concept or product description and are not to be deemed as a statement of guarantee properties. All persons responsible for applying the equipment addressed in this manual must satisfy themselves that each intended application is suitable and acceptable, including that any applicable safety or other operational requirements are complied with. In particular, any risks in applications where a system failure and/or product failure would create a risk for harm to property or persons (including but not limited to personal injuries or death) shall be the sole responsibility of the person or entity, applying the equipment, and those so responsible are hereby requested to ensure that all measures are taken to exclude or mitigate such risks.

This document has been carefully checked by ABB but deviations cannot be completely ruled out. In case any errors are detected, the reader is kindly requested to notify the manufacturer. Other than under explicit contractual commitments in no event shall ABB be responsible or liable for any loss or damage resulting from the use of this manual or the application of the equipment.

# Table of contents

# 1 Glossary of Terms

| Term | Description |
|---|---|
| CID-file | Configured IED description. According to IEC 61850, this file transports Data from the IED configuration tool to the IED. In the current engineering context, it is also used do designate a file containing a configured IED description to be imported into IET NG. In IEC 61850, V2, there is another file specified for this process, the IID-file. |
| HMI Signal | corresponds to one line in an event or alarm list; or one process information. Data point is the usual terminology when talking with a customer, exchanging lists with him etc. |
| IED | Intelligent Electronic Device, general term in IEC 61850 for everything programmable and electronic (e.g. HMIs or NCC gateways are also IEDs, not only control and protection devices). |
| ICD-file | IED Capability Description file. |
| IID-file | Instantiated IED Description file (IEC 61850, V2). Describes the project-specific configuration of an IED. |
| LN | Logical Node, in IEC 61850 a container for all data related to one specific function (e.g. PTOC for time-delayed overcurrent protection, or CSWI for Switch control). LNs are product-specific. |
| LNode | Logical Node Reference in a Substation Tree element. It can either be used as a standalone name to specify some desired functionality in an SSD-file, or it can connect an LN in an IED to an object of the Substation Tree. The latter is helpful to:<br>• attach a functionality to a substation object (e.g. a protection function PTOC to a bay or a control function CSWI to a specific circuit breaker or switch).<br>• distinguish functionalities (e.g. Protection Trip from a Main Protection Device or from a Backup Protection device).<br>• support automated naming in the HMI according to the Substation Tree. |
| SCD-file | Substation Configuration Description file. According to IEC 61850, this file contains all IEDs, a communication |

| | |
|---|---|
| | configuration section and a substation description section. |
| SCL-file | Substation Configuration Language file. General term for a file that is written in SCL. Includes all other files (SCD, SSD etc.) |
| Substation Tree | The tree with one Substation as its root, containing all the primary equipment for this substation (Voltage levels, Bays, Circuit Breakers, Disconnectors etc.) in a hierarchy defined by IEC 61850. |

# 2 Installation

## 2.1 System requirements

### 2.1.1 HW requirements

The minimum hardware requirements are:

- 300 MB of free hard disk space
- Dual-core processor
- 3 GB RAM

The recommended hardware requirements for medium to big projects are:

- 300 MB of free hard disk space
- 64bit operating system
- Quad-core processor
- 8 GB RAM
- SSD recommended for system drive

### 2.1.2 Operating system

The following operating systems are supported:

- Windows XP SP3
- Windows 2003 Server SP2
- Windows Vista SP1
- Windows 7
- Windows 2008 Server

### 2.1.3 Additional software components

IET600 needs some additional software components (these are contained in the IET Prerequisites package, see below):

- Windows Installer 4.5
- .NET Framework 4.0
- SQL Server 2008 SP1 (requires Service Pack 3 on Windows XP)

## 2.2 Installation Procedure

The IET600 installation consists of two installation packages:

(1) IET600 Prerequisites

- SQL Server 2008 SP2 (Note: Requires Service Pack 3 on Windows XP)
- Windows Installer 4.5
- .NET Framework 4.0

(2) IET600 Baseline

- IET600 Application
- Required runtime components

The IET600 Prerequisites must be installed before the IET600 Baseline installation. Re-installation of the IET600 Prerequisites may not be required for each successive IET600 Baseline installation.

### 2.2.1 IET600 Prerequisites Installation

Run the IET600 Installer Package and follow the InstallShield Wizard to complete the IET600 installation. The following steps must be fulfilled:

- Start IET600 Prerequisites InstallShield
- SQL Server – Specify the root directory for the IETSERVER
- Start installing the program

The IET600 Prerequisites are now installed in the folder. Proceed to the IET600 Baseline installation, as described in the next section.

### 2.2.2 IET600 Baseline Installation

Run the IET600 Installer Package and follow the InstallShield Wizard to complete the IET600 installation. The following steps must be fulfilled:

- Start InstallShield
- License Agreement – Read and accept the terms of the license agreement
- Customer Information – Enter User Name and Organization into corresponding field and specify if the license is to be designated for the specified user only or all users on the computer
- Destination Folder – Choose the default folder (C:\Program Files\ABB\) or specify another destination folder
- Click “Install” to start installation

IET600 is now installed in the Destination folder. A Shortcut is also available on the Desktop to start the program.

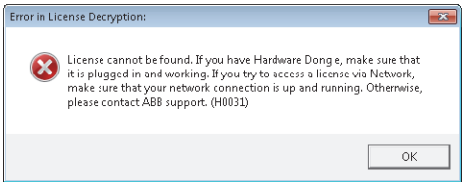
## 2.3 Getting a License for IET600

IET600 needs a License to run.

When you obtained IET600, you should have received a Dongle that contains the License:

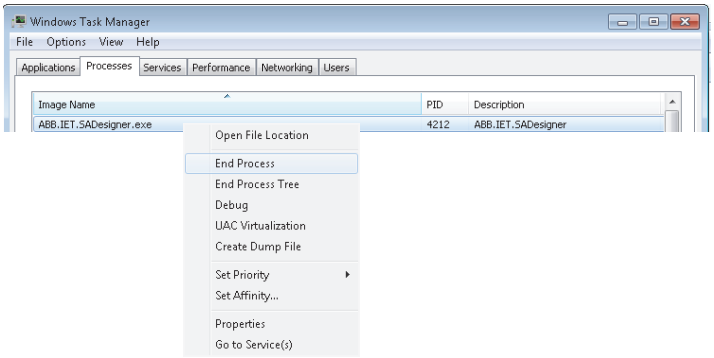
Connect this Dongle to a USB-Port of the PC on which you have IET600 installed (all necessary software components, drivers etc. have been installed along with IET600). It may take the system up to 10 seconds to recognize the License; after that time you can start IET600 and use it.

If you try to start IET600 and no license is found on your PC, you may get the following error message:




In this case, the Dongle has either not been plugged in or has not been recognized. If possible, correct the problem and click OK.

If you could correct the problem, click on OK; IET600 should start. Unfortunately, if you cannot correct the problem, if you click OK, the Dialog will reappear again immediately. There is no other remedy than to open the Task Manager, locate the process ‘ABB.IET.SADesigner.exe’ and kill the process (right-click -> “End Process”).




## 2.4 Updating IET600

### 2.4.1 Overview

An IET installation package has the version name included, e.g. ABB IET600 SAS Setup - Baseline 5.2.16.zip.

To allow you to easily check for available software updates, IET600 provides a built-in service which is described in this chapter.

### 2.4.2 Update Concept

We distinguish between:

- Bugfixes
- Updates

IET600 has an inbuilt check whether any of the above updates are available (see chapter 2.4.5, Checking for Updates).

### 2.4.3 Bugfixes

If the version differs only in the 4th digit (e.g. 5.2.16.2 against 5.2.16.1), it is a “bugfix”.

As the older version is considered to be buggy and should not be used in future, a bug fix version will overwrite the corresponding older version with the same 1st, 2nd and 3rd digit (e.g. 5.2.16.2 will overwrite 5.2.16.1, but not 5.2.15.0).

It is recommended to always install such “bugfix” versions. They do not require a new license.

### 2.4.4 Updates

See chapter 2.4.5 (Checking for Updates) how to find out whether Updates are available.

Versions that differ in the 3rd digit (e.g. 5.2.16 against 5.2.15) will typically contain such improvements or additional features, but no major changes.

Such a version will be installed in parallel to existing versions; i.e. existing versions will not be removed. However, if you open a project in the newer version, it may require a migration after which a project cannot be opened in older versions. It is strongly recommended to always open a project in the older version one more time and make a backup before opening it in the new version.

Changes in the operation of IET600 in such versions are typically slight and can be understood intuitively, no re-training is required.

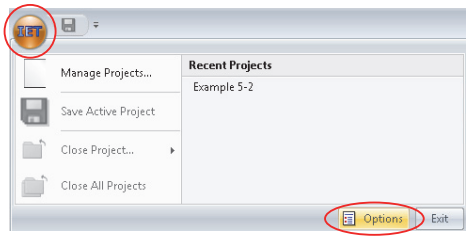
While such versions are tested against common project configurations, they very occasionally introduce a bug or change a behaviour in a way that is unexpected to the engineer. It is therefore recommended to delay an update when in critical periods of engineering; otherwise an update is recommended and uncritical as you can always go back to using an older version installed in parallel.

If the version differs in the 2nd digit (e.g. 5.3 against 5.2), it will contain major new features. Usually a training will be required for users to familiarize themselves with new editors, concepts or engineering processes.

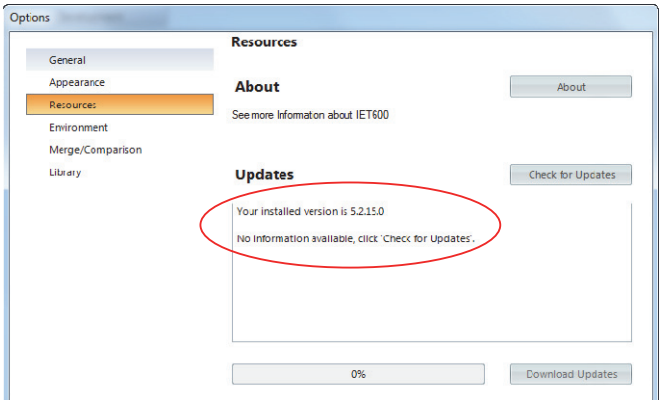
Such a version will be installed in parallel to existing versions; i.e. existing versions will not be removed. If you open a project in the newer version, typically a migration will be required, after which a project cannot be opened in older versions. It is strongly recommended to always open a project in the older version one more time and make a backup before opening it in the new version.

## 2.4.5 Checking for Updates

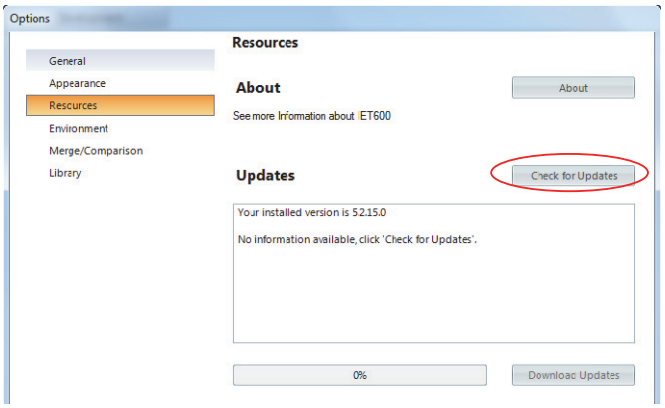
1. From the Application Menu, choose Options:



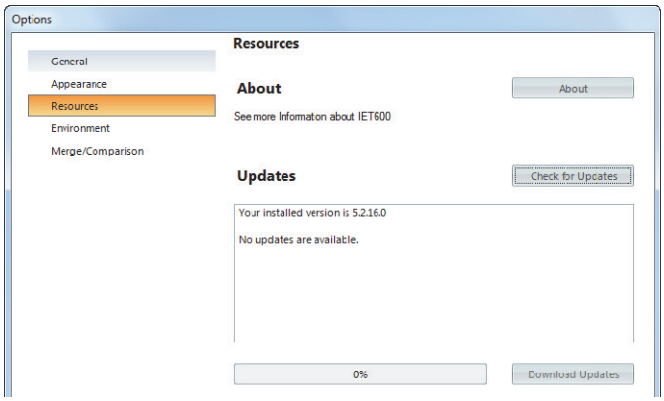

2. Open the “Resources” tab. It will show you the current IET600 version.




3. To see whether updates are available, click on ‘Check for Updates’:



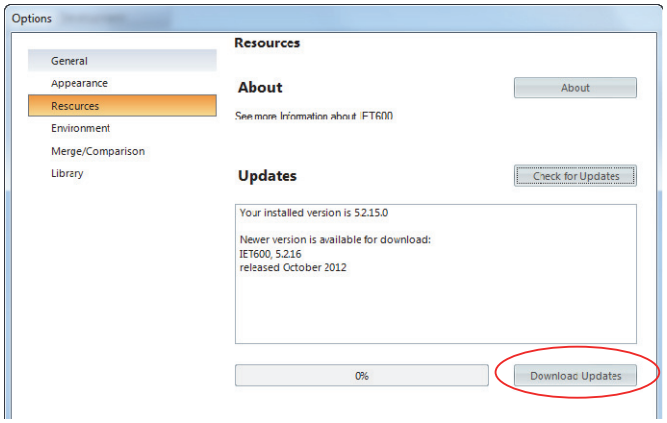
4. Either you will be informed that no updates are available:



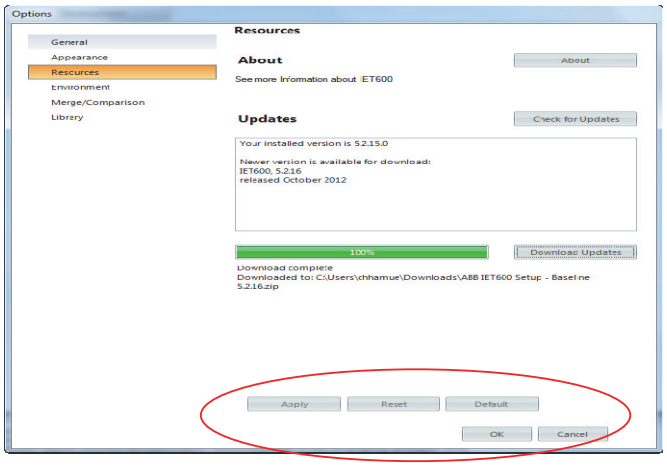

Or you will see available updates, as shown below.

If this update is available to you free of charge (which implies that it can be used with the same license), the 'Download Updates' Button will become enabled.

If you need to acquire a new license (for which you need to pay), you will be informed accordingly. In this case, the 'Download Updates' Button will not be enabled, you will receive your software through other channels.




5. If you click on 'Download Updates', you will be offered a Zip-file for download and you can select where to save it. A successful download will be confirmed:



6. Now you can install the downloaded Zip file as described in chapter 2.2 (Installation Procedure). Normally, no new prerequisites should be needed, otherwise you will be informed accordingly in step 4 above.

## 2.5 Removing older IET600 versions

An IET600 version can be removed by:

1. Re-running the Installer Package (not recommended).
2. Via the Control Panel.

Uninstalling an IET600 version removes only this particular version. Other versions as well as data such as IET projects, project backups, licenses etc. will not be removed.

### 2.5.1 Remove IET600 via Control Panel

1. From the OS Main Menu, open the Control Panel.
2. Open the appropriate system tool:
    - ‘Add/Remove Programs’ in XP.
    - ‘Programs and Features’ in Windows 7.
3. Locate the IET600 version you want to remove and open it by double-clicking.
4. A confirmation dialog will appear. Confirm appropriately for removal.
5. The IET600 version will be removed.

### 2.5.2 IET600 Prerequisites De-Installation

The Prerequisites can be removed either by

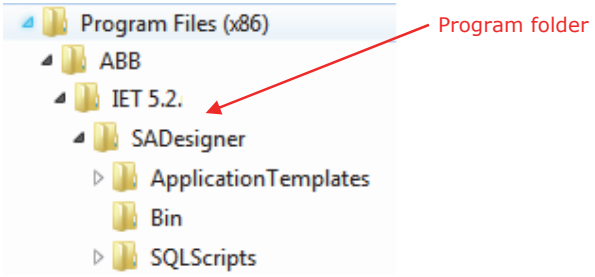
1. Re-running the Installer Package.
2. Using the ‘Add or Remove Programs’ dialog in Control Panel.

Follow the steps in the InstallShield to complete the remove.

Uninstalling the Prerequisites removes only the components installed by the Prerequisites setup. The installed Microsoft packages, such as SQL Server etc. will not be removed.

## 2.6 Folder Structure

The IET600 program folder can be found under Program Files > ABB, as shown below:


Program Files (x86)
ABB
IET 5.2.
SADesigner
ApplicationTemplates
Bin
SQLScripts
Program folder

# 3 Introduction

## 3.1 Overview

ABB's Integrated Engineering Tool is a system engineering tool for IEC 61850-based communication networks and substation automation systems (SAS).

IET600 enables a simplified, consistent and flexible approach to SA system engineering, thanks to:

- support of the IEC 61850 standard
- consistent and efficient system-wide data engineering and communication configuration
- easy reuse of engineering data
- the use of the standardized IEC 61850 data format, the System Configuration Language (SCL), for exchanging data with any 3rd party engineering tools
- integrated support for SYS600 database engineering

The core of the IET600 platform is the Project Database. The database contains all project data and is the common data source for other IET600 modules. The single source database ensures that data consistency is preserved through various engineering workflows.

IET600 contains various modules to complete the system engineering of an IEC61850-based substation, including:

- configuration of the substation topology
- configuration of the communication network
- configuration of the IEC61850 dataflow
- import and export of IEC-61850-SCL data for exchange with other tools
- export of project data for documentation

# 4 User Interface

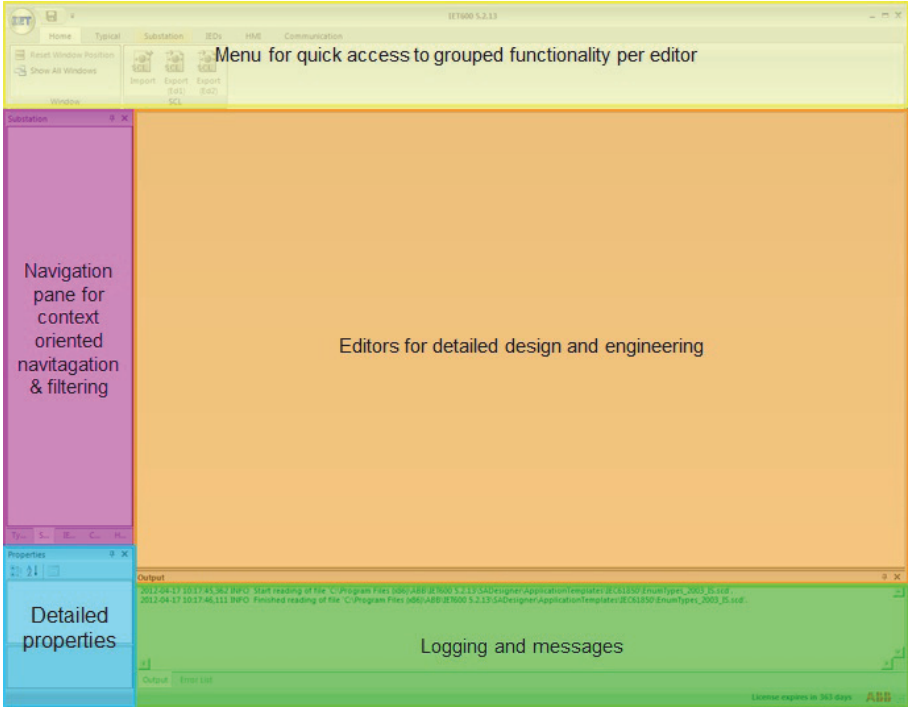
## 4.1 Overview

IET600 provides one user interface from which the user can access all project data as well as all functions. The user interface provides multiple navigation and display options to support the engineering process, combining the specification of the substation topology as well as the IEC 61850 communication in one common interface.

## 4.2 IET600 User Interface

The IET600 user interface is divided into distinct sections or “panes” for navigation and for the display of project data:

- The Menu on the top of the user interface for quick access to grouped functionalities
- The Navigation pane on the left side for context oriented navigation and filtering for the various Editors
- The Editors for detailed design and engineering of the substation, IEDs and IEC 61850 communication
- The Properties pane for detailed properties of the selected element in the Navigation
- The Logging & Messaging pane to support analysis and trouble-shooting

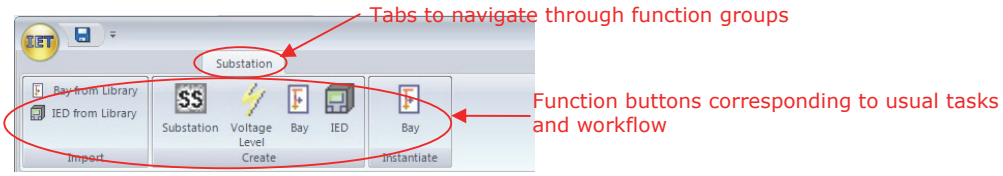



## 4.2.1 Menu

The Menu pane allows the user to quickly access relevant functions for the engineering task at hand. These functions are organized in a series of tabs, each related to a particular task, including

- Home – adjust windows, print, export SCL file
- Substation – import, build and instantiate full substation topology
- IEDs – import, build and configure IEDs
- Communication – build subnetworks, create RCB and GCB datasets, configure subnetworks
- HMI – Create station IEDs, import/export and configure HMI database and signal lists
- Grid Editor – Import/export entries into grid Editor

Using Menus provides the user with an intuitive guide through the engineering workflow. The relevant context view in the Navigation panel is displayed and the usual engineering steps are easily accessible across the Menu pane from left to right.



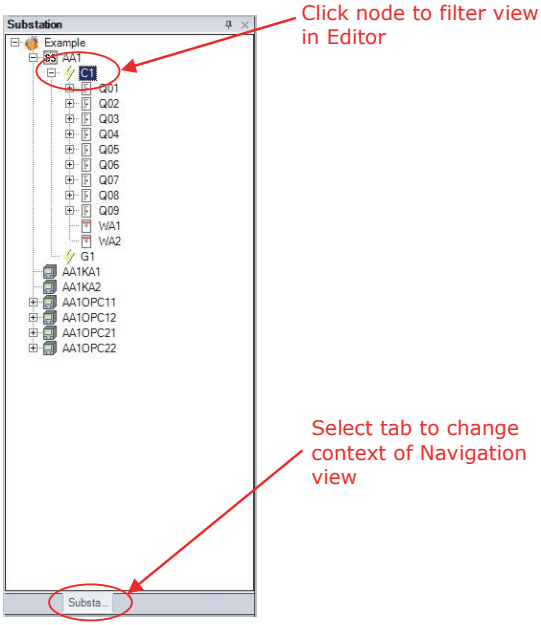
## 4.2.2 Navigation

The Navigation pane provides context oriented navigation of the Editors. The Navigation structure displays a hierarchal view of the substation element or nodes. Clicking on a particular node in the Navigation filters the data in the Editor pane. The top-most node is the project node, which shows all data in the Editor pane, without filtering.

The Navigation pane itself has five tabs, which corresponds to five different context views:

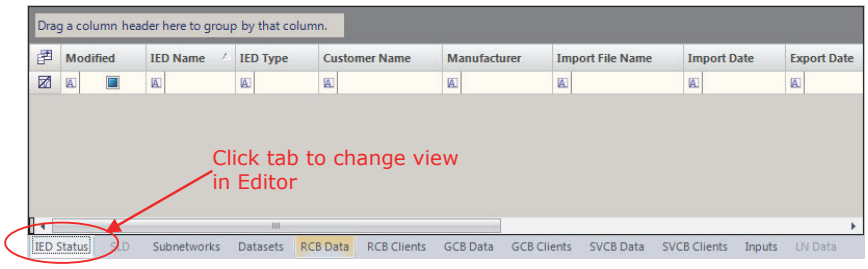
- Substation – Full substation topology and primary equipment nodes
- IEDs – IEDs nodes and corresponding functionality
- Communication – Subnetworks and connected IED access points
- HMI – Station IEDs nodes

Changing the tabs allow the user to change the context views in the Navigation structure.




## 4.2.3 Editors

The Editor pane is the main working area of the IET600 user interface. It is organized in various tabs for detailed substation design and engineering. Each tab displays a specific Editor layout (tabular and graphical) to perform detailed engineering tasks.

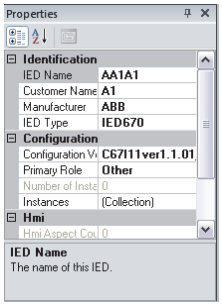



Each editor provides detailed views related to various engineering tasks:

| Editor | Description |
|---|---|
| Signal Clients | Assign HMI signals to HMI and GW clients |
| HMI Data | Configure HMI signal properties for MicroSCADA SYS600 |
| NCC Data | Configure NCC signal properties for COM500 |
| Dataset Editor | Configure dataset properties and content |
| RCB Editor | Configure RCB properties |
| RCB Clients | Assign client IEDs to RCB |
| GCB Editors | Configure GCB properties |
| GCB Clients | Assign client IEDs to GCB |
| SVCB Editors | Configure SVCB properties |
| SVCB Clients | Assign client IEDs to SVCB |
| Input Section | View and edit IED input section |

## 4.2.4 Properties

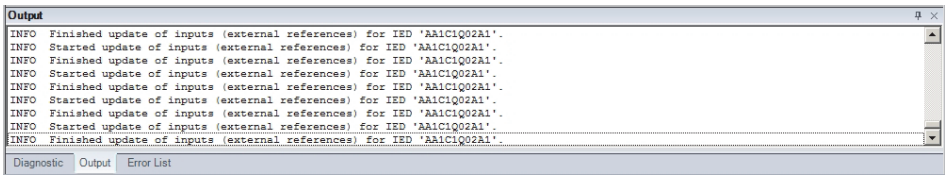
The Properties pane shows detailed properties for the selected element in the Navigation pane. The content of the properties pane will change depending on the type of node selected in the Navigation structure.



### 4.2.5 Logging and Messaging

The Messaging pane displays a filtered log of events to support analysis and trouble shooting of the project during engineering. The pane contains tabs to display logged events:

- Diagnostic
- Output
- Error list

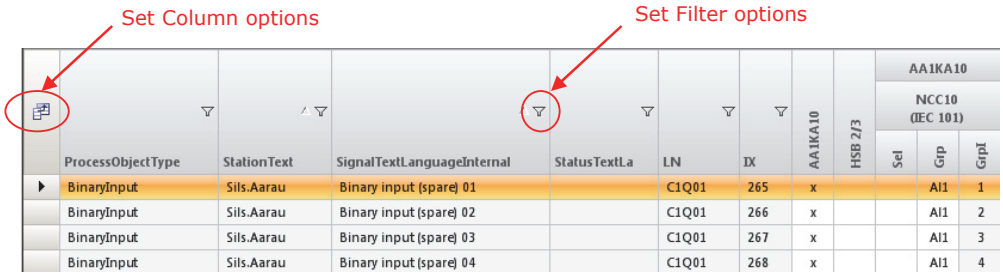



## 4.3 Using the Grid Editor

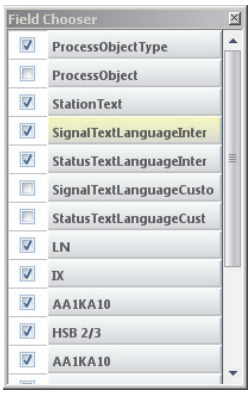
Several Editors in IET600 are implemented within a grid format (“Grid Editors”). These engineering tasks often requires configuration of many rows of data, which can profit from efficient filtering options and configuration shortcuts. This chapter will introduce some of the basic concepts introduced in IET Grid Editors.

### 4.3.1 Using Filters

To set the column options, click on the top left icon  :

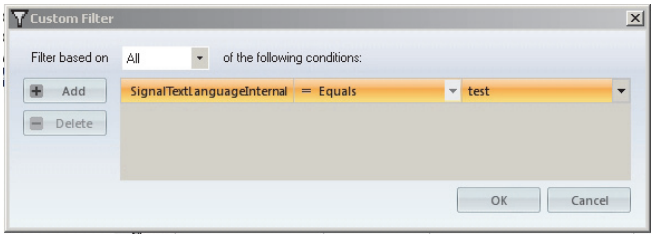



Select the fields to be shown in the editor by enabling/disabling the selection box in the Field Chooser window:



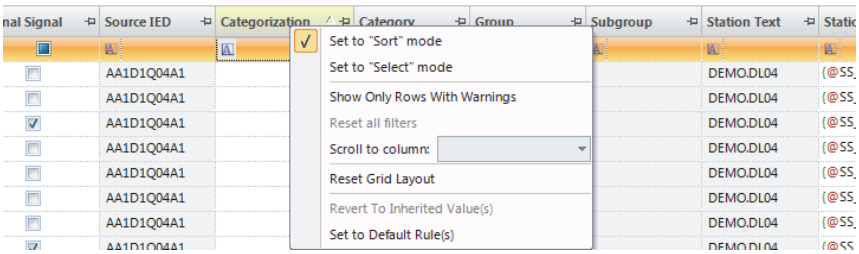

To change the Filter Options, click on the Filter Options icon and select:

- (All) - show all rows (reset filter)
- (Custom) - show all rows which match the conditions specified in the custom filter dialog




- (Blanks) - show all blank rows
- (NonBlanks) - show all rows with content (not blank)
- Text - show all rows with matching text

Several filtering options are available in the Grid Editors. Right-click on the filter row to access the context menu:




The following options are available:

- Set to “Sort” mode:
- Set to “Select” mode:
- Show Only Rows with Warnings
- Reset all filters
- Scroll to column: A search field to quickly navigate to a particular column
- Reset Grid Layout: Resets columns and filters to factory default

## 4.3.2 Editing functions

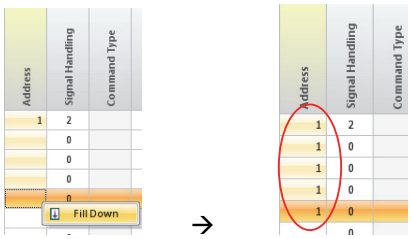
The Grid Editors provide additional editing functions to improve the efficiency of configuring multiple data fields. There are two main functions:

- Fill Down
- Copy & Paste
- Find and Replace
- Export to Excel

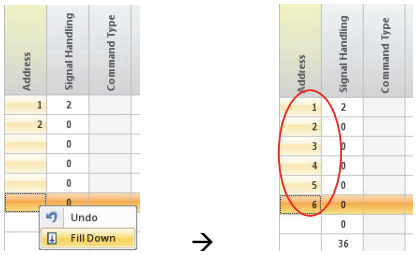
The “Fill Down” allows the user to automatically fill multiple cells in sequence. The “Fill Down” function operates differently for text fields than for numerical fields. For text fields, the “Fill Down” function works similar to a copy and paste, where the remaining fields will inherit the same text as the first field. For numerical fields the function can either copy the number in the first field into the remaining field or enter a sequence of numbers based on the first two selected fields.

To use the “Fill Down” function for text or number:

1. Select the first cell row in the selection (e.g. cell contains “1”)
2. Press SHIFT and hold, then select the last row in the selection
3. Right click and select “Fill down” from the context menu
4. The remaining fields will be filled with the content of the first row in the selection (text or number)




Note that if you select a column where the first two numbers in the selection are different, then the “Fill Down” function will enter a sequence of numbers in the remaining cells, based on the first two numbers. For example:

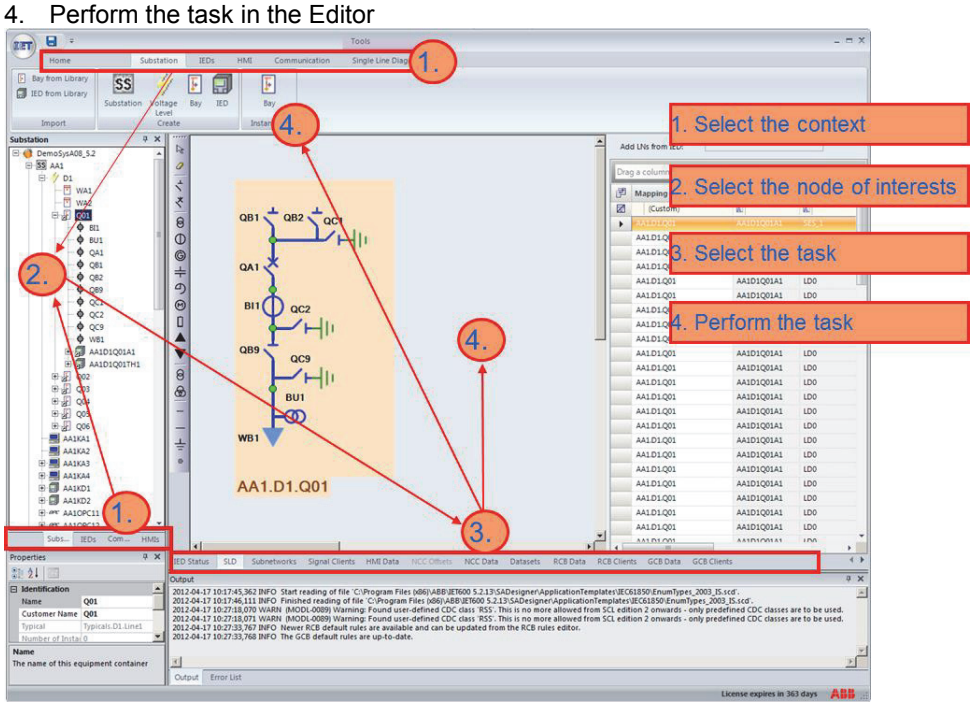


## 4.4 Navigation Concepts

This chapter describes the basic navigation concepts used in the IET600 user interface.

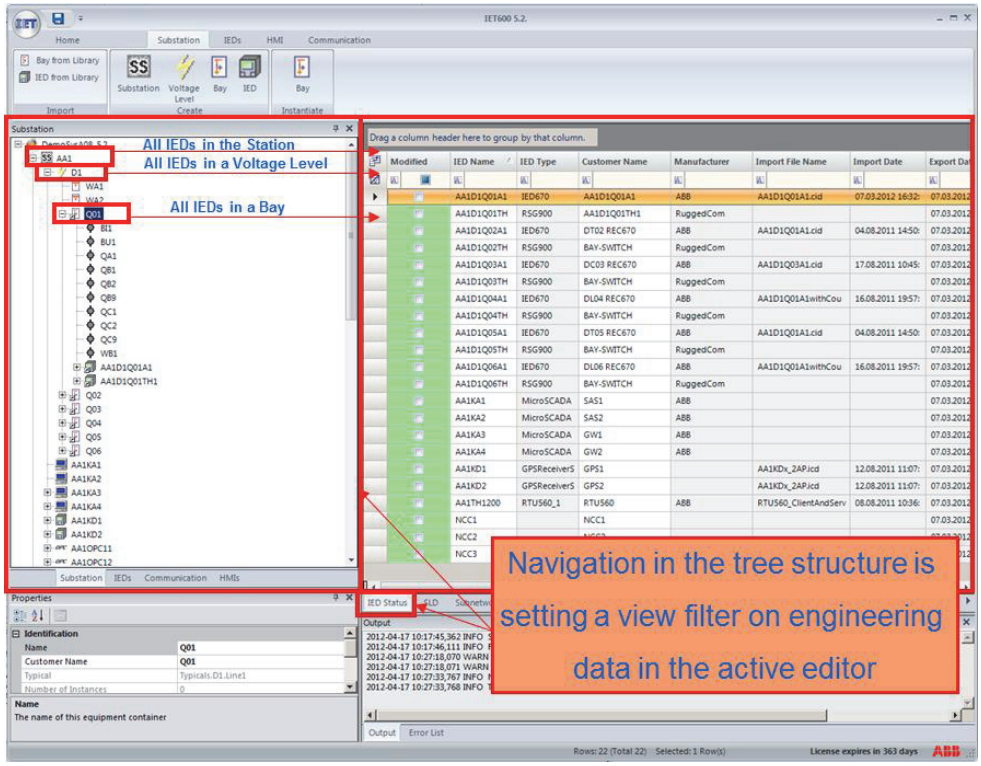
Generally, the typical navigation steps can be described as follows:

1. Select the context in the Menu pane
2. Select the node of interest in the Navigation pane
3. Select the task in the Editor pane
4. Perform the task in the Editor




Note that some editors are not available at the selected node in the navigation tab. In these cases, the editor tab is shown in a light grey colour and the editor shows the following message:


Moreover, navigating through the nodes in the Navigation pane is setting a view filter on the engineering data in the active editor. This allows the engineer to quickly “drill down” to the data of interest in the selected engineering editor.


All IEDs in the Station
All IEDs in a Voltage Level
All IEDs in a Bay
Navigation in the tree structure is setting a view filter on engineering data in the active editor

# 5 The IET600 Project

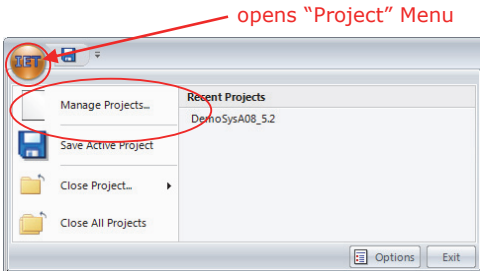
## 5.1 Overview

The first step in working with IET600 is to create a project. The IET600 project is a container for all project work.

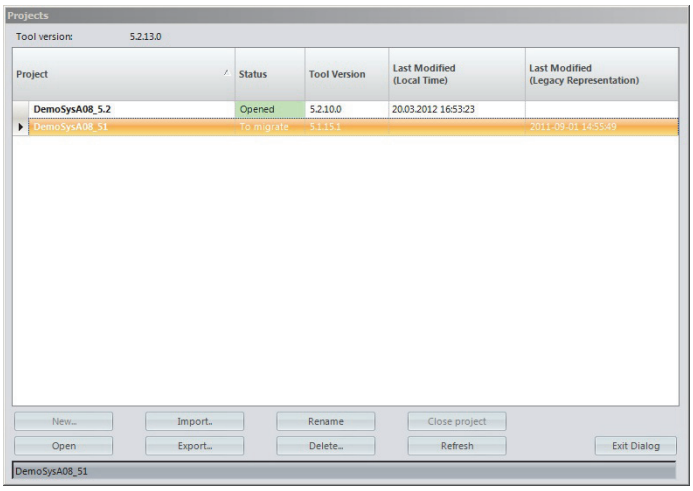
## 5.2 Working with IET Projects

### 5.2.1 Create a New Project

Projects are managed in the “Projects” dialog. Open this dialog by clicking on the round button on the top left of the user interface window and select “Manage Projects”.




The “Projects” dialog provides access to the projects stored in your computer and enables the addition and removal of projects.




To create a new Project, click on the “New” button and enter the name and directory for the new project.

## 5.2.2 Opening an Existing Project

To open an existing project, click on “Manage Projects” to open the “Projects” dialog. Select the project from the list and select “Open”.

The “Status” column shows the state of the project:

- Opened – project is already opened
- To migrate –project needs to be migrated in order to open in the current tool version
- Closed – indicates that the project can be opened without migration

It is also possible to open a second project in the same session. To open a second project, simply open the second project after the first is opened.

Note that the second project is “read-only” which means that data in this project cannot be modified during this session. However, you may copy data from the second project (e.g. entire bays, IEDs, signals, etc) into the first “active” project.

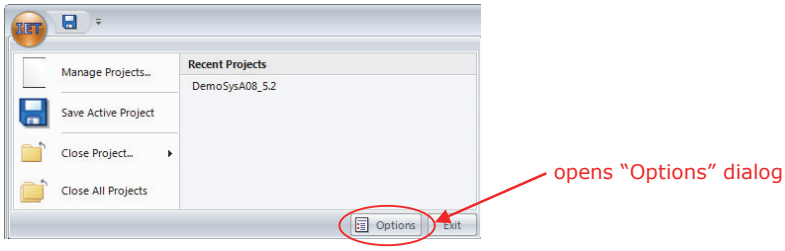
## 5.2.3 Import and Export Projects

A project can be easily exported and imported in order to either transfer the project to other users or to create a backup. The import/export functions can be accessed from the Projects dialog using the “Import” and “Export buttons. The exported project will be stored in a ".ietprj" file, which can be imported by other users.

## 5.2.4 Customize Project Options

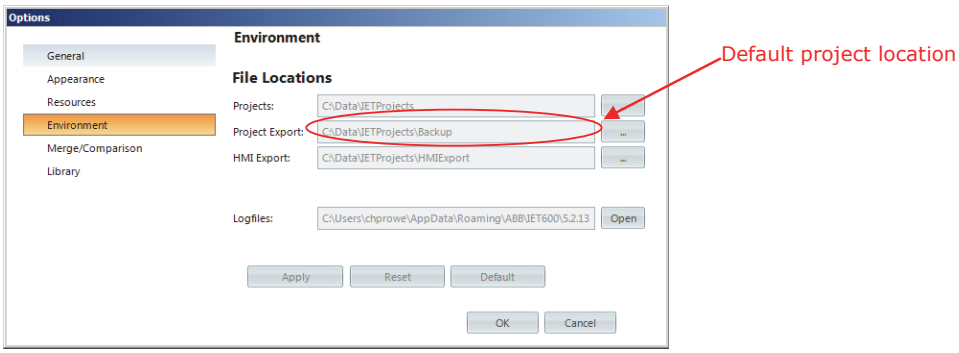
Open the projects menu and click the button “Options” to open the Options dialog. The dialog provides the possibility to customize:

- Appearance – application language and colour scheme
- Resources – information on IET600 installation
- Environment – File locations for projects, project export

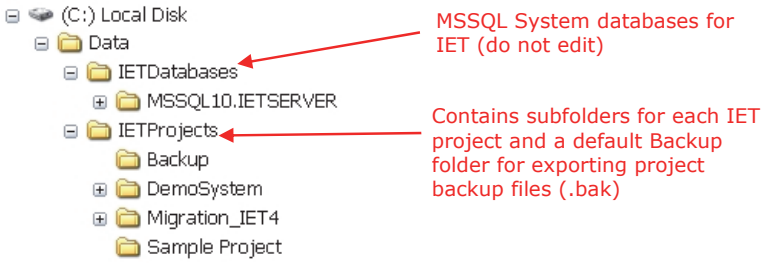



## 5.2.5 Organize Project Data

The project is saved under the directory specified in the "Options" dialog.

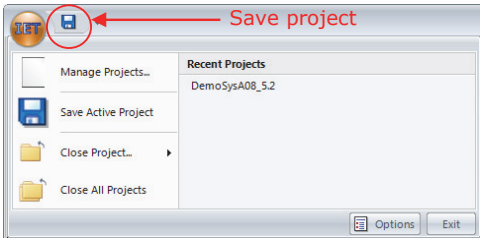



The project files are then stored according to the specified path. The default location of the IET600 files is located under C:\Data\IETProjects.




## 5.2.6 Saving Project Data

Note that IET600 does not save automatically. You must press the save button to save current state of the project. Tip: save your work regularly to shorten saving times.

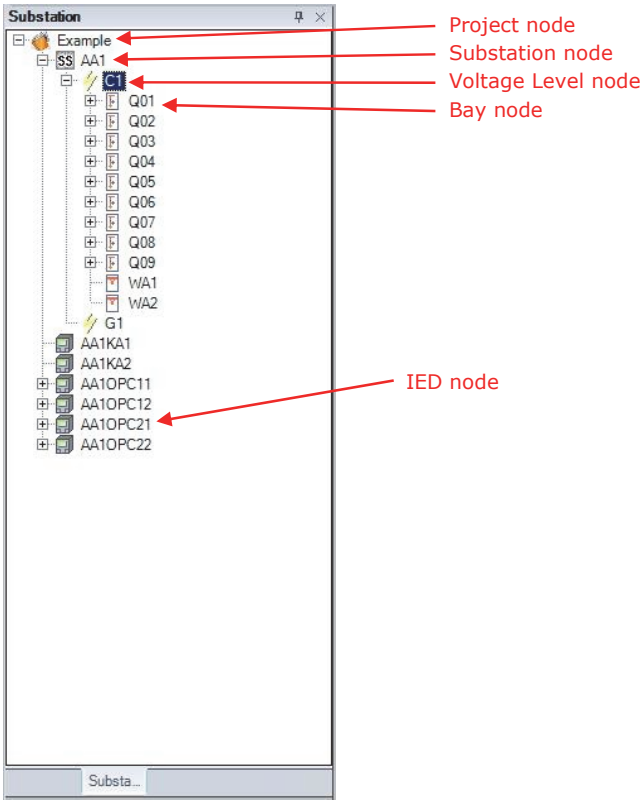


# 6 Substation Specification

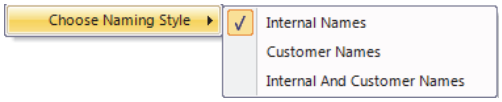
## 6.1 Introduction

### 6.1.1 Overview

The substation topology is defined in the Navigation menu. The Navigation structure of the substation consists of the substation, voltage level and bay nodes. The following section will describe how to build this structure in IET600.




The names shown in the Substation navigation tree can be displayed according to internal names, customer names or both. To change the naming style, right click on the background area of the Substation navigation pane and right click to display the following menu:



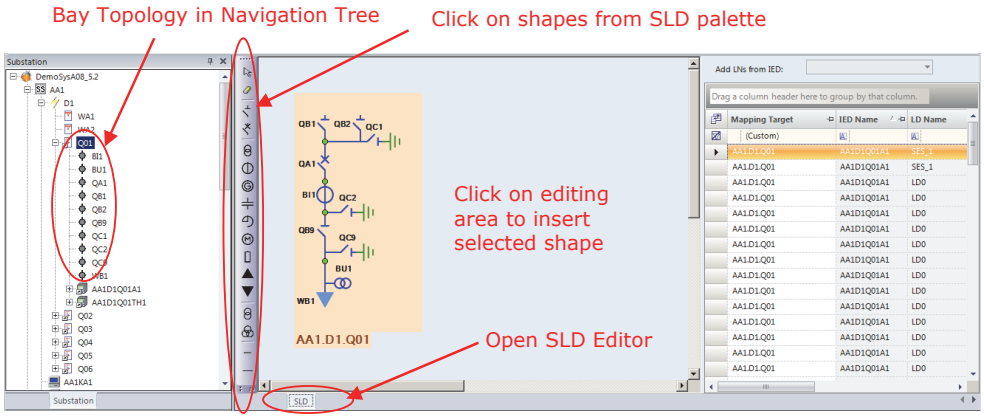

And select the naming style to display.

IET600 supports several ways to build the substation structure. The following sections describe these different methods.

Note that the substation topology can also be imported from an SCD file.

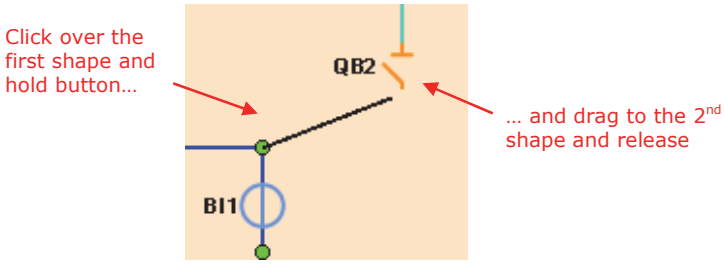
## 6.1.2 SLD Editor

The SLD Editor is a graphical editor for the configuration of the substation section. The editor provides a palette of shapes representing the primary equipment in the bay as well as the interconnection between these equipment. Moreover, the bay equipment can be connected to the Busbar to define a complete single line diagram for a voltage level.




To draw the single line diagram for a bay:

1. Select Bay node in the Substations navigation pane
2. Click on the primary equipment (e.g. a circuit breaker)
3. Move cursor to the Bay drawing area (orange box) and click to drop the equipment into the drawing area
4. Repeat steps 1 to 3 to draw all primary equipments
5. Connect the primary equipment by selecting the Connect Two Elements shape  , then click and drag the line from the first element to the second:

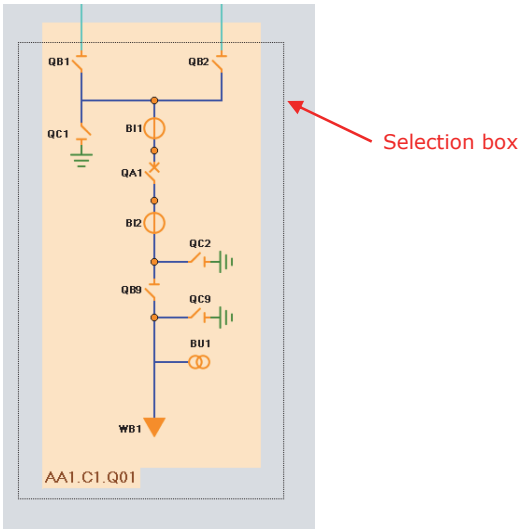



Note that to view the shape names, simply move your cursor above the shape and the Tooltip box will appear.

Alternatively, you can right click on the connectivity node and select “Attach Conducting Equipment” to connect to available shapes.

Multiple shape selection and Copy & Paste are also supported. To select multiple shapes, and copy and paste:

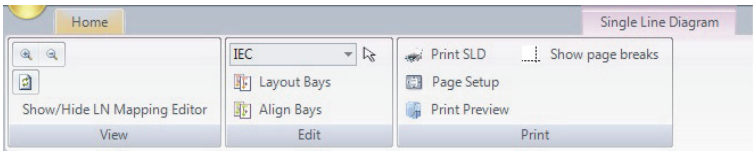
1. Select Arrow shape 
2. Press down “Ctrl” key on the keyboard
3. Click and drag the selection box over the shapes (grey dashed box) and release to select the shapes (shapes change colour to orange)




4. Press “Ctrl-C” and “Ctrl-V” shortcuts on the keyboard to cut and paste, respectively

In the Menu pane, the Single Line Diagram menu provides several functions to Edit the single line diagram:

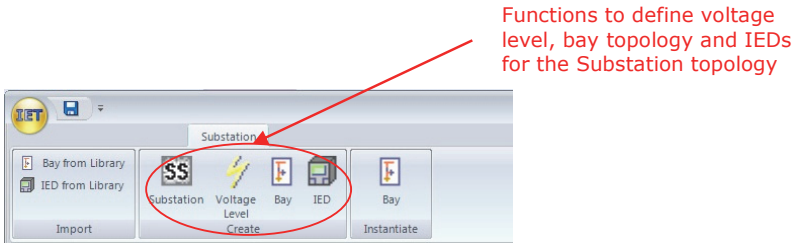
- IEC/ANSI/Custom: Shapes for SLD editor according to selected standard
- Align Bays: aligns multiple bays in the same drawing (e.g. voltage level) to avoid overlapping bays
- Layout Bays: aligns multiple bays in the same drawing (e.g. voltage level) and orders bays in alpha-numerical order from left to right




## 6.2 Build Substation Topology Manually

Open the Substation tab in the Menu to access the IET600 functions to build the substation topology. In the Build group, the following functions are available:

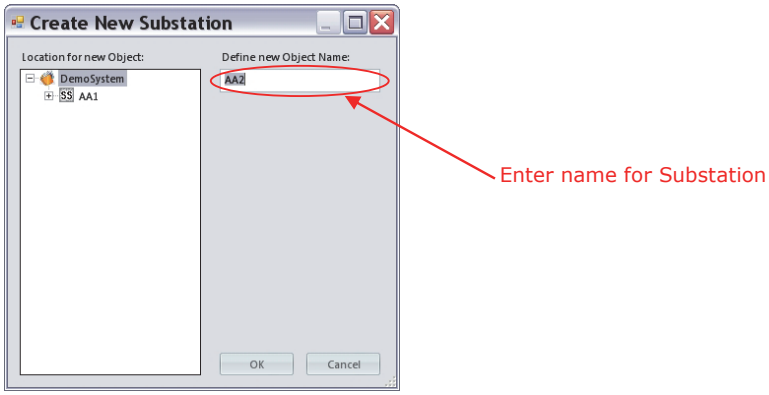
- Build Substation
- Build Voltage Level
- Build Bay
- Build IED




Alternatively, these functions can be accessed through the Context Menu of the respective nodes in the Navigation Tree.

First create the substation node:

1. Select the project node and right click to open the Substation dialog. Alternatively, select the Substation function in the menu to open the same dialog
2. Enter a new Object Name for the substation
3. Click OK and a new Substation node is created in the Navigation view

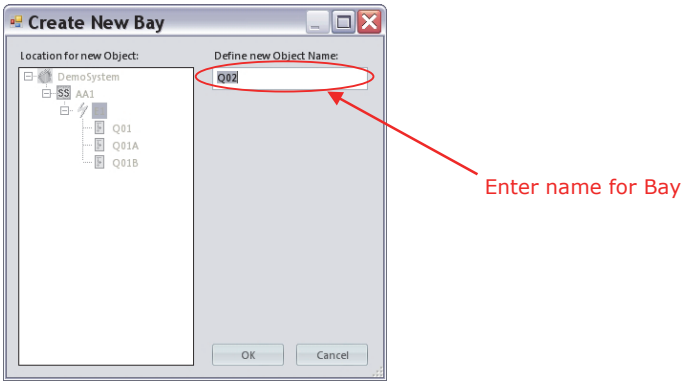



Next, create a new voltage level:

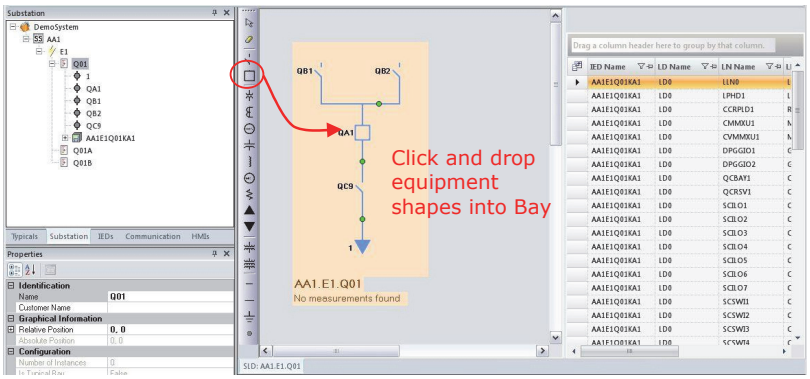
1. Select the substation node and right click to open the menu. Alternatively, click on the Voltage Level function in the menu to open the same dialog
2. Enter a new Object Name for voltage level. The name must be unique within the substation
3. Click OK and a new Voltage Level node is created in the Navigation view


Finally, create a new Bay:

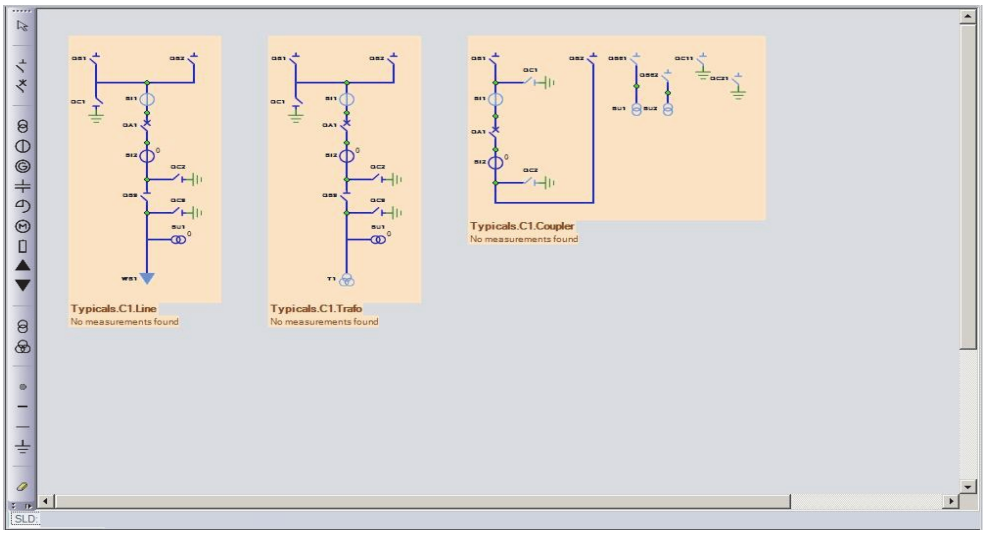
1. Select the Voltage Level node, right click and select “Create New Bay” or click on the Bay function in the menu to open the Bay dialog
2. Enter an Object name for the new Bay in the dialog
3. Click OK and a new bay node is created in the Navigation view




Lastly, the bay topology itself, including the primary equipment and configuration needs to be defined.




To define the full topology of the substation, repeat the steps described above for each successive bay. The full substation topology can be viewed from the voltage level node.




## 6.3 Build Substation Topology by Importing SCD File

To create the substation structure by importing an SCD file, please refer to chapter 13.1: SCL files.

# 7 IED Data Engineering

## 7.1 Introduction

### 7.1.1 Overview

In IET600, you have several ways to engineer IEDs:

- create an IED manually, and then add/edit their data
- copy a configured IED from the same or another project

### 7.1.2 IED Status Editor

The IED Status editor provides an overview of all IEDs in the project, including revision history information. The following properties are available:

| Property | Description |
|---|---|
| Modified | Indicates if changes to the IED have been made since the last SCD export. The flag is reset with SCD export. |
| IED Name | IED name |
| IED Type | IED product family (from imported CID/ICD/IID file) |
| Customer Name | Customer specific name |
| Manufacturer | Manufacturer name (from imported CID/ICD/IID file) |
| Import File Name | Path and file name of imported CID/ICD/IID file |
| Import Date | Date and time of CID/ICD/IID file import |
| Export Date | Date and time of SCD file export |
| Configuration Revision | IED Configuration Revision (from imported CID/ICD/IID file) |
| Software Revision | IED Software revision (from imported CID/ICD/IID file) |
| User Name | Identifies PC user name at file import |
| Primary Role | Imported from SCL file (if available). Can be modified in the IED properties window. |
| HMI Aspect Count | Number of HMI aspects defined in HMI Data editor per IED |

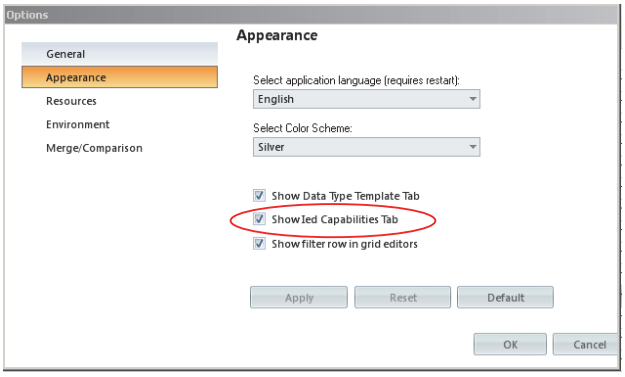
| | |
|---|---|
| PCM Template | Reference to PCM600 IED Configuration Template |
| PCM Object Type | Reference to PCM600 Object/IED Type |
| Capability Description | Reference to Manufacturer and IED Type in the IED Capabilities editor (read-only) |

### 7.1.3 IED Capabilities Editor

The IED Capabilities editor defines the tool behaviour for SCL import and dataflow generation for specific device types. The capabilities of the IED may be defined in the IED Services part of the SCL file, but this is not always available in the ICD/CID/IID/SCD file. Therefore, the IED Capabilities editor allows the user to specify these attributes for the purpose of correct engineering in the IET tool. Note that the changes in the IED Capabilities tab will not be written into the SCL IED Services of the exported SCD file.

It is not recommended to change the default settings in the IED Capabilities editor. The default options are set according to typical engineering process for defined ABB devices and reflect product-specific know-how and engineering experience. Expert knowledge and extreme care must be taken, as changing these settings may result in improper functioning of the system.

To enable the IED Capabilities editor, open the main menu (top left circle) and select Options and enable the checkbox “Show Data Type Template Tab”:




The IED Capabilities editor is shown in the IEDs tab and shows the following properties:

1. General IED properties

| **Property** | **Description** |
|---|---|
| Manufacturer | IED manufacturer |
| IED Type | IED Type / Family |
| Configuration Version | (not used) |

2. Capabilities related to first import of an ICD file

   These properties serve to simplify engineering processes in the system tool. Datasets/RCB/GCB contained in the ICD file can be deleted per default upon SCL import and regenerated in the system tool.

| **Property** | **Description** |
|---|---|
| Ignore Dataflow on First Import | TRUE if all Datasets, RCBs and GCBs should be deleted when importing the ICD/CID file into an empty IED |

3. Capabilities related to RCB Client configuration

| **Property** | **Description** |
|---|---|
| CanModify Report Enabled Max | TRUE if Report Enabled Max attribute can be modified |

4. Capabilities related to serve as initialization values for IET600-specific attributes

| **Property** | **Description** |
|---|---|
| Default Report Control Block Status | Default status of RCB when imported from SCL file. IedDefinedFix prevents any changes to the imported RCB. IedDefinedConfigurable allows changes to be made to the imported RCB. |
| Default Goose Control Block Status | Default status of GCB when imported from SCL file. IedDefinedFix prevents any changes to the imported GCB. IedDefinedConfigurable allows changes to be made to the imported GCB. |
| Default SampledValues Control Block Status | Default status of SVCB when imported from SCL file. IedDefinedFix prevents any changes to the imported SVCB. IedDefinedConfigurable allows changes to be made to the imported SVCB. |
| Default Dataset Status | Default status of dataset when imported from SCL file. IedDefinedFix prevents any changes to the imported dataset. IedDefinedConfigurable allows changes to be made to the imported dataset. |

5. Capabilities related to limitations not defined via SCL file

   While IEDs may specify a maximum number of instances per RCB in the SCL file, this does not correspond to the number of concurrent connections which are possible. This property provides an additional parameter to take such limits into account.

| Property | Description |
|---|---|
| Max Concurrent Report Clients | Maximum number of allowed control blocks instances or clients. |

6. Capabilities that serve to override values of an SCL file in project

   These properties serve to override values contained in an SCL file. If these values are not defined, values from the SCL file will be observed.

| Property | Description |
|---|---|
| Override for Service ConfDataset Max | Change the maximum number of datasets from the value specified in the SCL file. If not defined, values from the SCL file will be observed. |
| Override for Service ConfDataset Max Attributes | Change the maximum number of dataset attributes from the value specified in the SCL file. If not defined, values from the SCL file will be observed. |
| Override for Service ConfDataset Modify | TRUE means that a preconfigured dataset may be modified. |
| Override for Service ConfReportControl BufMode | Specifies the buffer modes (buffered, unbuffered, both) allowed to configure for new control block types. If not defined, values from the SCL file will be observed. |
| Override for Service ConfReportControl BufConf | TRUE means that the buffered attribute of preconfigure report control blocks can be changed via SCL. |
| Override for Service ConfReportControl Max | Maximum number of instantiated report control blocks. If this number is equal to the number of preconfigured instances, then no new instances can be created. |
| Override for Service Goose Max | Maximum number of GOOSE control blocks, which can be configurable for publishing (max=0 means that the device is only a GOOSE client) |
| Override for Client Service Goose | TRUE means that client services for GCB is enabled |
| Override for Client Service SampledValues | TRUE means that client services for SVCB is enabled |
| Override for Client Service Buffered RCB | TRUE means that client services for buffered RCB is enabled |
| Override for Client Service Unbuffered RCB | TRUE means that client services for unbuffered RCB is enabled |

### 7.1.4 LN Data Editor

The LN Data editor provides a view on the IEC 61850 model of the IED, including the following attributes.

- IED
- LD
- LN
- Path
- Value
- Short Address
- Description

Note that most of these attributes are “read-only”, as they are defined in the IED’s ICD/CID/IID file. Some attributes are editable; however these shall only be used for engineering of COM581 gateways.

## 7.2 Create IEDs

### 7.2.1 Overview

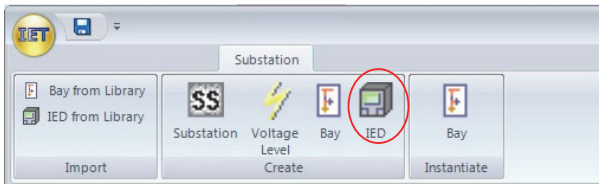
You can create IEDs in several locations:

- in the IED Tree below a project.
- in the Substation Tree, below a Substation, Voltage Level or Bay.

### 7.2.2 Create an IED

An IED can be created using either the Substation menu or in the Substation or IEDs navigation tree using the context menu.

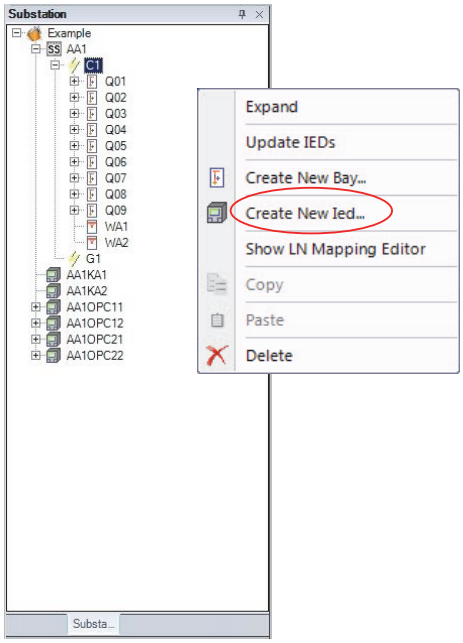
1. In the Substation menu, select the “IED” button:



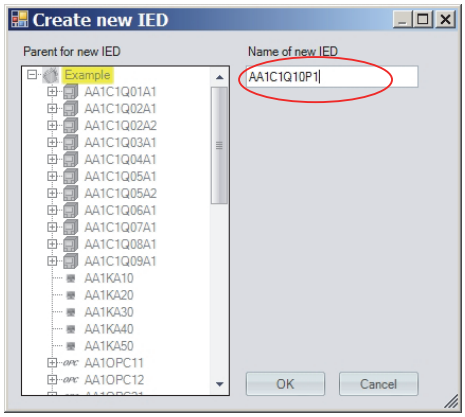

Alternatively, select the Substations or IED navigation, and click on a node under which you want to create an IED. This could be either:

- a ‘Substation’, ‘Voltage Level’ or ‘Bay’ node in the ‘Substation’ tab, or
- the ‘Project’ node in the IED tree

Right-click the node -> the Context Menu for that node appears. Select “Create New IED”.




2. This opens the dialog “Create new IED”.




3. IET600 proposes a name, which you can adapt according to your needs. Invalid names are indicated by an error mark (invalid names typically are either duplicate names or contain characters that are not allowed for a particular node).
4. Click “OK” to create the new IED or “Cancel” to close the dialog without creating an IED.
5. If you created a new IED, the Navigation Tree will change to either “Substation” or “IEDs”.

Note that an IED node is simply a container for the IED. To build the IED 61850 model, import the ICD file for the IED, as described in the next section.

## 7.3 Update IEDs

### 7.3.1 Overview

Updating an IED involves importing the ICD or IID file for a given IED. IEDs can be updated:

- from the Main menu -> IEDs tab -> Update IEDs
- from the context menu of an IED (single IED only)
- from the context menu of a node that can have IEDs below it, e.g. Substation, Voltage Level and Bay

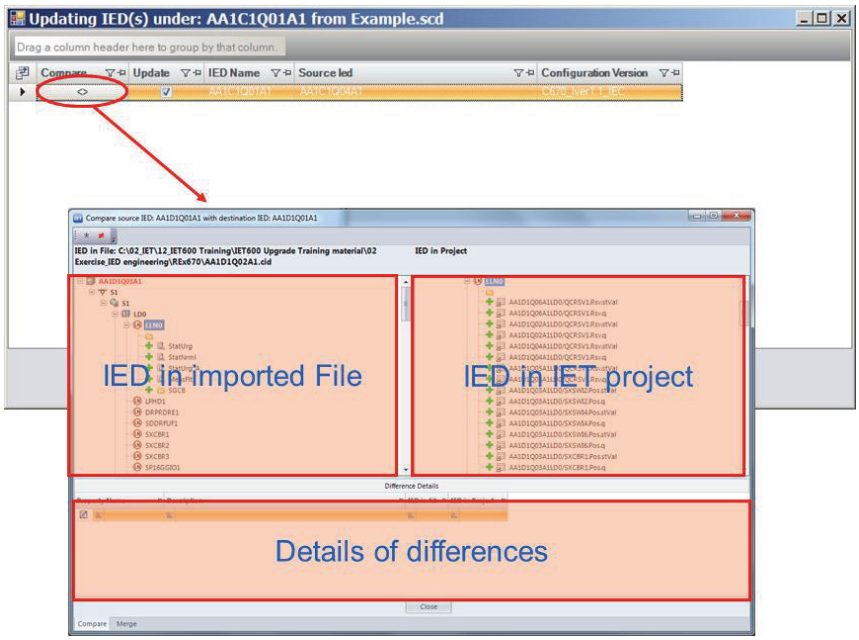
### 7.3.2 Compare and Merge function

Compare and merge dialogs provide support for reimport of SCL data. The Compare and Merge dialog can help reduce errors in projects due to incorrect changes in the project.

The Compare and Merge dialog is available when updating or re-importing SCL files into the project (e.g. Update IED). The dialog provides:

- preview of the content in import file
- compare differences between data in project and in the import file
- merge data or cancel import

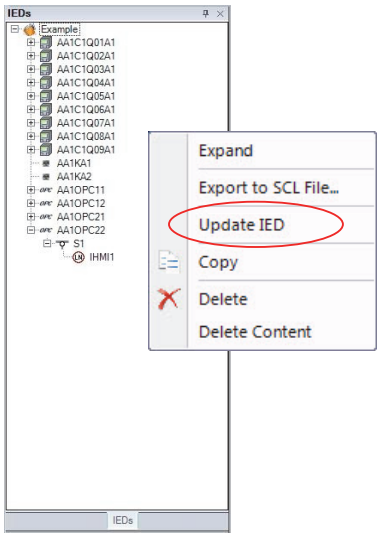
The dialog is opened by clicking on the Compare column during Update IED, as shown below:



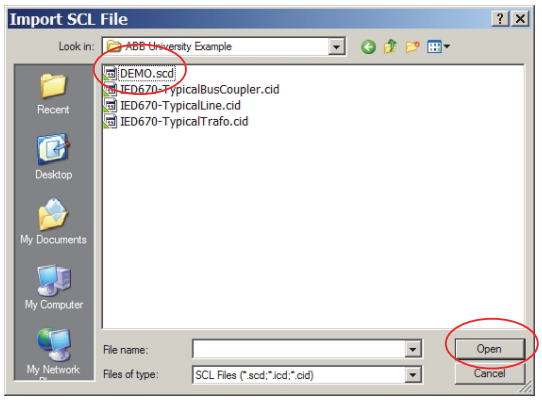

On the left-hand side of the dialog is the IED model in the import file and on the right-hand side is the existing IED model in the project. By selecting a node in either window, the details of the differences are shown in the bottom window.

## 7.3.3 Update One IED

1. Right-click on an IED -> the IED context menu appears:

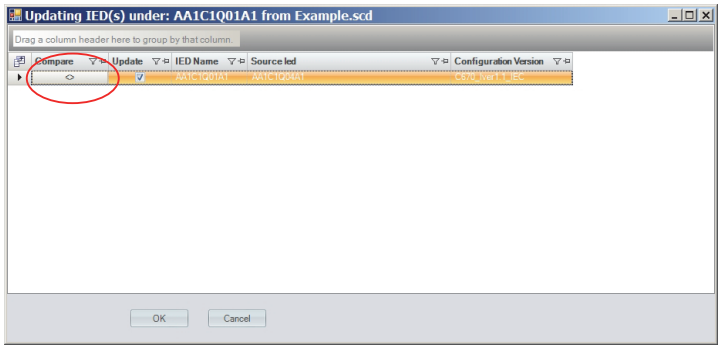


2. Select “Update IED” -> the normal dialog for selecting a file appears:




Select any valid SCL file (SCD-, ICD-, IID-file) and click “Open”.

3. The following dialog appears:

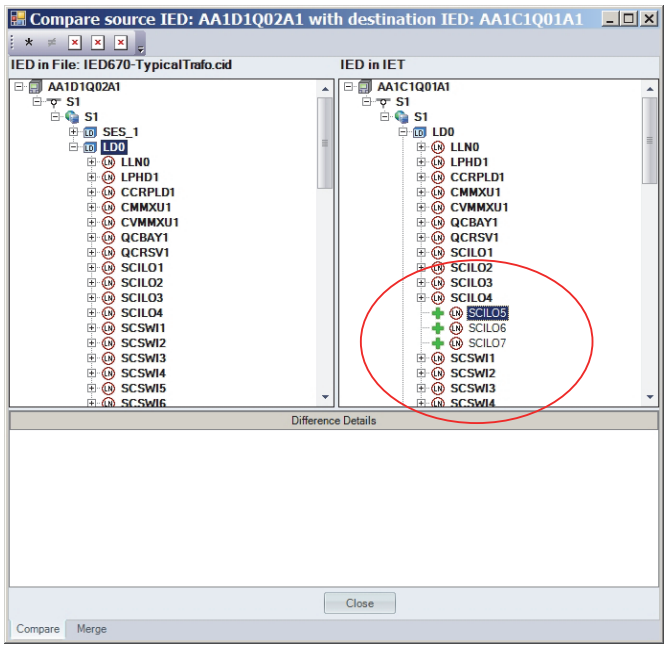



IET600 will automatically match IEDs with the same name in IET600 and in the file. Alternatively you can select the Source IED (in the file).

The “Compare” column indicates whether the IET600 IED and the IED in the file are equal (==) or different (<>). You can click on the symbol, which will open the Compare and Merge dialog as described in the previous section.

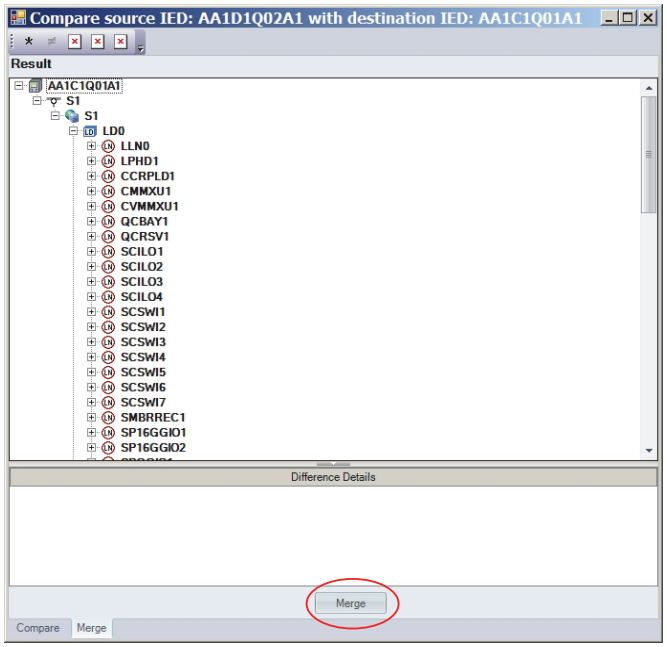
4. To import the IED from the file, click “OK”. The IED will be imported without further questions.

5. To check the differences between the Source and the Destination IED, click the button in the “Compare” column. The following dialog will open:




6. Check the differences. If for some reason (e.g. unexpected differences), you do not want to import the file, click “Close”. The dialog will close and nothing will be imported.

7. If you want to proceed, select the tab “Merge”. Now the dialog looks as follows:

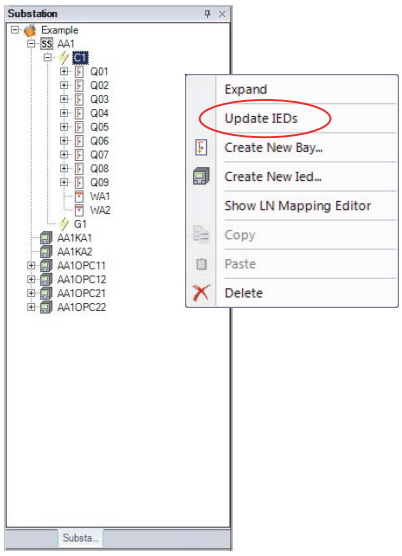



In this dialog you can see the result of the import, i.e. how the IED will look like after the import.

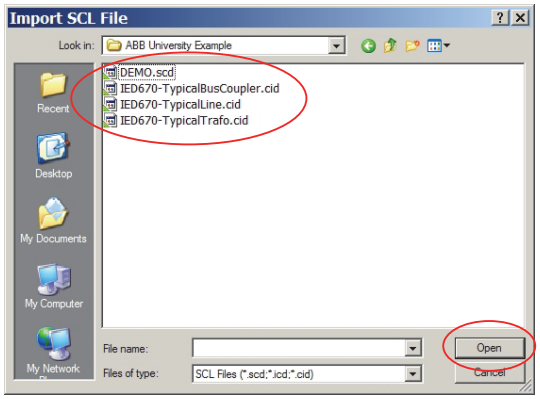
8. To import the IED from the file, click “Merge”. The IED will be imported and the dialog will close. To exit without import, change back to the tab “Compare” and click “Close” there.

### 7.3.4 Update Several IEDs Together

1. Right-click on a Project, Substation, Voltage Level or Bay-> the context menu appears:

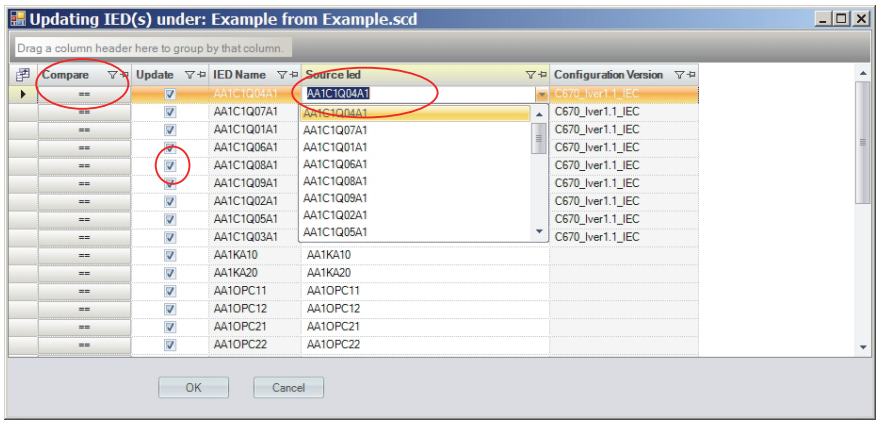



2. Choose “Update IEDs” -> the normal dialog for selecting a file appears.




Select any valid SCL file (SCD-, ICD-, IID-file) and click “Open”.

3. The following dialog appears:




IET600 will automatically match IEDs with the same name in IET600 and in the file. Alternatively you can select the Source IED (in the file).

The “Update” column allows you to select/deselect individual IEDs for updating. Only IEDs marked with ☑ will be included in the update.

The “Compare” column indicates whether the IET600 IED and the IED in the file are equal (==) or different (<>). You can click on the symbol, which will open the Merge and Compare dialog as described in the previous sections.

4. Clicking “OK” in the “Update IEDs” dialog will update the selected IEDs. Clicking “Cancel” will close the dialog without updating IEDs (except those for which you have not done an individual merge!).

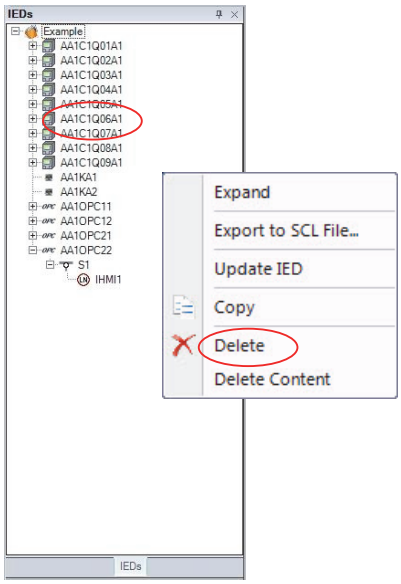
## 7.4 Delete IEDs

### 7.4.1 Overview

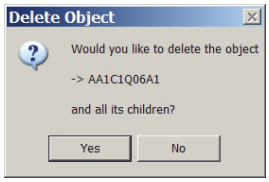
Basically, you can delete any IED from its context menu. You can either delete the IED totally or you can remove its content, leaving the IED empty.

## 7.4.2 Delete a Complete IED

1. Right-click on the IED you want to delete -> the context menu appears:



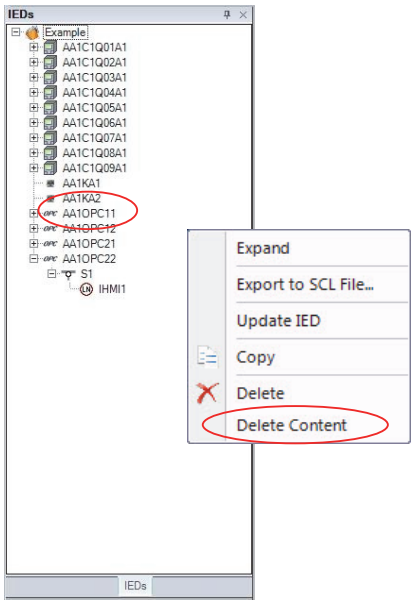

2. Select “Delete” -> a confirmation dialog appears:




3. Confirm deletion with “Yes” or cancel it with “No”.

### 7.4.3 Delete the Contents of an IED

1. Right-click on the IED whose contents you want to delete -> the context menu appears:




2. Select “Delete Content” -> a confirmation dialog appears:


3. Confirm deletion with “Yes” or cancel it with “No”.

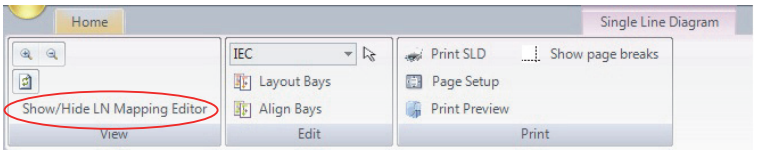
## 7.5 Map LNs to Substation Elements (“LN Mapping”)

### 7.5.1 Overview

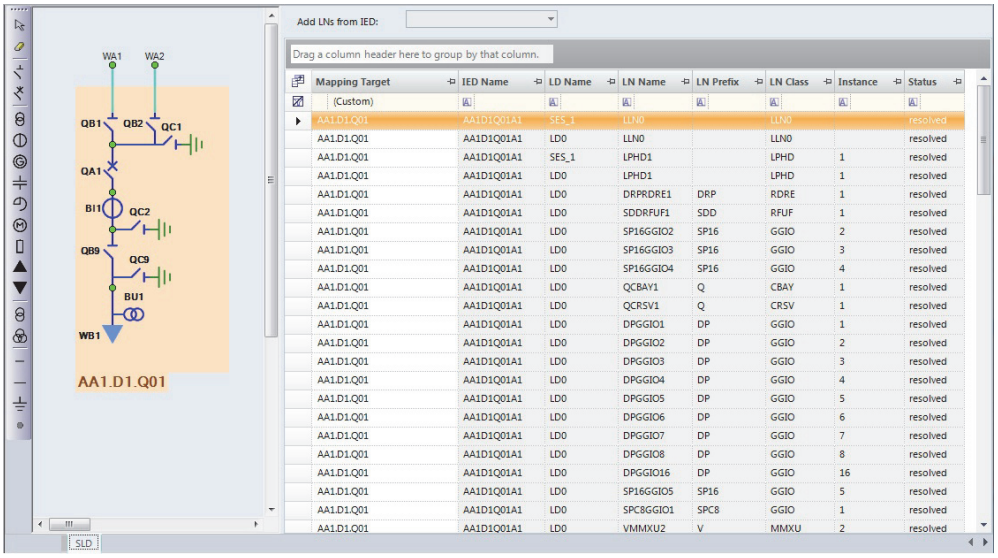
According to part 6 of the IEC 61850 standard, the SCD file shall describe not only the system specification in terms of the single line diagram and configured IEDs in terms of logical nodes (LNs), but also the relation between these LNs to parts and equipment of the single line. The LN Mapping table in IET600 serves as an editor configure this relation. The table is available in the Single Line Diagram (SLD) editor.

Since the LN mapping information is not available from the IED’s ICD/IID file, it must be entered manually in the LN mapping table. Note that this information must be synchronized with the functional/LN mapping used in the IED configuration tool (e.g. PCM600).

In the Single Line Diagram menu, enable the “Show/Hide LN Mapping Editor”:




The LN mapping table is shown on the right hand side of the SLD editor, as shown below:




| Mapping Target | IED Name | LD Name | LN Name | LN Prefix | LN Class | Instance | Status |
|---|---|---|---|---|---|---|---|
| (Custom) | | | | | | | |
| AA1.D1.Q01 | AA1D1Q01A1 | SES_1 | LLN0 | | LLN0 | | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | LLN0 | | LLN0 | | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | SES_1 | LPHD1 | | LPHD | 1 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | LPHD1 | | LPHD | 1 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DRPRDRE1 | DRP | RDRE | 1 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | SDDRFUF1 | SDD | RFUF | 1 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | SP16GGIO2 | SP16 | GGIO | 2 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | SP16GGIO3 | SP16 | GGIO | 3 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | SP16GGIO4 | SP16 | GGIO | 4 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | QCBAY1 | Q | CBAY | 1 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | QCRSV1 | Q | CRSV | 1 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO1 | DP | GGIO | 1 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO2 | DP | GGIO | 2 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO3 | DP | GGIO | 3 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO4 | DP | GGIO | 4 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO5 | DP | GGIO | 5 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO6 | DP | GGIO | 6 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO7 | DP | GGIO | 7 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO8 | DP | GGIO | 8 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | DPGGIO16 | DP | GGIO | 16 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | SP16GGIO5 | SP16 | GGIO | 5 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | SPC8GGIO1 | SPC8 | GGIO | 1 | resolved |
| AA1.D1.Q01 | AA1D1Q01A1 | LD0 | VMMXU2 | V | MMXU | 2 | resolved |

Each row in the table refers to a LN and its properties. The following properties are shown:

- Mapping Target – reference to the bay and primary equipment
- IED Name, LD Name, LN Name, LN Prefix, LN Class, Instance – LN properties
- Status – indicates the source of the LN:
  - "resolved" indicates a LN from an IED in the project,
  - "virtual" indicates that a LN has been defined from an HMI signal and is not yet mapped to an IED

The last point refers to a concept of decoupling between the IED's IEC 61850 model and the HMI signals, which is described in the chapter on HMI Engineering.

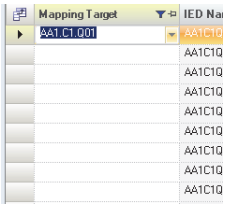
Note that the LN Mapping table is only available in a single bay view. By default, only IEDs which have been allocated to this bay will be shown in the LN Mapping table. To map LNs from an IED which belongs to another bay, select the field "Add LNs from IED" and select another IED in the project.

### 7.5.1.1. Using the "Fill Down" function

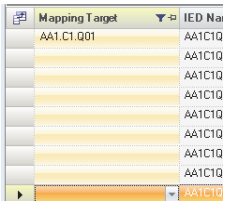
The "Fill Down" function is available in the Grid Editor for the LN Mapping Editor. The "Fill Down" function allows you to copy a Mapping Target in one cell and apply it to a

range or selection of cells. To use this function, you need to perform the following steps:

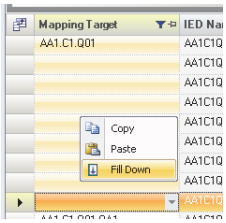
1. Select the first row in the selection



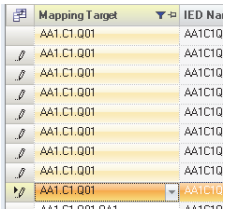

2. Press SHIFT and select the last row in the selection




3. Right click and select “Fill down” from the context menu




4. The empty fields will be filled with the content of the first row in the selection




### 7.5.1.2. LN Mapping Restrictions

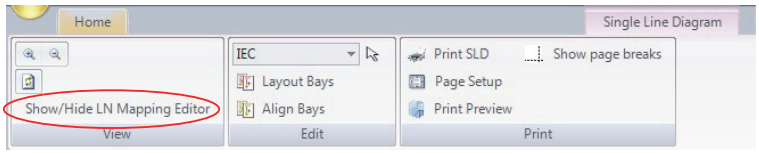
IET600 places some restrictions to the LN Mapping Editor. These restrictions were implemented to improve the quality of the substation section in the SCD file. A summary of these restrictions are listed below:

- LN type CSWI and SCILO can only be mapped to equipment of the type Circuit Breaker (CBR) or Disconnector (DIS)
- LN type XCBR and RSYN can only be mapped to equipment of the type CBR
- LN type XSWI can only be mapped to equipment of the type DIS

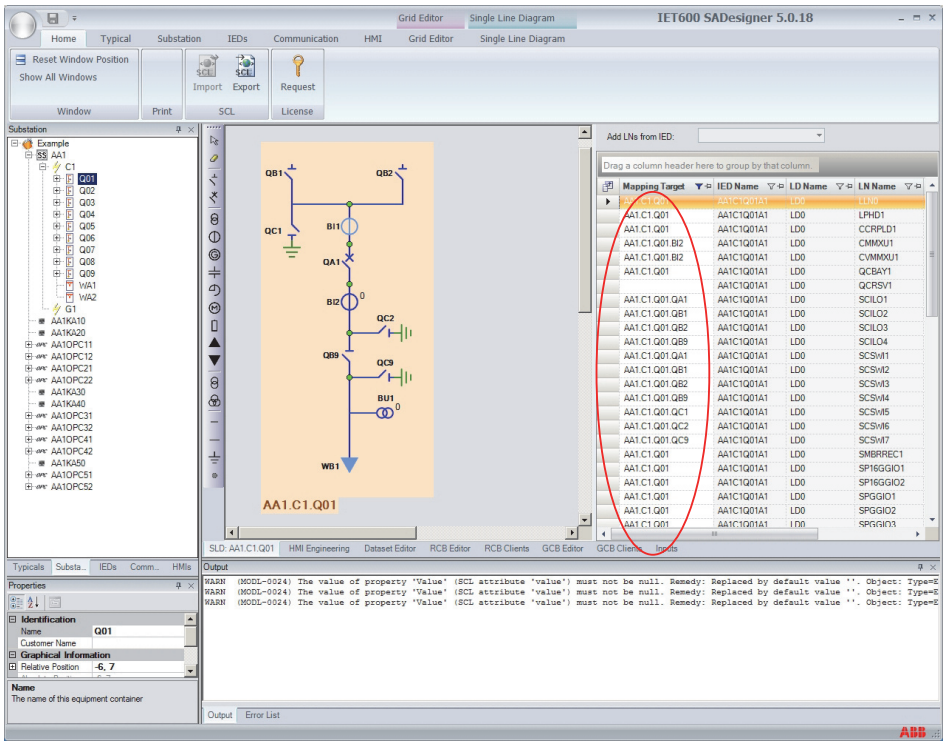
- LN type MMXU can only be mapped to equipment of the type Current Transformer (CT) or Voltage Transformer (VT)
- LN type LLN0 and LPHD can only be mapped to the bay
- One exception to the above rules is that all LN types can be mapped to the bay

### 7.5.2 Map LNs in Bays

1. Select the “Substation” tab in the Navigation pane
2. Select the Node you want in the configuration Tree (“Substation”, Voltage Level or Bay). The selection of this node determines which LNs are shown for mapping and which equipment is available for mapping (basically all LNs and all equipment below that node).
3. Select the SLD tab in the Editors.
4. If you have selected a Voltage Level or Bay, the LN Mapping Editor will show on the right of the Single Line Diagram. If it is hidden; it can be made visible from the tab Single Line Diagram -> Show/Hide LN Mapping Editor.




5. On the right of the single line diagram is the LN Mapping Editor. The first column shows the mapping target and the remaining columns to the right show the available LNs and their properties. The first column allows you to select a Substation item (Bay or Equipment) to map to that LN




Note: to select the LN Mapping target, one can either type in the target, select it from a drop down menu, or use the “Fill Down” function to fill multiple rows automatically. IET600 will apply the LN mapping restrictions as described in the previous section.

# 8 Communication Engineering

## 8.1 Introduction

### 8.1.1 Overview

The substation communication configuration is defined in the Communication navigation pane and menu. The Communication navigation structure defines the subnetwork and IEDs (respectively, IED access points) which are assigned to this subnetwork. The specific communication properties are defined in the Subnetworks editor, including:

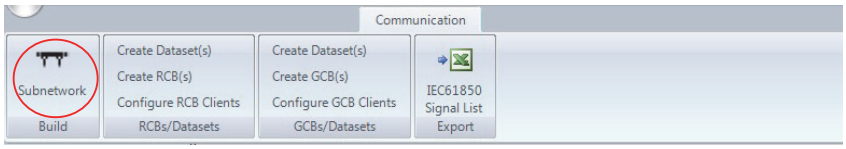
- Subnetwork name
- IP Address
- Subnet

The following section will describe how to build this communication structure and define the communication properties in IET600.

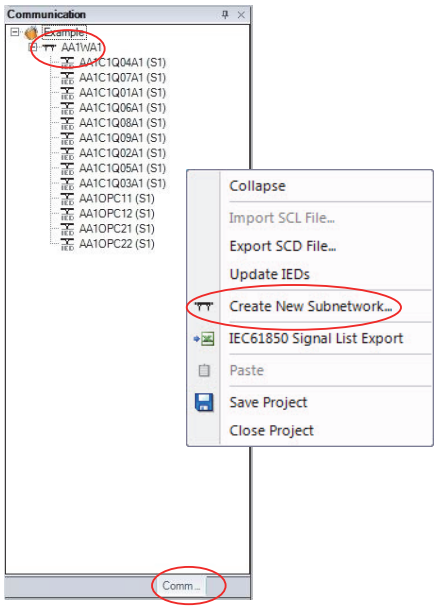
## 8.2 Subnetwork Configuration

### 8.2.1 Create a Subnetwork

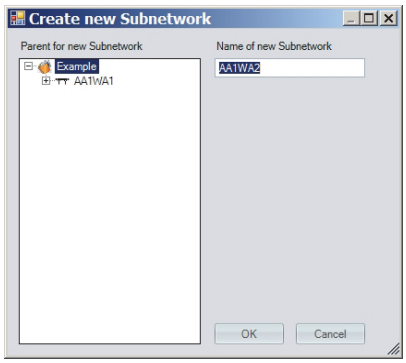
1. Open the “Communication” menu tab and select the “Subnetworks” button:




Alternatively, select the Communication navigation tab, right click on the project node and select “Create New Subnetwork”:




2. The Dialog “Create New Subnetwork” appears:

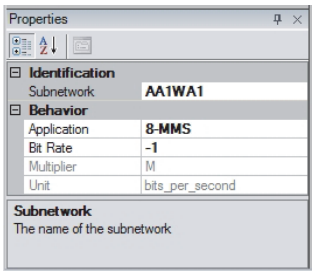



3. (A Subnetwork can only be created in the Root Node which should be pre-selected).

   The tool proposes a name, which you can adapt according to your wishes. Invalid names are indicated by an error mark (Invalid names typically are either duplicate names or contain characters that are not allowed for a particular node).

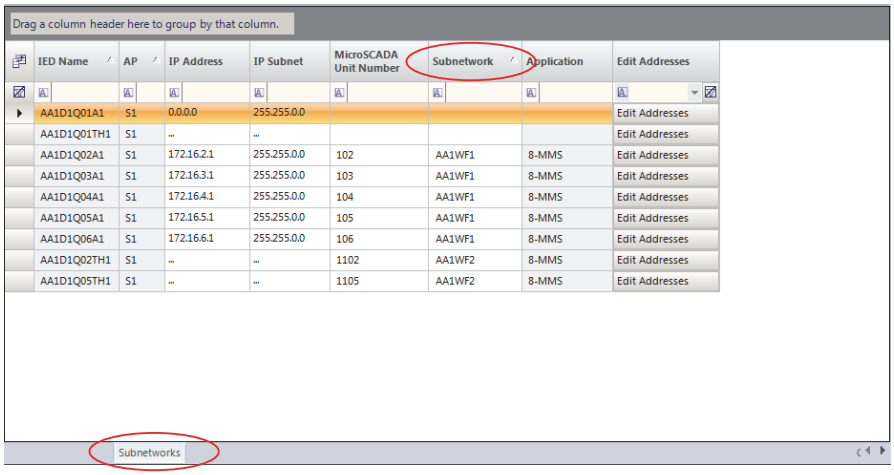
4. Click “OK” to create the new Subnetwork or “Cancel” to close the dialog without creating a Subnetwork.
5. If you created a new Subnetwork, the Navigation Panel will change to “Communication” and the Subnetwork will appear there. The editor will not change.

A Protocol description (e.g. 8-MMS, SPA, IEC104 etc.) can currently only be added in the “Properties” window:

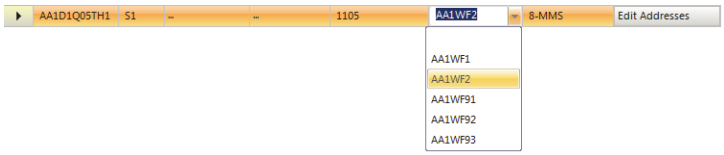



## 8.2.2 Connect IEDs to a Subnetwork in the Subnetwork Editor

1. Select a tab in the navigation tree where the Subnetwork Editor is visible (“IEDs” or “Communication” tab); select an appropriate node (usually the “Communications” tab and the Root node or a subnetwork node is a good choice).
2. Select the “Subnetwork” Editor:




3. In the Column “Subnetwork”, a Combo box offers all existing Subnetworks as choice:




4. Select the Subnetwork to which you want to add the IED/Access Point. To disconnect it, select the empty row.
5. After exiting the Combo Box, the Communication Tree will be updated, the IED (or, more precisely, its “Connected Access Point” can be seen under the new Subnetwork).

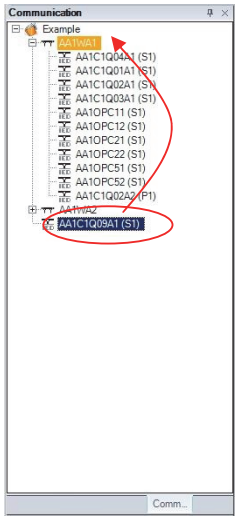
An IED can have several Access Points that can be connected to different Subnetworks. It is not allowed, however, to have an IED with two Access Points and connect them both to the same Subnetwork.

Use the “Fill Down” mechanism to rapidly connect multiple IEDs/Access Points to a Subnetwork. Use an appropriate sorting mechanism to avoid mapping two Access Points of the same IED to the same Subnetwork, e.g. by sorting first on the Access Point and the on the IED.

## 8.2.3 Connect IEDs to a Subnetwork in the Communication Tree

1. Select the “Communication” tab in the Navigation Panel.

2. Drag and drop an IED (or, more precisely, its “Connected Access Point”) to the Subnetwork you want to connect it to. (To disconnect it, you currently need to use the Subnetwork Editor. Drag-n-dropping it to the Project node will not work).




3. After exiting the Combo Box, the Communication Tree will be updated, the IED (or, more precisely, its “Connected Access Point”) can be seen under the new Subnetwork. Unconnected IEDs/Access Points appear directly under the root node.

## 8.2.4 Edit IP addresses

1. Select the “Subnetwork” Editor:

Drag a column header here to group by that column.

| IED Name | AP | IP Address | IP Subnet | MicroSCADA Unit Number | Subnetwork | Application | Edit Addresses |
|---|---|---|---|---|---|---|---|
| AA1D1Q01A1 | S1 | 0.0.0.0 | 255.255.0.0 | | | | Edit Addresses |
| AA1D1Q01TH1 | S1 | ... | ... | | | | Edit Addresses |
| AA1D1Q02A1 | S1 | 172.16.3.1 | 255.255.0.0 | 102 | AA1WF1 | 8-MMS | Edit Addresses |
| AA1D1Q03A1 | S1 | 172.16.3.1 | 255.255.0.0 | 103 | AA1WF1 | 8-MMS | Edit Addresses |
| AA1D1Q04A1 | S1 | 172.16.4.1 | 255.255.0.0 | 104 | AA1WF1 | 8-MMS | Edit Addresses |
| AA1D1Q05A1 | S1 | 172.16.5.1 | 255.255.0.0 | 105 | AA1WF1 | 8-MMS | Edit Addresses |
| AA1D1Q06A1 | S1 | 172.16.6.1 | 255.255.0.0 | 106 | AA1WF1 | 8-MMS | Edit Addresses |
| AA1D1Q02TH1 | S1 | ... | ... | 1102 | AA1WF2 | 8-MMS | Edit Addresses |
| AA1D1Q05TH1 | S1 | ... | ... | 1105 | AA1WF2 | 8-MMS | Edit Addresses |

Subnetworks

2. Enter the IP Addresses directly in the column “IP Address”. Duplicate addresses will be marked with a red circle (Exception: OPC servers can have the same addresses. Unfortunately, this is so, even if they belong to different IEDs! So here the engineer must be careful to avoid problems).
3. If you need to edit additional address information (e.g. OSI addresses), click on the button in the “Edit Addresses” column. A dialog will pop up which will allow editing additional address attributes. This should not normally be needed, though.

The “Fill Down” mechanism will increase the address by 1 in each field. This is usually not wanted, however. Currently there is no mechanism to automatically calculate the IP addresses according to a more sophisticated scheme.

# 9 Dataflow Engineering

## 9.1 Introduction

### 9.1.1 Overview

Dataflow is a general term for the functionality and configuration that is needed to exchange data between different IEDs. To configure the dataflow, the data model of the participating IEDs and the communication (the connection between the IEDs) must be more or less defined.

In IET600, there are several ways to engineer Datasets, RCBs and GCBs:

- create datasets, RCBs and GCBs manually, and then add/edit their data
- copy a configured dataset, RCB or GCB from the same or another project
- import a configured dataset, RCB or GCB from an IED SCL (CID/IID) file

### 9.1.2 Engineering Approach

The IEC 61850 standard distinguishes:

1. A Dataset that specifies what data are sent. It can basically be understood as a defined sequence of data items that can be sent as a whole packet only.
2. A Control Block that specifies how the data are sent. IEC 61850 defines:
   - Report Control Blocks (RCBs)
   - GOOSE Control Blocks (GCBs)
   - Sampled Value Control Blocks (SVCBs)
   - GSSE Control Blocks (obsolete, should not be used)
   - Log Control Blocks
   - Setting Group Control Blocks

IET600 stresses a slightly different approach. From an engineering point of view, it is helpful to distinguish:

1. Vertical communication from bay level IEDs to HMIs or NCC Gateways. The vertical communication includes:
   - configuring Datasets
   - configuring Report Control Blocks (RCB)
   - configuring Clients for the RCB in a matrix
2. Horizontal communication, essentially between IEDs with similar functionality:
   - configuring Datasetsconfiguring Goose Control Blocks (GCB)configuring receivers for GCB in a matrix

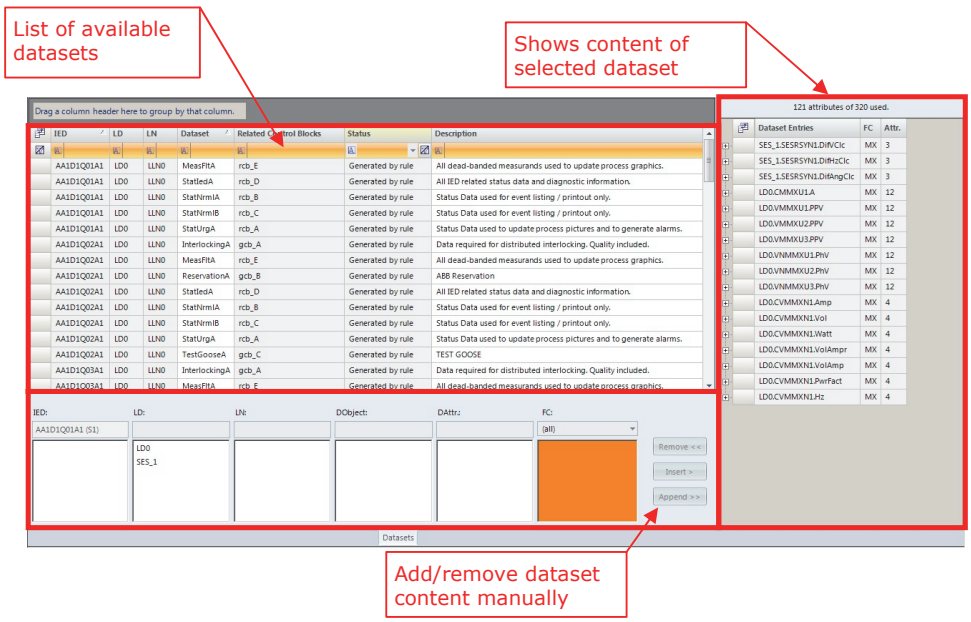
3. Analogue Data Communication (Sampled Values), essentially between merging units (MU) and IEDs:
   - configuring Datasets
   - configuring Sample Value Control Blocks (SVCB)
   - configuring receivers for SVCB in a matrix

## 9.2 Dataset Configuration

### 9.2.1 Datasets Editor

The Datasets editor provides the following views:

- all available datasets and its properties
- dataset entries of selected dataset
- Manual selection/removal of dataset entries



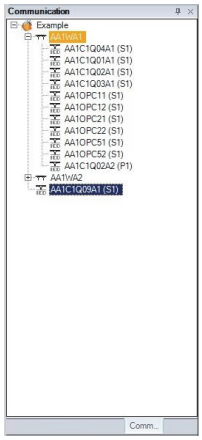

The following dataset properties are shown:

| Property | Description |
|---|---|
| Type | IED Type |
| IED | IED Name |
| AP | Access Point name |
| Srv | Server name |
| LD | Logical Device where RCB is defined |

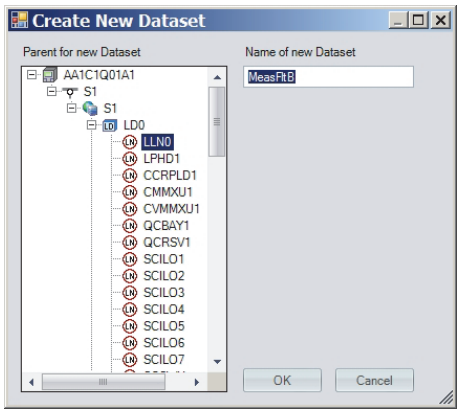
| | |
|---|---|
| LN | Logical Node where RCB is defined |
| Dataset | Name of dataset |
| Related Control Blocks | Reference to related RCB/GCB data |
| Status | Describes the method by which the dataset was created. IedDefinedFix or IedDefinedConfigurable are datasets which have been imported with SCL file. ManuallyConfigured are datasets are defined manually in IET. |
| Description | Text field which can be used to define purpose of dataset |

## 9.2.2 Create Datasets

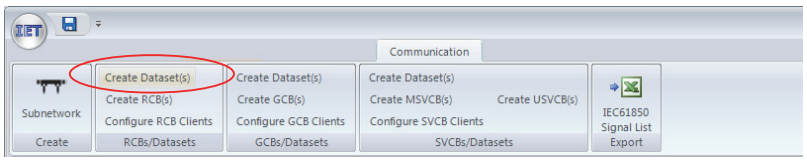
1. Select an appropriate IED in the ‘Substation’, ‘IEDs’ or ‘Communication’ Navigation Tree where the dataset shall be created. To have the full choice of IEDs, select the ‘Project’ node.




2. Select the Datasets Editor
3. Select a row and right-click to select “Insert new row” from the context menu.
4. A dialog “Create New Datasets” is opened. The Tree in the dialog will show all IEDs below the node you selected. Navigate to the IED and then to the LN where you want to create the new Dataset. For ABB IEDs, datasets are normally allocated under the node LD0/LLN0.




Alternatively, this dialog can also be opened from the Communication menu tab using the “Create Datasets” button:




5. Once you select an LN, the tool proposes a name, which you can adapt according to your wishes. Invalid names are indicated by an error mark (Invalid names typically are either duplicate names or contain characters that are not allowed for a particular node).
6. Click “OK” to create the new Dataset or “Cancel” to close the dialog without creating a Dataset.
7. If you created a new Dataset, the Navigation Tree will remain where it is and the Editor will change to “Dataset”.

## 9.2.3 Edit Datasets

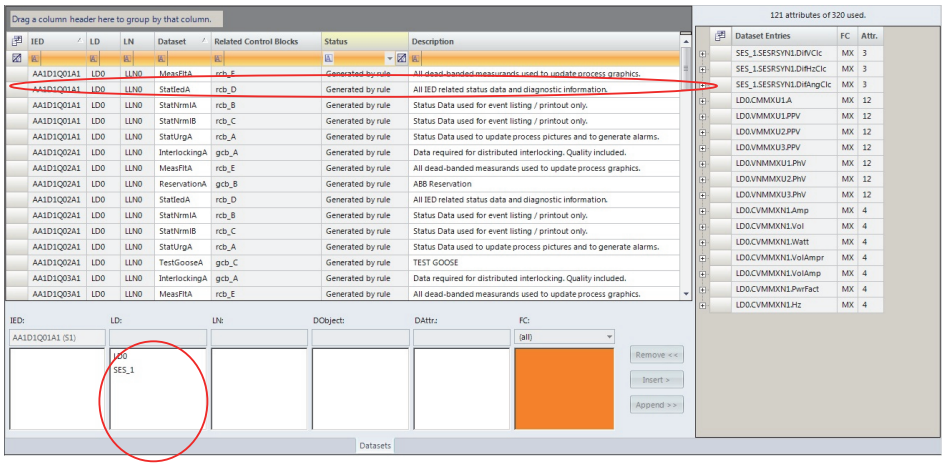
In the Dataset Editor, you can edit the name and the description of a Dataset. If you change the name, attached RCBs or GCBs will automatically be updated.

To attach a Dataset to a Control Block is currently not possible in the Dataset Editor; you have to use the RCB Editor or GCB Editor, respectively.

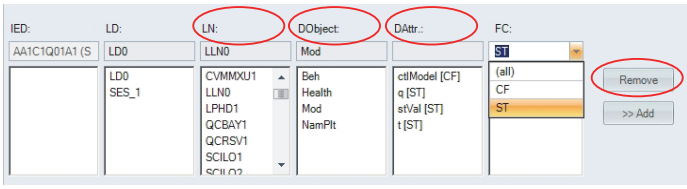
## 9.2.4 Add Dataset Entries

To add Dataset Entries, you need to use the lower part of the editor.

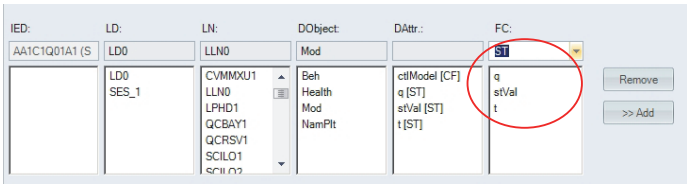
1. Select the Dataset to which you want to add entries.




2. The selected Dataset determines the IED. The first list shows the available LDs.
3. Select LD -> a choice of LNs appears. Continue selecting LN, Data Objects (DOIs) and Data attributes (DAIs) as appropriate (it is recommended to use Data Objects for RCB-Datasets and Data Attributes for GCB-Datasets).




4. If the functional constraint is not unique yet by your selection, select it now. You can only select FCs that make sense for spontaneously sent data (e.g. ST or MX).
5. In the “Data Result List, you see all the attributes that will be added to the Dataset. If it is empty, your selection is not valid yet (typically when the Data Object ahs attributes with several Functional Constraints - e.g. CF and ST - and you have not selected the FC yet).




6. After verifying the attributes in the result box, click “Add” to add your selection to the Dataset entries.

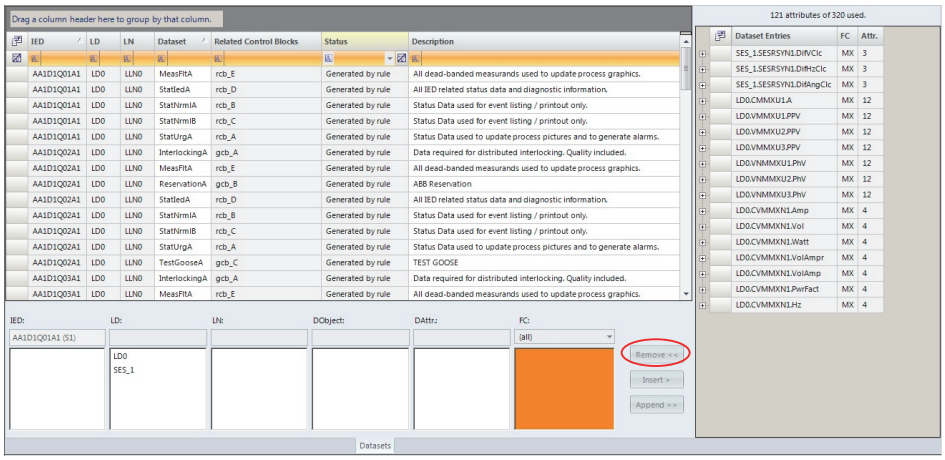
   It is not possible to add the same data twice to the same attribute, e.g. you cannot add CMMXU1.A[MX] twice. You can still add CMMXU1.A.phsA[MX] after having added CMMXU1.A[MX], although they partially contain the same data.

   It is not possible to add a whole LN, e.g. CMMXU1[MX].

## 9.2.5 Delete Dataset Entries

To delete Dataset Entries, you have two possibilities:

1. Select the entries you want to delete in the Dataset Entry part of the editor. Right-click and select “Delete Row(s)” or press the “Delete” key. A confirmation dialog appears, confirm the Deletion or cancel it.
2. Select the entries you want to delete in the Dataset Entry part of the editor. Click the Button “Remove” in the lower part of the editor. A confirmation dialog appears, confirm the Deletion or cancel it.




## 9.3 RCB Data Configuration

### 9.3.1 RCB Data Editor

The RCB Data editor defines the parameters for the reporting service, according to IEC 61850 part 7-2.

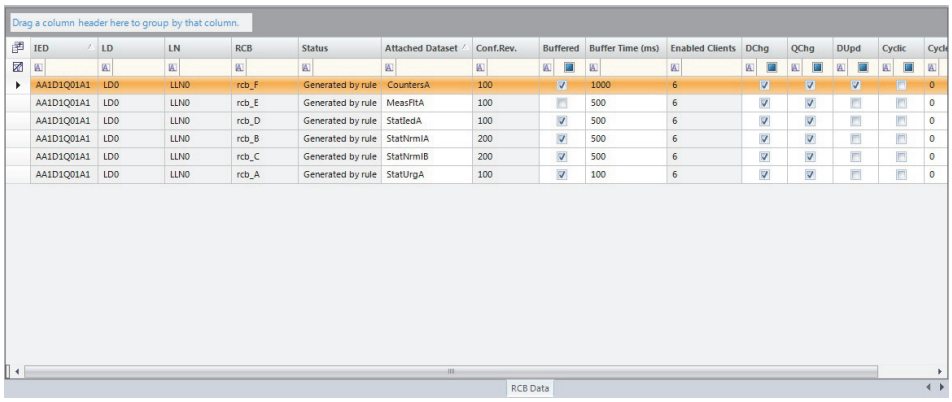
| **Property** | **Description** |
|---|---|
| IED Type | IED Type |
| IED | IED Name |
| AP | Access Point name |
| Srv | Server name |
| LD | Logical Device where RCB is defined |
| LN | Logical Node where RCB is defined |
| RCB | RCB name |
| Identifier (rptID) | RptID uniquely identifies a RCB per client on each IED. By default it is calculated from the full identification (path name) of the RCB. |
| Status | Describes how the RCB was created. This attribute is |

| | |
|---|---|
| | automatically generated by IET. |
| Attached Dataset | Select the name of the dataset, as specified in the Dataset Editor. |
| Conf.Rev. | Version number will be incremented on changes of the dataset related to the control block. Changes are the deletion or insertion of a member of the dataset, reordering of the member, or changing the dataset reference of the control block. Changes will increment this value by 10000. |
| Buffered | In general, only buffered reporting is used for all kinds of data, except for measurements. |
| Buffer Time (ms) | Defines the amount of time that the report is delayed (after a change) to wait for further changes before it is sent. |
| Enabled Clients | The number of instances or clients mapped to the RCB. This value is automatically calculated by IET based on the number of clients mapped in the RCB Clients editor. By default, there is always 1 additional RCB instance, unless number of clients = Max Enabled Clients. |
| DChg | TRUE specifies the condition to trigger reporting for monitored data is on *Data Change*. |
| QChg | TRUE specifies the condition to trigger reporting for monitored data is on *Quality Change*. |
| Dupd | TRUE specifies the condition to trigger reporting for monitored data is on *Data Update*. |
| Cyclic | TRUE specifies that the reporting for monitored data is *Cyclic*, according to the specified Cycle Time. |
| Cyclic Time (ms) | Cycle time for cyclic-triggered reporting |
| SeqNum | TRUE indicates the chronological order of reports is used in RCBs |
| BufOvfl | TRUE indicates that a flag to indicate Buffer Overflow for the BRCB is used. |
| Dataset | TRUE indicates usage |
| Entry ID | TRUE indicates usage |
| Entry Time | TRUE indicates usage |
| DataRef | TRUE indicates usage |
| Reason Code | TRUE indicates usage |
| Config Rev. | TRUE indicates usage |

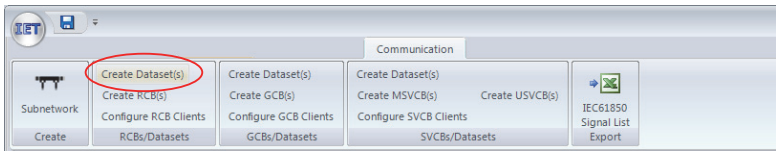
## 9.3.2 Create RCBs

1. If you know, where you want to create your RCB, you can preselect an appropriate node in the ‘Substation’, ‘IEDs’ or ‘Communication’ Navigation Tree. To have the full choice of IEDs, select the ‘Project’ node.

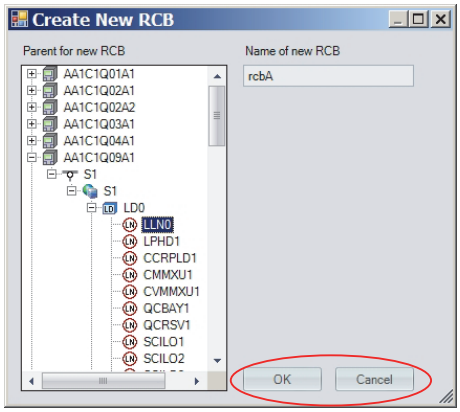
   Open the RCB Data editor and select a row in the RCB list, right-click and select “Insert new row” from the context menu.




   Alternatively, select the ‘Communication’ menu tab (the ‘Communication’ Navigation tab will automatically be selected) and select “Create RCB(s)”




2. The dialog “Create New RCB” opens. The Tree in the dialog will show all IEDs below the node you selected. Navigate to the IED and then to the LN where you want to create the new RCB. Note that for ABB IEDs, the RCBs are allocated to the LD0/LLN0 logical node.




3. Once you select an LN, the tool proposes a name, which you can adapt according to your wishes. Invalid names are indicated by an error mark (Invalid names typically are either duplicate names or contain characters that are not allowed for a particular node).

4. Click “OK” to create the new RCB or “Cancel” to close the dialog without creating an RCB.

5. If you created a new RCB, the Navigation Tree will remain where it is and the Editor will change to “RCB Editor”. The new RCB will have a standard configuration, which you may edit, if needed.

Remember: instead of creating new RCBs, a deleted RCB in the same LN can be reused (see below, chapter 9. 3. 5: Reuse “Deleted” RCBs).

### 9.3.3 Delete RCBs

An RCB once created cannot be renamed or totally removed anymore. The reason is that, once it is exported, the RCB and its last version number needs to be remembered. If this is not done, an RCB with the same name and the same version number, but different content and/or behaviour could be created at a later time. This would lead to confusion and even potentially dangerous situations.

So an RCB is “deleted” by setting its status to “deleted”. Deleted RCBs will remain in the tool forever, but will not be exported in SCD/ICD/IID files, and will not count when evaluating limits for RCBs.

1. Select the RCB Editor
2. Select any node that has RCBs below it in the Navigation Tree (the IED where you want to create it, is usually a good choice).
3. Select one or several rows with the RCBs intended for deletion.
4. Right-click and select “Delete Row(s)” or press the “Delete” key. A confirmation dialog appears, confirm the deletion or cancel it.
5. The RCBs will remain, but their Datasets will be removed and their status will go to “deleted”.

### 9.3.4 Edit RCBs

The RCB configuration can be edited in the RCB editor; normally this should not be necessary however. There are several specialties to be observed:

- An RCB cannot be renamed. To rename an RCB, delete it and create a new RCB with the intended name.
- Removing a Dataset from an RCB will automatically put the RCB into a “deleted” state.
- Configuration changes will cause the configRev value to be increased to the next multiple of 10’000. This will happen only once between import and export, not with every change.

### 9.3.5 Reuse “Deleted” RCBs

A deleted RCB can be reused by simply attaching a Dataset to it. Instead of creating a new RCB, it is recommended to reuse “deleted” RCBs first.

## 9.4 RCB Client Configuration

### 9.4.1 Prerequisites

To do a sensible RCB client configuration, the potential clients and their communication configuration should be known (typically HMIs and NCC Gateways). Therefore add these IEDs and configure them to the subnetworks before doing the RCB client configuration.

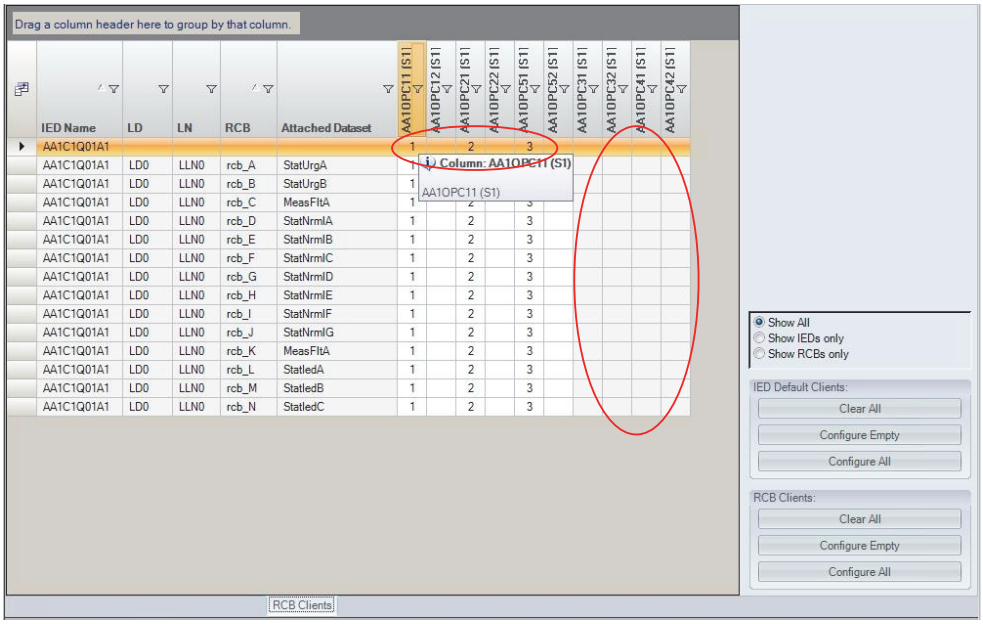
Bay-level IEDs can be added and configured easily also in later stages of the project.

### 9.4.2 RCB Client Editor

The rows of this editor show IEDs (or their Access Points, respectively) and RCBs. The rows depend on the node selected in the Navigation Tree

The columns show the available Client IEDs (and their Access Points). Valid clients are IEDs that have IHMI-, ITCI-, ITMI- or IARC-LNs. The columns do not depend on the node selected in the Navigation Tree (i.e. all Client IEDs are always shown). The client IED names are vertical to save space. To avoid neck strains, use the Tooltip to see the text horizontally.

If a Client IED is not on the same Subnetwork as a Server IED or RCB, it cannot be configured as a Client. In the Editor, this shows as greyed-out cells that cannot be edited.




For those cells that can be edited, you can use “+” or “*” or “x” to add an additional client to the existing ones. You can use numbers to change the client sequence or add clients. You can use “-“, the space bar or “Delete” to delete existing clients. The result will always show the clients as sequential numbers from 1 to the highest configured client.

The reason for this is that the IEC61850 standard does not allow you to reserve slots for later clients (e.g. to configure a sequence 1, 2, 4 with the intention to add a client 3 later). If you want to do that, you need to create a dummy client.

On the right, the editor has a set of Radio Buttons that allow fast filter setting (show either Default Clients or RCBs or both together; the latter is the default).

It also has some Buttons to allow fast, semi-automatic configuration of Default clients and RCB clients. The buttons operate only on the rows displayed in the editor, so by cleverly choosing objects in the Navigation tree and/or setting filters, one can do complex configurations for a big substation quite rapidly.

### 9.4.3 Basic Workflow for RCB Client Engineering

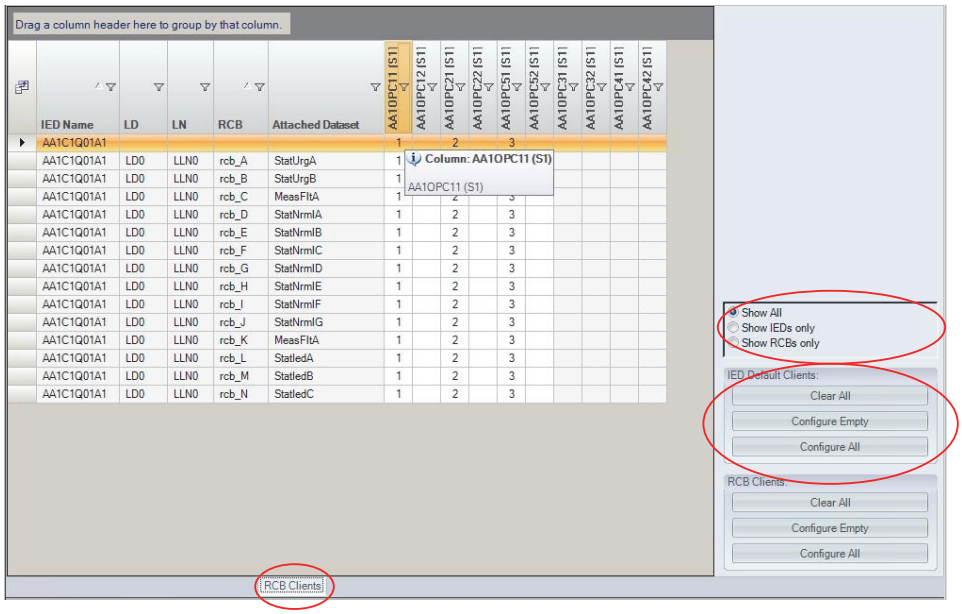
Configuring RCB Clients consists of two or three steps:

1. Configure the Default clients for each IED (or, more precisely, for its Access Points).
2. Propagate these defaults to all RCBs of the same IED/Access Point.
3. Configure individual clients for RCBs, if needed.

### 9.4.4 Configure Default Clients

For each IED, or rather, for each Access Point of an IED, Default clients should be configured. These “Default clients” will e.g. be used by the rule-based RCB generation to automatically configure clients. They can also easily be propagated to all RCBs in that IED/Access Point.

1. In the Navigation Tree, select any node that has IEDs/Access Points below it in the Navigation Tree (usually a Subnetwork, a Voltage Level or the whole Project is a good choice).
2. Select the RCB Client Editor.
3. If you have IEDs with more than one Access Points, make the column “AP” visible in the RCB Client Editor.
4. To have a better overview, you may choose to select “Show IEDs Only”.

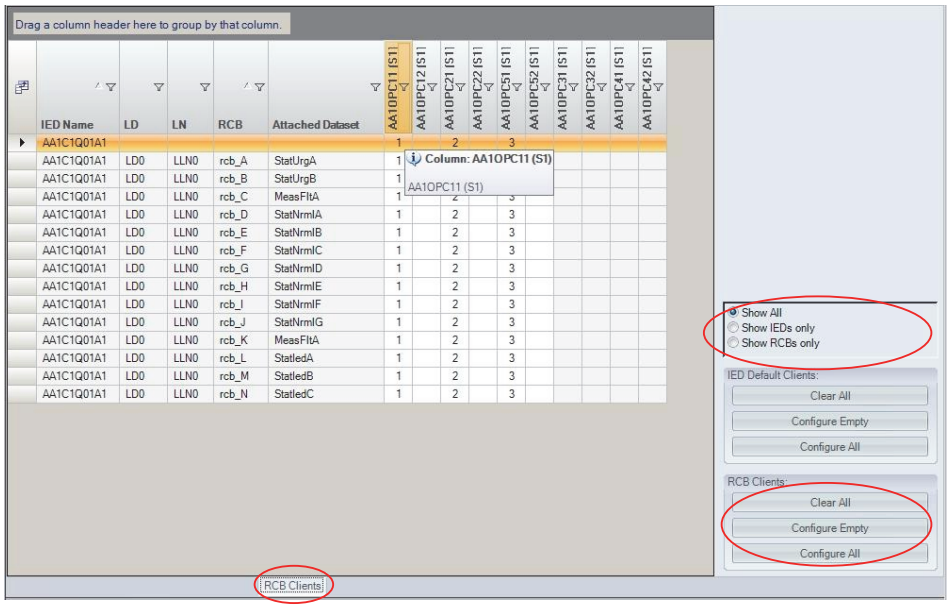



5. Configure the IEDs/Access Points with Default Clients:
   - “Clear All” removes all Default Clients.
   - “Configure Empty” automatically fills out only Default Clients for those IEDs/Access Points that have no clients configured yet.
   - “Configure All” deletes all existing Default clients and automatically fills them out afterwards.
   - Use the mechanisms described above (chapter 9. 4. 2: RCB Client Editor ), if you need to individually configure RCBs.

## 9.4.5 Configure RCB Clients

Precondition: you must have Default clients configured, otherwise the automatic RCB client configuration will not work.

1. In the Navigation Tree, select any node that has RCBs below it in the Navigation Tree (usually a Subnetwork, a Voltage Level or the whole Project is a good choice).
2. Select the RCB Client Editor.
3. If you have IEDs with more than one Access Points, make the column “AP” visible.
4. To have a better overview, you may choose to select “Show RCBs Only”.




5. Configure the RCBs with Clients.
   - “Clear All” removes all RCB Clients in the editor.
   - “Configure Empty” copies the Default Client Configuration of this IED/Access Point to its RCBs (only for those RCBs that have no clients configured yet).
   - “Configure All” deletes the existing RCB clients and then copies the Default Client Configuration of this IED/Access Point to its RCBs.
   - Use the mechanisms described in the previous chapter if you need to individually configure the clients of one or few RCBs.

## 9.5 GCB Data Configuration

### 9.5.1 GCB Data Editor

The GCB Data editor defines the parameters for the reporting service, according to IEC 61850 part 7-2.

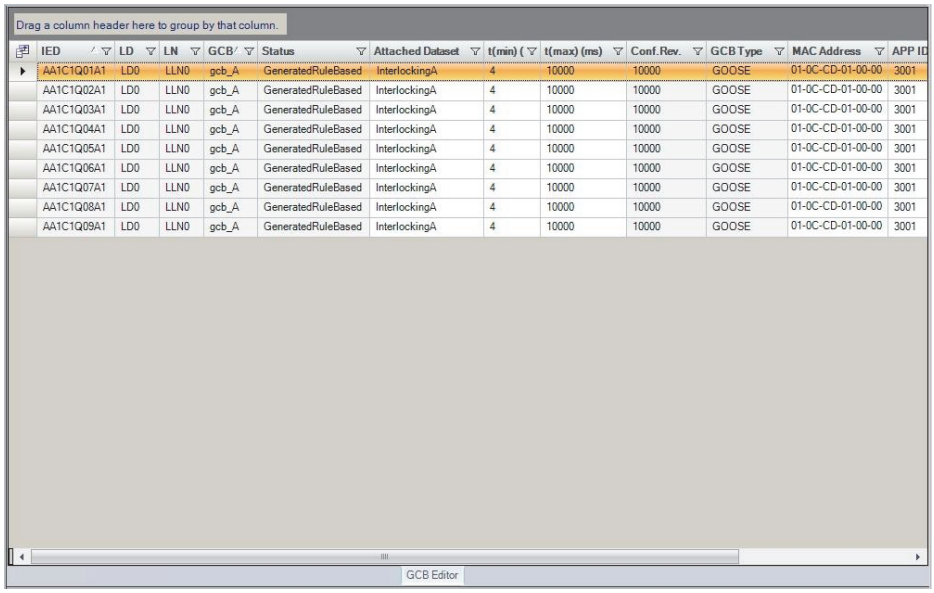
| Property | Description |
|---|---|
| IED Type | IED Type |
| IED | IED Name |
| AP | Access Point name |
| Srv | Server name |
| LD | Logical Device where GCB is defined |
| LN | Logical Node where GCB is defined |

| | |
|---|---|
| GCB | GCB name |
| Application (appID) | Identifies purpose or application of the GOOSE message. It should be unique for an application within the subnetwork. |
| Status | Describes how the GCB was created. This attribute is automatically generated by IET. |
| Attached Dataset | Select the name of the dataset, as specified in the Dataset Editor. |
| t(min) (ms) | Minimum cycle time for sending messages (e.g. after a change in dataset) |
| t(max) (ms) | Max cycle time for sending messages (e.g. used to detect missing or delayed messages) |
| Conf.Rev. | Version number will be incremented on changes of the dataset related to the control block. Changes are the deletion or insertion of a member of the dataset, reordering of the member, or changing the dataset reference of the control block. Changes will increment this value by 10000. |
| GCB Type | GOOSE |
| MAC Address | Identifies source and allow filtering of the GOOSE message. It should be unique for an application within the subnetwork. |
| APP-ID | Identifies source and allow filtering of the GOOSE message. It should be unique for an application within the subnetwork. |
| VLAN-ID | Controls the dataflow and real-time behaviour of the messages within a VLAN network. HEX value between 000-FFF is used, where value 000 (default) indicates that only the VLAN user priority is used. |
| VLAN Priority | Controls the dataflow and real-time behaviour of the messages within a VLAN network. Priority value between 0-7 is used, where higher numbers indicate higher priority message. 4 is recommended for GOOSE. |

### 9.5.2 Create GCBs

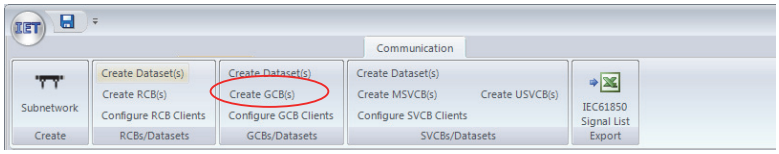
1. If you know, where you want to create your GCB, you can preselect an appropriate node in the ‘Substation’, ‘IEDs’ or ‘Communication’ Navigation Tree. To have the full choice of IEDs, select the ‘Project’ node.

   Open the GCB Data editor and select a row in the GCB list, right-click and select “Insert new row” from the context menu.

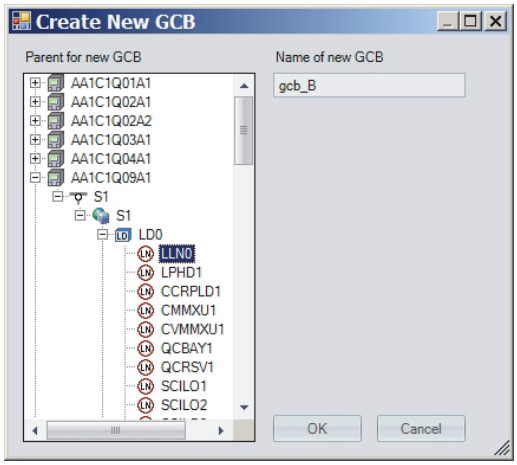



| IED | LD | LN | GCB | Status | Attached Dataset | t(min) ( | t(max) (ms) | Conf.Rev. | GCB Type | MAC Address | APP ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AA1C1Q01A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |
| AA1C1Q02A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |
| AA1C1Q03A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |
| AA1C1Q04A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |
| AA1C1Q05A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |
| AA1C1Q06A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |
| AA1C1Q07A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |
| AA1C1Q08A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |
| AA1C1Q09A1 | LD0 | LLN0 | gcb_A | GeneratedRuleBased | InterlockingA | 4 | 10000 | 10000 | GOOSE | 01-0C-CD-01-00-00 | 3001 |

Alternatively, you can select the ‘Communication’ menu tab (the ‘Communication’ Navigation tab will automatically be selected) and click on “Create GCB(s)”.




2. The Dialog “Create New GCB” appears. The Tree in the dialog will show all IEDs below the node you selected. Navigate to the IED and then to the LLN0 where you want to create the new GCB. Note that for ABB IEDs, the GCBs are allocated to the LD0/LLN0 logical node.




3. Once you select an LN (e.g. LD0/LLN0), the tool proposes a name, which you can adapt according to your wishes. Invalid names are indicated by an error mark (Invalid names typically are either duplicate names or contain characters that are not allowed for a particular node).

4. Click “OK” to create the new GCB or “Cancel” to close the dialog without creating a GCB.

5. If you created a new GCB, the Navigation Tree will remain where it is and the Editor will change to “GCB Editor”. The new GCB will have a standard configuration, which you may edit, if needed.

### 9.5.3 Delete GCBs

A GCB once created cannot be renamed or totally removed anymore. The reason is that, once it is exported, the GCB and its last version number needs to be remembered. If this is not done, a GCB with the same name and the same version number, but different content and/or behaviour could be created at a later time. This would lead to confusion and even potentially dangerous situations.

So a GCB is “deleted” by setting its status to “deleted”. Deleted GCBs will remain in the tool forever, but will not be exported in SCD/ICD/IID files, and will not count when evaluating limits for GCBs.

1. Select the GCB Editor
2. Select any node that has GCBs below it in the Navigation Tree (the IED where you want to create it, is usually a good choice).
3. Select one or several rows with the GCBs intended for deletion.
4. Right-click and select “Delete Row(s)” or press the “Delete” key. A confirmation dialog appears, confirm the deletion or cancel it.
5. The GCBs will remain, but their Datasets will be removed and their status will go to “deleted”.

### 9.5.4 Edit GCBs

The GCB configuration can be edited in the GCB editor; normally this should not be necessary however. There are several specialties to be observed:

- A GCB cannot be renamed. To rename a GCB, delete it and create a new GCB with the intended name.
- Removing a Dataset from a GCB will automatically put the GCB into a “deleted” state.
- Configuration changes will cause the configRev value to be increased to the next multiple of 10’000. This will happen only once between import and export, not with every change.
- Although it is not demanded by the standard, some vendors do not allow address combinations of the same MAC address with a different AppID or vice versa. We should follow an addressing scheme that always changes both together.

### 9.5.5 Reuse “Deleted” GCBs

A deleted GCB can be reused by simply attaching a Dataset to it.Instead of creating a new GCB, it is recommended to reuse “deleted” GCBs first.

## 9.6 GCB Client Configuration

### 9.6.1 Prerequisites

To do a sensible GCB client configuration, most of the Bay-level IEDs should be known and should be attached to the correct Subnetworks.

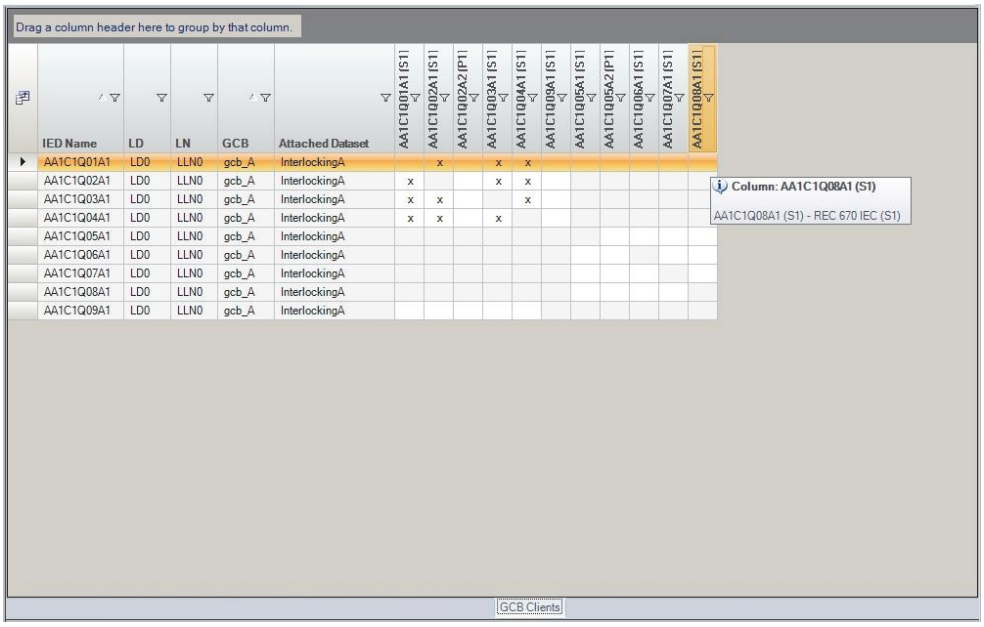
### 9.6.2 GCB Client Editor

The rows of this editor show GCBs (“Senders”). The rows depend on the node selected in the Navigation Tree.

The columns show the IEDs available as GOOSE clients (“Receivers”). To save space the client IED names are vertical. To avoid neck strains, use the Tooltip to see the text horizontally.

If the Client IED is not on the same Subnetwork as the GCB sender, it cannot be configured as a Client. In the Editor, this shows as greyed-out cells that cannot be edited.

Also an IED cannot be configured to send GOOSE to itself, this shows as greyed-out cells too.

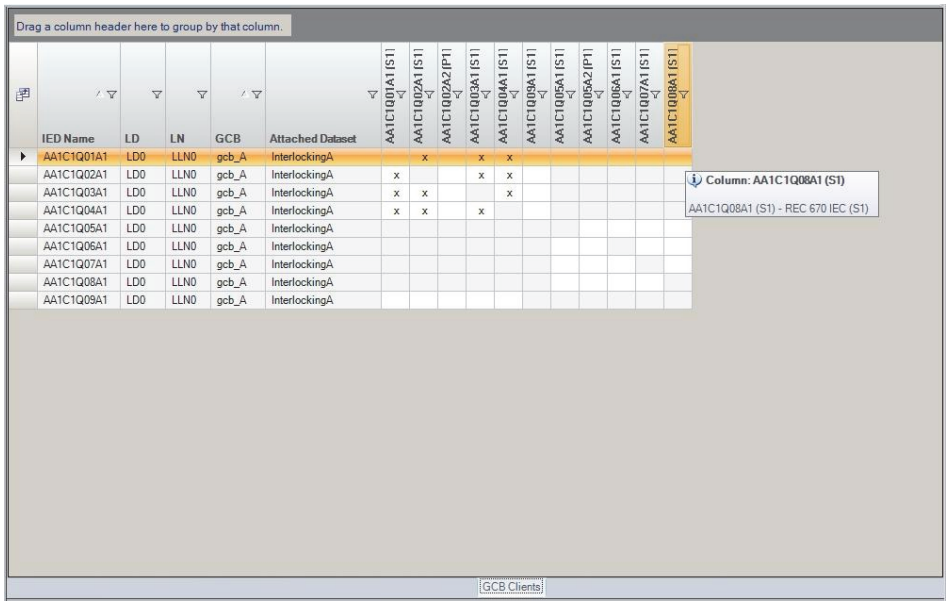



For those cells that can be edited, you can use “+” or “*” or “x” to add the IED in the column as a client to the GCB in the row. You can use “-“, the space bar or “Delete” to remove an existing clients.

Upon adding or removing clients, the corresponding input sections are updated automatically. Be aware that removing a client will remove all its inputs immediately (including their intAddrs that serve as a reference to the Signal Matrix in the PCM600 tool).

### 9.6.3 Configure GCB Clients

1. In the Navigation Tree, select any node that has RCBs below it in the Navigation Tree (usually a Subnetwork, a Voltage Level or the whole Project is a good choice).
2. Select the GCB Client Editor.
3. If you have IEDs with more than one Access Points, make the column “AP” visible.




4. Configure the GCBs with Clients:
   - Right-click and select “Remove all GCB Client connections” to remove all existing connections in the editor and all related inputs. (Do not use this action, if you have configured mappings in the Signal Matrix in PCM600, otherwise you will loose all those mappings).

     This action works only on the visible rows and visible columns in the editor. By cleverly selecting the node in the Navigation Tree, setting filters and changing the visibility of columns, you can fine-tune this cleanup procedure.
   - Right-click and select “Connect all GCBs to Clients”. This will connect all GCBs in rows to all Clients in columns, if allowed by the Subnetwork configuration.

     This action works only on the visible rows and visible columns in the editor. By cleverly selecting the node in the Navigation Tree, setting filters and changing the visibility of columns, you can fine-tune this fast mapping procedure.
   - Use the mechanisms described above in the previous chapter if you need to individually configure clients.

## 9.7 SVCB Data Configuration

### 9.7.1 SVCB Data Editor

The SVCB Data editor defines the parameters for the reporting service, according to IEC 61850 part 9-2.

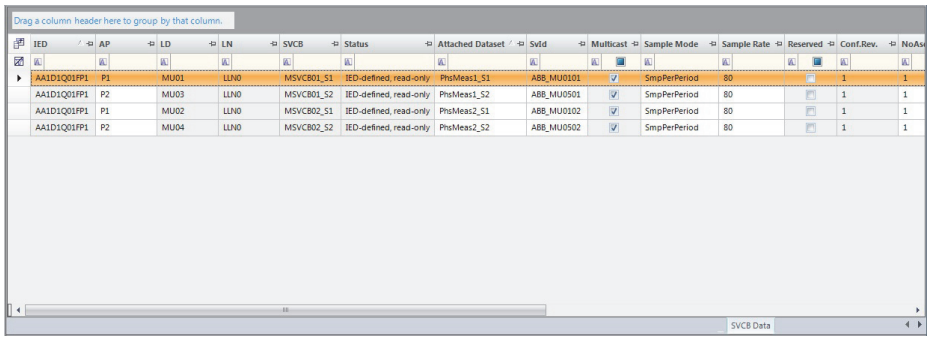
| **Property** | **Description** |
|---|---|
| IED Type | IED Type |
| IED | IED Name |
| AP | Access Point name |
| Srv | Server name |
| LD | Logical Device where GCB is defined |
| LN | Logical Node where GCB is defined |
| SVCB | SVCB name |
| Application (appID) | Identifies purpose or application of the SV message. It should be unique for an application within the subnetwork. |
| Status | Describes how the SVCB was created. This attribute is automatically generated by IET. |
| Attached Dataset | Select the name of the dataset, as specified in the Dataset Editor. |
| SvId | No default value, must be unique within the subnetwork. It is also recommended to make unique within substation to avoid reconfiguration when changing subnetworks. |
| Multicast | FALSE indicates that this is a Unicast control block, a maximum of one client IED shall be assigned to the instance |
| Sample Mode | Specifies if the sample rate is defined either in 1) “SmpPerPeriod” in units of samples per nominal period, 2) “SmpPerSec” samples per second, or 3) “SmpPerSmp” seconds per sample. If missing, default is 1) |
| Sample Rate | The value shall be interpreted depending on the value of Sample Mode |
| Reserved | TRUE indicates that SVCB is currently exclusively reserved for the client that has set the value to TRUE. Other clients shall not be allowed to set any attribute of that SVCB |
| Conf.Rev. | Version number will be incremented on changes of the dataset related to the control block. Changes are the deletion or insertion of a member of the dataset, reordering of the member, or changing the dataset reference of the control block, or any change of a value of an attribute of MSVCB. Changes will increment this value by 10000. |
| NoAsdu | |

| | |
|---|---|
| MAC Address | Identifies source and allow filtering of the SV message. It should be unique for an application within the subnetwork. |
| APP-ID | Identifies source and allow filtering of the SV message. It should be unique for an application within the subnetwork. |
| VLAN-ID | Controls the dataflow and real-time behaviour of the messages within a VLAN network. HEX value between 000-FFF is used, where value 000 (default) indicates that only the VLAN user priority is used. If not used it shall be set to 0 |
| VLAN Priority | Controls the dataflow and real-time behaviour of the messages within a VLAN network. Priority value between 0-7 is used, where higher numbers indicate higher priority message. |
| IncludeRefreshTime | True indicates usage |
| IncludeSampleRate | True indicates usage |
| IncludeSecurity | True indicates usage |

## 9.7.2 Create SVCBs

Preselect an appropriate node in the ‘Substation’, ‘IEDs’ or ‘Communication’ Navigation Tree containing the IED where the SVCB is to be created. To have the full choice of IEDs, select the ‘Project’ node.

1. Open the SVCB Data editor and select a row in the SVCB list, right-click and select “Insert new MSVCB” or “Insert new USVCB” from the context menu.




2. A new row is created in the editor. Select a dataset to the SVCB under the “Attached Dataset” field
3. Modify the remaining properties as required.

## 9.7.3 Delete SVCBs

A SVCB once created cannot be renamed or totally removed entirely from the project to maintain a consistent revision history of that SVCB. Once a SVCB has been exported, its last version number needs to be remembered by the system tool. If this is not done,

a SVCB with the same name and the same version number, but different content and/or behaviour, could be created at a later time, which would lead to confusion and even potentially dangerous situations.

So a SVCB is “deleted” by setting its status to “deleted”. Deleted SVCBs will remain in the tool forever, but will not be exported in SCD/ICD/IID files, and will not count when evaluating limits for SVCBs.

Select the SVCB Data editor

1. Select any node that has SVCBs below it in the Navigation Tree
2. Select one or several rows with the SVCBs intended for deletion.
3. Right-click and select “Delete Row(s)” or press the “Delete” key. A confirmation dialog appears, confirm the deletion or cancel it.
4. The SVCBs will remain, but their Datasets will be removed and their status will go to “deleted”.

### 9.7.4 Edit SVCBs

The SVCB configuration can be edited in the SVCB editor; normally this should not be necessary however. There are several specialties to be observed:

- A SVCB cannot be renamed. To rename an SVCB, delete it and create a new SVCB with the intended name.
- Removing a Dataset from an SVCB will automatically put the SVCB into a “deleted” state.
- Configuration changes will cause the configRev value to be increased to the next multiple of 10’000. This will happen only once between import and export, not with every change.

### 9.7.5 Reuse “Deleted” SVCBs

A deleted SVCB can be reused by simply attaching a Dataset to it. Instead of creating a new SVCB, it is recommended to reuse “deleted” SVCBs first.

## 9.8 SVCB Client Configuration

### 9.8.1 Prerequisites

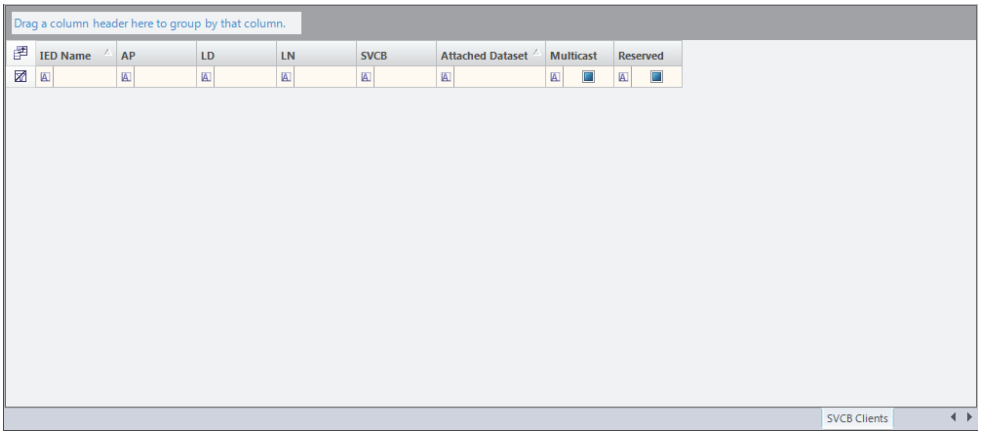
To do a sensible SVCB client configuration, most of the Bay-level IEDs should be known and should be attached to the correct Subnetworks.

### 9.8.2 SVCB Client Editor

The rows of the SVCB Client editor show SVCBs from the selected node in the Navigation Tree. The last columns in the editor shows the IEDs available as SVCB clients. To save space the client IED names are vertical. Use the Tooltip to see the text horizontally.

If the Client IED is not on the same Subnetwork as the SVCB sender, it cannot be configured as a Client. In the Editor, this shows as greyed-out cells that cannot be edited.

Also an IED cannot be configured to send SVs to itself, this shows as greyed-out cells too.




To send a SVCB to a receiving IED, enter “+” or “*” or “x” in the IED column at the end of the SVCB row. You can use “-“, the space bar or “Delete” to remove an existing clients.

Upon adding or removing clients, the corresponding input sections are updated automatically. Be aware that removing a client will remove all its inputs immediately (including their intAddrs that serve as a reference to the Signal Matrix in the PCM600 tool).

### 9.8.3 Configure SVCB Clients

1. In the Navigation Tree, select any node that has SVCBs below (usually a Subnetwork, a Voltage Level or the whole Project is a good choice).
2. Select the SVCB Client Editor.
3. If you have IEDs with more than one Access Points, make the column “AP” visible
4. Configure the SVCBs Clients:
   - Right-click and select “Remove all SVCB Client connections” to remove all existing connections in the editor and all related inputs. (Do not use this action, if you have configured mappings in the Signal Matrix in PCM600, otherwise you will lose all those mappings).

     This action works only on the visible rows and visible columns in the editor. By cleverly selecting the node in the Navigation Tree, setting filters and changing the visibility of columns, you can fine-tune this cleanup procedure.
   - Right-click and select “Connect all SVCBs to Clients”. This will connect all SVCBs in rows to all Clients in columns, if allowed by the Subnetwork configuration.

     This action works only on the visible rows and visible columns in the editor. By cleverly selecting the node in the Navigation Tree, setting filters and changing the visibility of columns, you can fine-tune this fast mapping procedure.

- Use the mechanisms described above in the previous chapter if you need to individually configure clients.

# 10 HMI Engineering

## 10.1 Introduction

### 10.1.1 Overview

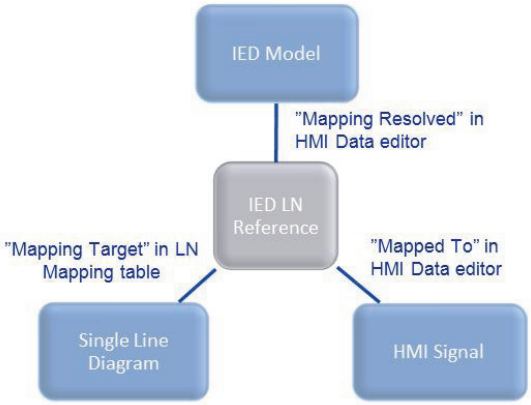
HMI data are basically organized as “signals”; a signal can be understood as what appears on one line in an Event List or Alarm List.

HMI signals can be mapped to an IEC 61850 IED as well as to process data in the single line diagram. The advantages of this approach are:

- Signals can be configured (and exchanged with the HMI configuration tool), before actually knowing the detailed data structure of an IED.
- Process data can be deleted or changed without loosing the respective HMI information.

This concept allows station-level engineering and bay-level engineering to be completed in parallel, and later merged automatically by IET600.

The relation between the HMI signals, the IED model and SLD process data can be depicted as follows:




The attribute “Mapped To” in the HMI Data editor provides the reference to the IED model, whether real or virtual. A virtual reference means that this reference has not been mapped to an IED model. This “virtual” reference can be updated by the real IED model at a later stage in engineering, at which point the reference becomes “resolved”. The status of a HMI signal reference can be seen in several views: in the LN Mapping table of the SLD or in the column “Mapping Resolved” in the HMI Data editor.

In the SLD editor in the LN Mapping table under the column “Status”. This field shows either

- “virtual” for signals which has not been mapped to an IED in the project but have a reference to an HMI signal,
- “resolved” for signals which are mapped to an IED or

- “unresolved” for LN references which has not been mapped to an IED nor to an HMI signal (this situation may occur if a signal and IED model has been deleted)

In the HMI Data editor under the column “Mapping Resolved”. If this field is not selected, this means that the reference in the column “Mapped To” has not been mapped to an IED model in the project.

## 10.2 Create HMI Client

### 10.2.1 Overview

IET600 allows engineering multiple HMI Clients and configuring different signals or different signal configurations for each HMI. Furthermore it allows configuring Hot-Standby Systems. All configuration of these HMI client IEDs is done in the HMI Navigation Tree.

IET600 allows the configuration of Hot-Standby systems. They consist of an HSB node with two or more related HMIs. Signals of all HMIs participating in one HSB are identical, while the communication configuration of each HMI is different.

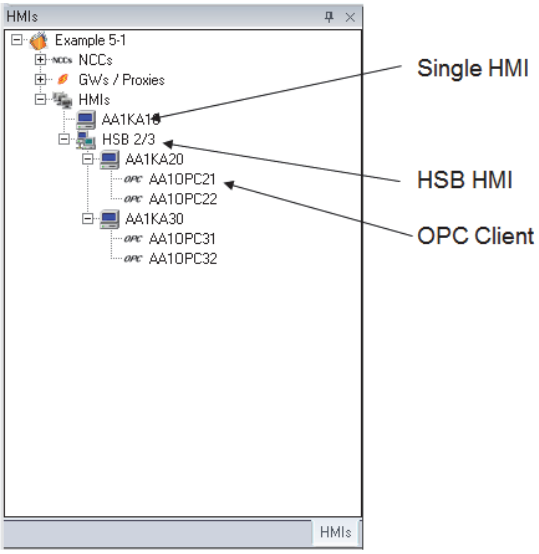
On a Substation level, there are 2 types of Client IEDs that you can configure:

- HMIs
- NCC Gateways

All these IEDs are created in the “HMIs” tree of the Navigation Panel. This chapter deals with the creation and configuration of HMIs as Client IEDs.

There are 3 types of HMI IEDs

- OPC servers (actually a software component, one or several of which can run in an HMI).
- HMIs (a physical IED).
- HSBs (a cluster of several HMIs that are redundant and have identical HMI Signals, but different communication configurations).




OPC servers count as IEDs only for historical reasons. They always need to be children of an HMI in the HMI tree but appear as IEDs in the IED tree.

As multiple HMIs are possible, it is important to be aware how the HMI signals are presented:

- A Bay-Level IED in the IED tree may send different signals to different HMIs. The IED has a collection of signals for each HMI it is sending signals to. When selecting a Bay-Level IED in the IED tree; the user needs to choose one HMI to display the signals that this particular Bay-Level IED is sending to this particular HMI. So in this case, it is a server's or sender's view (which process signals are sent from this IED).
- When selecting an HMI in the IED or HMI Navigation tree, you see all the process signals that belong to this particular HMI. This is a client's or receivers view (which process signals are received by this particular HMI).

As in the former case, it is easy to forget to select the proper HMI in case of multiple HMIs, the latter is the recommended approach to edit HMI signals.

The following icons are used in the HMI navigation tree to represent the various HMI and GW / Proxy types:

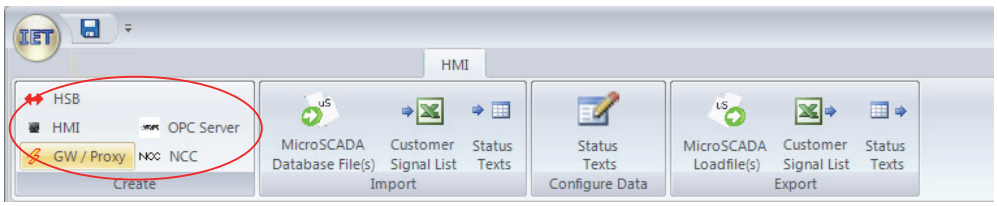
| **Icon** | **Description** |
|---|---|
| | Single HMI |
| | Gateway / proxy with integrated HMI |
| | HSB HMI |
| | HSB GW / proxy with integrated HMI |
| | Standalone GW / proxy |

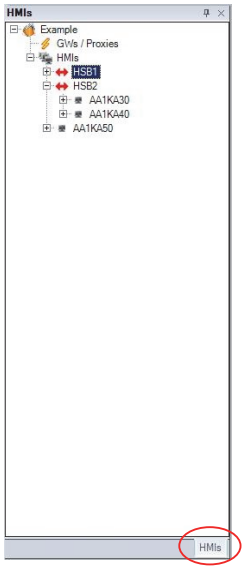
| | |
|---|---|
| | Single NCC |
| | HSB NCC |

The next chapters will describe how to configure the HMI clients in IET600. Gateways and NCC configuration will be described in subsequent chapters.

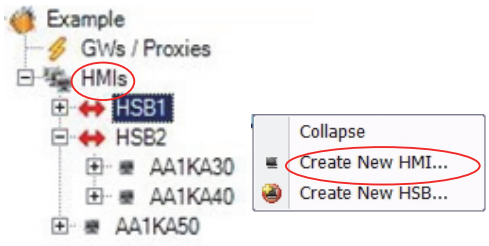
## 10.2.2 Create an HMI

An HMI can be created either from the HMIs navigation panel or the HMI menu. The steps will be shown using the HMI Navigation panel. However, the HMI devices can also be created from the HMI Menu.

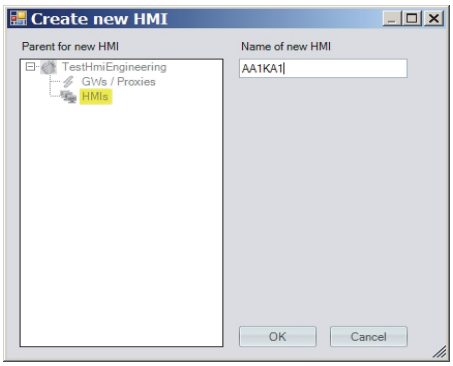



1. Select the “HMIs” Tab in the Navigation Panel or the HMI menu
2. Select the “HMIs” node, and right-click the node -> the Context Menu for that node appears. Select “Create New HMI …”




3. The Dialog “Create New HMI” appears.




An HMI can only be created below the “HMIs” Node which should be pre-selected in the Dialog Tree

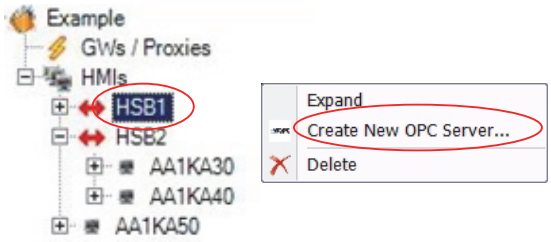
4. The tool proposes a name, which you can adapt according to your wishes. Invalid names are indicated by an error mark (Invalid names typically are either duplicate names or contain characters that are not allowed for a particular node).
5. Click “OK” to create the new HMI or “Cancel” to close the dialog without creating an IED.

Although not mandatory, it is recommended to add some HMI parameters now, e.g. configure the MicroSCADA OI length in the “Properties” Window.

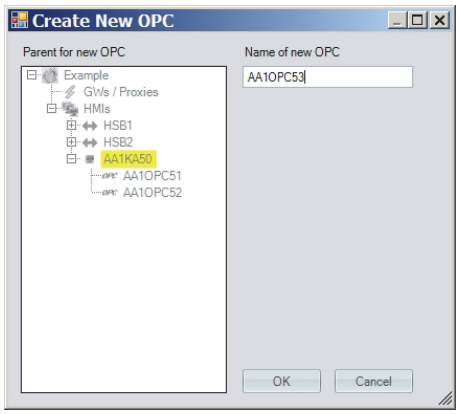
## 10.2.3 Add OPC Servers to an HMI

An OPC Server can be added to an HMI either from the HMI navigation panel of the HMI Menu.

1. Select the “HMI” Tab in the Navigation Panel or the HMI menu
2. Select an HMI (or HSB) node
3. Right-click the node -> the Context Menu for that node appears.



4. Select “Create new OPC Server …”
5. The Dialog “Create New OPC Server” appears.




6. The tool proposes a name, which you can adapt according to your wishes. Invalid names are indicated by an error mark (Invalid names typically are either duplicate names or contain characters that are not allowed for a particular node).

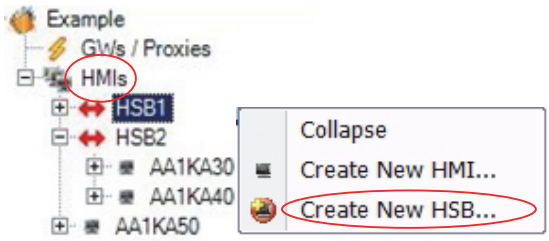
   Be aware of one constraint: the OPC server name must be unique within the project. So it is currently not possible to create an HMI1 with OPC servers AA1OPC1 and AA1OPC2, and then have an HMI2 with OPC servers AA1OPC1 and AA1OPC2 as well. The latter OPC servers need to be named differently, e.g. AA1OPC3 and AA1OPC4.
7. Click “OK” to create the new OPC Server or “Cancel” to close the dialog without creating an OPC Server.

   Although not mandatory, it is recommended to add some communication parameters now, i.e. configure the IP-Address and attach the OPC server to a subnetwork according to chapter 8. 2. 2: Connect IEDs to a Subnetwork in the Subnetwork Editor.

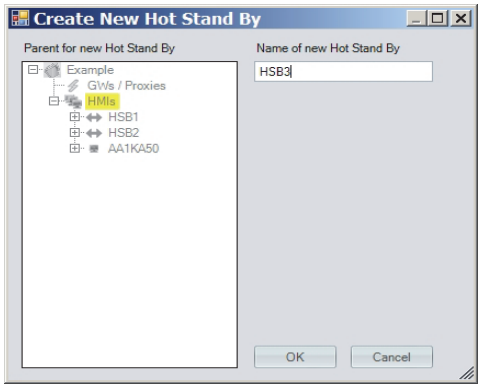
### 10.2.4 Create a Hot-Standby System (HSB)

An HSB system can be added from the HMI navigation panel of the HMI Menu.

1. Select the “HMI” Tab in the Navigation Panel.
2. Select the “HMIs” node.
3. Right-click the node -> the Context Menu for that node appears. Select “Create New HSB …”

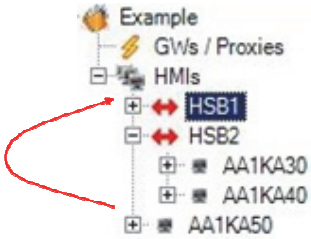



4. The Dialog “Create New Hot Stand By” appears:




(An HSB can only be created below the “HMIs” Node which should be pre-selected).

5. The tool proposes a name, which you can adapt according to your wishes. Invalid names are indicated by an error mark (Invalid names typically are either duplicate names or contain characters that are not allowed for a particular node).
6. Click “OK” to create the new HSB or “Cancel” to close the dialog without creating an IED.
7. After creating the HSB node, create at least 2 HMIs as described above (see chapter 10. 2. 2: Create GW / Proxy with integrated HMI). Drag and drop the HMIs onto the HSB node.




If the HMI nodes were created earlier and already contain HMI signals, the HMI signals of the first HMI will be used for the HSB, the signals of the subsequent HMIs will be thrown away).

## 10.3 Configure HMI Status Texts

### 10.3.1 Overview

We use the term “Status Texts” to describe the part of a signal that describes the process state, e.g. “on/off” or “intermediate/open/close/faulty”.

To imitate the mechanism of most current HMIs, Status Texts have been implemented as follows:

- Each signal has a reference to a status text, the “key”. E.g. the MicroSCADA EH attribute references Status Texts.
- Depending on the signal type, the signal can take different values (2 in case of a binary value or binary command, 4 in case of a double binary, an irregular number in case of analogue values). So, Status Texts come in groups; the number of items in such a group is determined by the signal type it refers to.
- A collection of Status Texts exists in the project, independent of HMI Signals. If HMI signals are created or imported, the Status Texts are connected via its key, if it exists in the import. Otherwise they have to be configured manually.

So sometime very early in the project, before configuring any HMI signals, you need to import such a Status Text collection.

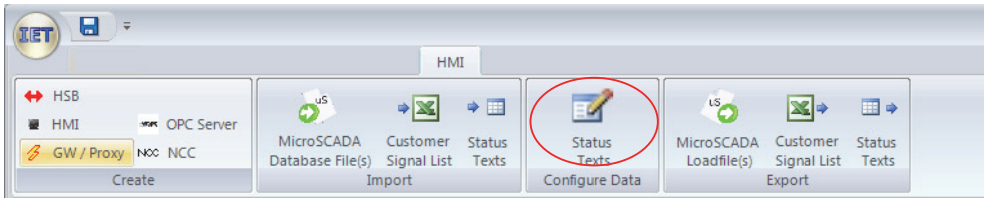
To import and export the HMI Status text from IET to MicroSCADA, see chapter 13. 2. 2 and chapter 13. 2. 3, respectively.

### 10.3.2 View/Edit Status Texts

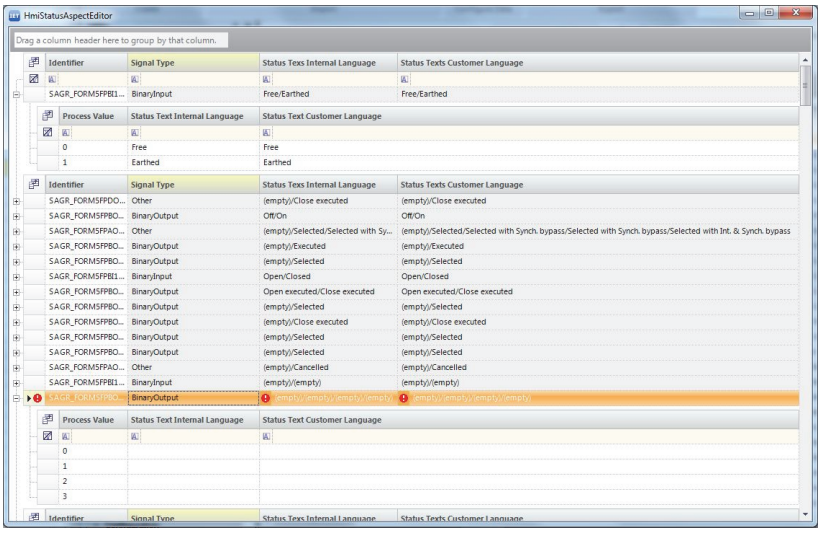
Imported Status Text from MicroSCADA can be edited and re-exported from IET600 and imported into the MicroSCADA application, thereby ensuring that changes to the Status Text during IET600 engineering will be consistent to the MicroSCADA project.

To view the Status Text which have been imported into IET:

1. select the HMI menu and under Configure Data, click “Status Text”:




2. The following dialog appears:




## 10.4 Create HMI Signals

### 10.4.1 Overview

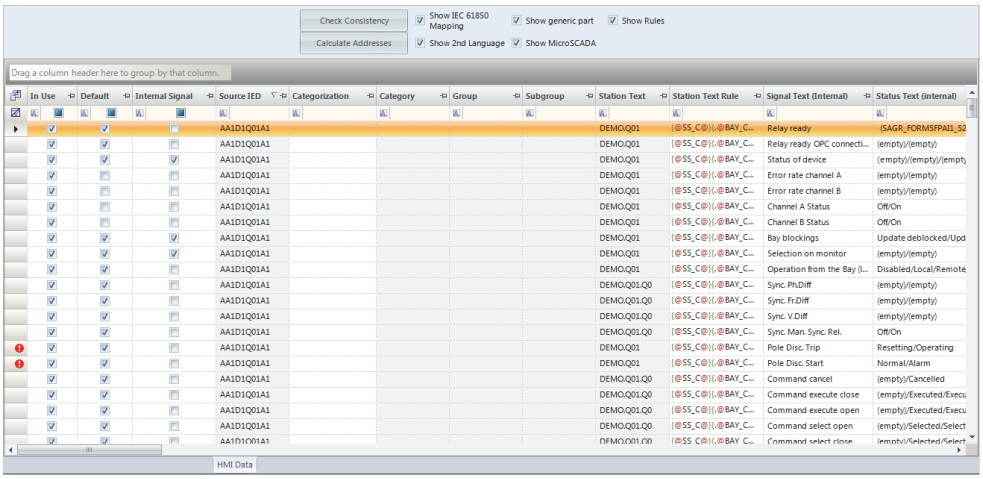
HMI signals are created and configured in the HMI Data editor, as described in chapter 10.6.2: HMI Data Editor. There are several ways to create HMI signals:

- Copy HMI signals from other IEDs or other HMIs
- Import a MicroSCADA Loadfile (LOF file) or CSV file
- Create HMI signals manually

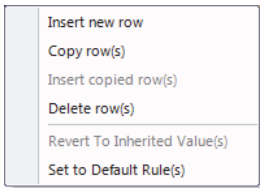
### 10.4.2 Copy HMI Signals from another IED

This is basically done as a simple copy (the selected signals from one IED) and paste (these signals to another IED).

1. Select the Substation, IED or Communication tab (do not use the “HMI” Tab) in the Navigation Panel.
2. Select the IED whose signals you want to copy.
3. Select all the rows with the signals you want to copy (from the leftmost “row selector” column of the editor).




4. Right-click in the editor -> the Context Menu for the editor appears.




5. Select “Copy” or “Copy row(s)” -> all the selected rows will be copied.
6. Select the another IED (destination for signals) in the navigation tab
7. Right-click in the HMI editor -> the Context Menu for the editor appears.
8. Select “Insert copied rows”
9. The signals are inserted to this IED

Note that when copying and pasting an IED, its HMI signals are copied along.

## 10.4.3 Copy HMI Signals from another HMI

Again, this is basically done as a simple copy (the selected signals from one HMI) and paste (these signals to another HMI).

1. Select the “HMI” tab in the Navigation Panel.
2. Select the HMI from which signals you want to copy.
3. Click on the HMI editor and select all the rows with the signals you want to copy (from the leftmost “row selector” column of the editor).
4. Right-click in the editor -> the Context Menu for the editor appears.
5. Select “Copy row(s)”.
6. Select the destination HMI from the navigation tab
7. Select “Insert copied row(s)”.The signals are inserted to this HMI. The mapping references are kept, so the source IED is the same.

### 10.4.4 Import a MicroSCADA Database Loadfile (LOF or CSV file)

To engineer a new project from scratch, one should work with HMI signals from a similar project.

For retrofits or upgrades of old HMIs however, one should import the actual data from the existing HMI.

- for MicroSCADA versions 8.x, it is recommended to export a LOF file from MicroSCADA and use this for import into IET600 .
- for MicroSCADA versions 9.x and newer, it is recommended to use a CSV Export from the Database (described below).
- Only if neither is feasible, use a LOF-file export from an IET 4.x project. Be aware, though, that this may not represent the actual state of the MicroSCADA process database, as customers may have made modifications directly in MicroSCADA.

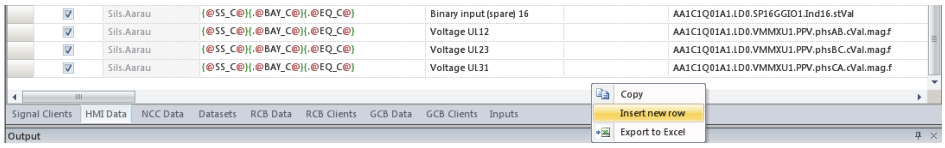
A LOF-file import takes advantage of existing IN attributes to map imported HMI signals to current IED data. To take full advantage of this mapping mechanism, the Substation and bay-level IED data should already exist (e.g. from an SCD import) and LN mapping should be done before a LOF file import.

For details of LOF file import, see chapter 13. 3. 2: Import MicroSCADA Database file (LOF or CSV file).

### 10.4.5 Create HMI Signals manually

HMI / MicroSCADA signals can also be created manually in IET600.

1. Select an IED node and open the HMI Data editor
2. Scroll to the bottom of the grid editor and right click in the editor area to open the context menu → select “Insert new row”




3. A new row is created with default HMI / MicroSCADA properties.

## 10.5 Select signals for HMI client

### 10.5.1 Overview

Once HMI signals have been created for a particular IED, they can now be “mapped” to the HMI clients. Mapping to the HMI clients allows IET600 to generate the corresponding loadfiles for that HMI, for example for MicroSCADA loadfiles.

### 10.5.2 Signal Clients Editor

The Signal Clients Editor allows provides the mapping of the available signals in the project to a specific station-level client, which could be an HMI or GW / Proxy.

Station signals and its properties

Client selection

| ProcessObjectType | StationText | SignalTextLanguageInternal | StatusText | LN | IX | AA1KA10 | HSB 2/3 |
|---|---|---|---|---|---|---|---|
| BinaryInput | Sils.Aarau | Close interlocked | | C1Q01QA1 | 17 | x | |
| BinaryInput | Sils.Aarau | Open interlocked | | C1Q01QA1 | 16 | x | |
| BinaryInput | Sils.Aarau | Close interlocked | | C1Q01QB1 | 17 | x | |
| BinaryInput | Sils.Aarau | Open interlocked | | C1Q01QB1 | 16 | x | |
| BinaryInput | Sils.Aarau | Close interlocked | | C1Q01QB2 | 17 | x | |
| BinaryInput | Sils.Aarau | Open interlocked | | C1Q01QB2 | 16 | x | |

This editor also used for NCC engineering, as described in chapter 12.5.2: Signal Clients Editor.

### 10.5.3 Select Signals to be Sent to HMI

Use the Signal Clients editor to define the signals to be sent to the HMI client.

1. Select an IED node and open the Signal Clients editor
2. Find signal row(s) which shall be sent to the HMI (these signals should be generated in the HMI loadfile)
3. Find the column on the right side labelled with the HMI name and mark the cell with an “x” or double click the cell

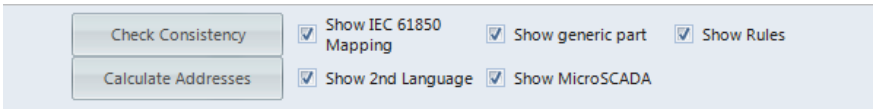
In most cases, one would send all of the signals to the HMI. To do this, use the “Fill Down” function as described in chapter 4.3.2: Editing functions.

## 10.6 Configure HMI data

### 10.6.1 Overview

An HMI signal has two parts: a generic part that is useful for any HMI and a MicroSCADA-specific part that is useful for MicroSCADA engineering only. Currently, an export is possible only for MicroSCADA data (see chapter 13. 3: HMI Signal Information); later it is planned to have an export of generic HMI data for configuration of other HMIs like 800xA or RTU560.

In the HMI Signals Editor, you can select whether you want to see the generic part or the MicroSCADA part or both by selecting the checkboxes shown above the grid.



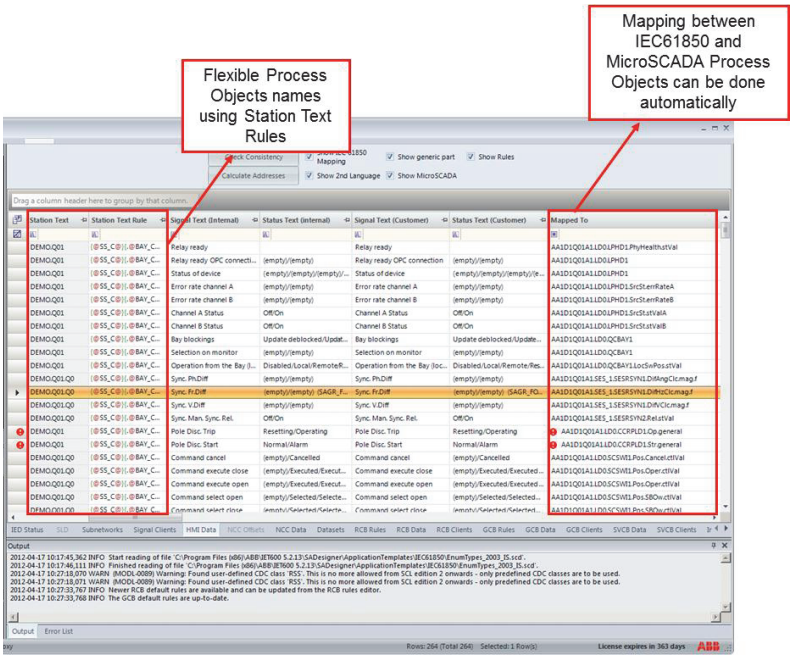

The HMI data engineering consists of configuring the individual HMI (MicroSCADA) signal properties. These properties can be set in the HMI Data editor.

Moreover, HMI signals are attached to IEDs in the project. This is achieved via a mapping to the IEC 61850 model of the IED, as described in chapter 10.6.5: Mapping signals to IEC 61850.

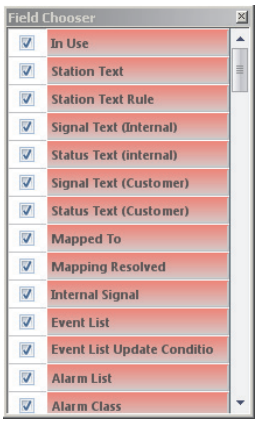
## 10.6.2 HMI Data Editor

The HMI Data Editor provides the signal specification or MicroSCADA properties of each signal. This grid editor includes configuration of:

- Generic HMI signals – a generic description of the signal and its behaviour
- MicroSCADA signals – MicroSCADA Process Object attributes
- IEC 61850 Mapping – mapping of signals to the IEC 61850 model of an IED
- HMI Rules – rule-based configuration for several HMI attributes
- Signal and status text in multiple languages




To view all available columns in the HMI Data editor, click on the Field Chooser icon  to view all available columns




### 10.6.3 Generic HMI Signals

To view the generic HMI signals in the HMI Data editor, select “Show generic part” in the header. Most of these generic HMI signals correlate to specific MicroSCADA attributes, as shown in the table below:

| **Generic Signal Attributes** | **Equivalent MicroSCADA Attribute** |
|---|---|
| In Use | IU |
| Station Text | n/a* |
| Signal Text (Internal) | OX |
| Status Text (Internal) | (Derived from EH) |
| Internal Signal | n/a* |
| Event List | HE |
| Event List Update Condition | HA |
| Alarm List | AL |
| Alarm Class | AC |
| Alarm Delay | AD |
| Printer | n/a* |
| Printer Update Condition | PA |

Signals marked with n/a* indicate that these generic signal attributes do not have a direct relation to a MicroSCADA attribute. They are however useful for identifying the signal within the project.

### 10.6.4 HMI Text Rules Engine

IET600 provides a flexible rule-based engine for the configuration of the following HMI properties:

- LNRule: rule-based configuration of MicroSCADA LN attribute
- OIRule: rule-based configuration of MicroSCADA OI attribute
- INRule: rule-based configuration of MicroSCADA IN attribute
- ONRule: rule-based configuration of MicroSCADA ON attribute
- Station Text Rule: rule-based configuration of generic signal

The rule engine utilizes the following syntax to compile a station text:

| **Variable** | **Description** |
|---|---|
| SS | Name of Substation |
| SS_C | Customer Name of Substation |
| VL | Name of voltage level |
| VL_C | Customer Name of voltage level |
| BAY | Name of bay |
| BAY_C | Customer Name of bay |
| FN | Function Name (not used) |
| SFN | Sub-function Name (not used) |
| EQ | Name of equipment |
| EQ_C | Customer Name of equipment |
| SN | Name of subnetwork |
| IED | Name of IED |
| IED_C | Customer Name of IED |
| LDInst | IEC 61850 Logical Device instance |
| LN | IEC 61850 Logical Node name |
| DARef_IN | IEC 61850 Data attribute reference |

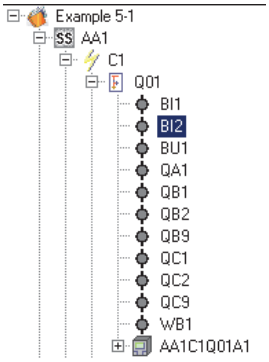
The above variables can be constructed into an expression which defines the text rule, with the help of the following syntaxes:

| Variable | Description |
|---|---|
| {expression} | Contains expression |
| @variable@ | Contains variable |
| variable! | Must include variable (mandatory) |
| \| | OI Separation (for including spaces in OI attribute) |

For example, for the station shown below, the

LNRule : {@SS@}{@VL@}{@BAY@}{@FN@}{@SFN@}{@EQ@}

LN: AA1C1Q01BI2




Note that you may also concatenate text strings to the above expressions. For example, if a string “Demo” is added to the end of the LNRule:

LNRule : {@SS@}{@VL@}{@BAY@}{@FN@}{@SFN@}{@EQ@}Demo

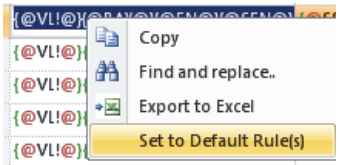
LN: AA1C1Q01BI2Demo

Some further examples are shown below:

| | |
|---|---|
| {.@EQ_C@} | Dot will be the prefix of variable IED, e.g. “.REC670”. If variable name is empty, the dot will not be visible |
| {@BAY_C@}.{@EQ_C@} | Dot will always be shown, no matter if variables exist |

Note that when IET600 cannot resolve a mandatory rule (using “!” syntax), then the signal row will be marked with a . These rules must be resolved by the user, either by satisfying the rule or changing it.

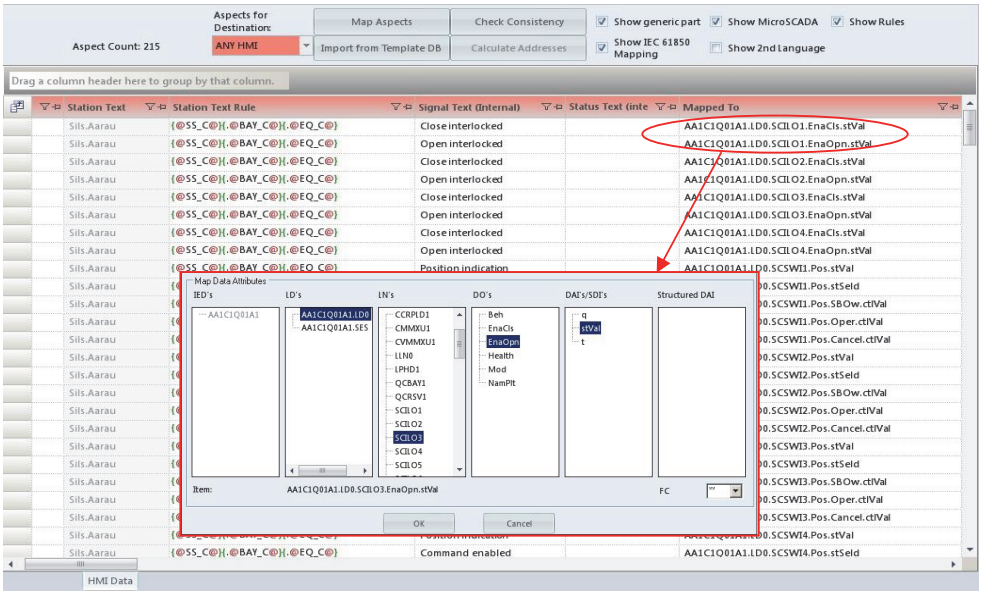
Alternatively, the text rule field can be set to a default rule by right-click on the selected rule field and select “Set to Default rule(s)”:




## 10.6.5 Mapping signals to IEC 61850

The “Mapped To” property for each HMI signal provides the link or association to the IEC 61850 model of an IED. When the IED has been mapped to the primary equipment (as described in chapter 7.5: Map LNs to Substation Elements (“LN Mapping”)), IET600 can automatically assign several HMI and MicroSCADA properties according to this mapping.

To map the signal, click on the Mapped To cell to open the Map Data Attributes dialog:



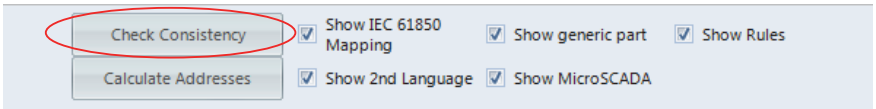

The Map data Attributes dialog shows the IED’s IEC 61850 model and allows you to select which attribute to allocate to the signal.

The “Mapping Resolved” property shows which signals has been mapped successfully to the IED IEC 61850 model. Signals which have not bee mapped successfully are marked with a red cell.

## 10.6.6 HMI Data Consistency Check

The HMI Data Consistency Check provides a check for errors in the configured MicroSCADA loadfile. This function can be executed from the HMI navigation tab.

1. Select the specific HMI client node from the HMI navigation tree
2. Select the HMI Data editor and click on “Check Consistency” button




The consistency check provides a check on:

- Duplicated or missed LN:IX combinations
- Duplicated or missed LN:OA:OB combinations
- Wrong address range for OB values
- Duplicated ON values
- Aspects references which are not contained in a dataset
- Invalid AN attributes

Note that IET will still allow you to export a MicroSCADA loadfile with warnings. It is however strongly recommended to resolve these errors in the project before exporting the loadfile. Duplicated or wrong values may result in an error during the import of the loadfile into MicroSCADA.

# 11 Bay-level Gateway Engineering

## 11.1 Overview

IET600 supports the integration of all IEC 61850 bay level gateways using the CID file of the gateway (Bottom-up). The CID file contains the IEC 61850 model of the gateway itself as well as a IEC 61850 representation of the connected devices. This file must be generated by the gateway configuration tool and will be imported into IET600 for system engineering.

## 11.2 SPA-ZC40x

The CET for SPA-ZC40x is the configuration tool for the SPA/IEC 61850 gateway. Use CET to configure and download the configuration of the SPA-ZC40x gateway and create a CID file. Please refer to the CET for SPA-ZC40x manual for further information.

# 12 NCC / Gateway Engineering

## 12.1 Overview

### 12.1.1 Prerequisites

NCC engineering is a general term for the functionality needed to exchange data between the station computer or gateway to clients outside of the substation, such as the Network Control Center (NCC). To configure this data exchange, the data model of the IEDs in the system and the communication (between the IED and station computer/gateway) must be defined.

### 12.1.2 Engineering Approach

Currently, IET supports the following gateway devices:

- COM500 (integrated gateway of SYS600 MicroSCADA Pro and SYS600C)
- COM581
- RTU560

The engineering of the above gateways are quite different, requiring specific device-dependent configuration files.

Note that all gateways (except COM500) are treated as IEDs and requires an import of the ICD file to generate the IEC 61850 structure.

The following NCC protocols are currently supported:

- IEC101
- IEC104
- DNP 3.0

The following steps are required for engineering of the gateway for one NCC connection:

1. define Gateway devices and data connection to NCCs (e.g. single connections, hot standby configuration)
2. define data points or signals which are to be sent to the NCC, including conditions for transmission
3. define grouping of data signals
4. configure protocol-specific information (addressing information, data type mapping, etc.)
5. export signal list (if needed for customer discussions or verification)
6. export device-specific configuration files

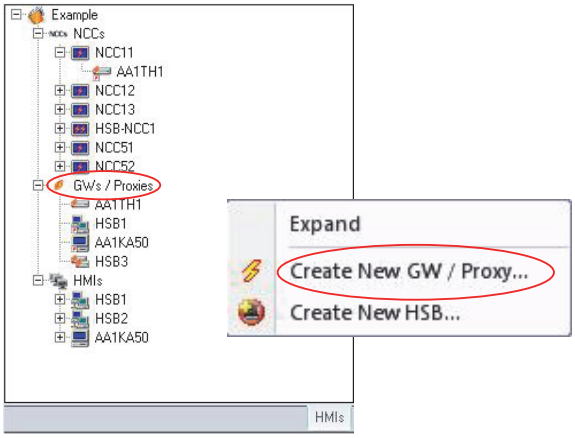
One important goal of NCC engineering is to provide the customer with a list of data points which will be delivered to the NCC. This list is normally required at an early stage in the project and should be consistent with the final engineered system.

For more than one NCC, IET provides support to copy or duplicate the configuration data from one NCC to reuse for other NCC connections (e.g. 2 redundant gateways with identical configuration).

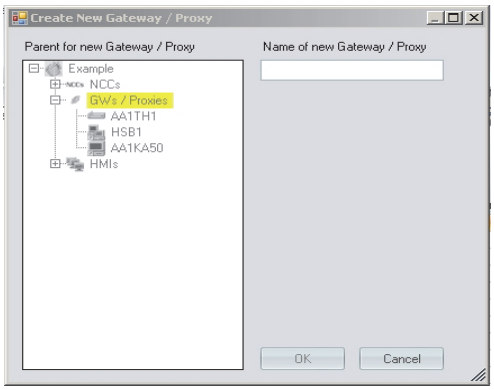
## 12.2 Create Gateways / Proxy

### 12.2.1 Create Stand-alone Gateway / Proxy

1. Open the HMI tab and select the GWs / Proxies node, right click to open the context menu → select Create new GW / Proxy…



2. Specify the name of the new GW / Proxy → click OK

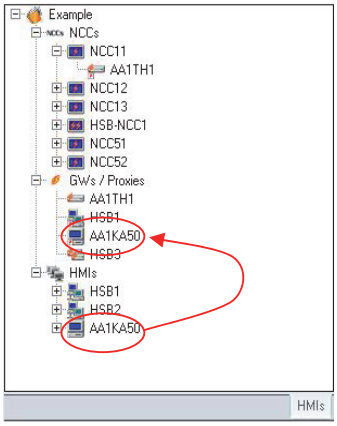



3. The new GW is now shown in the HMI navigation tab

## 12.2.2 Create GW / Proxy with integrated HMI

Integrated HMI/GW devices need to be defined under both the HMIs Node and the GWs / Proxies node of the HMI navigation tab.

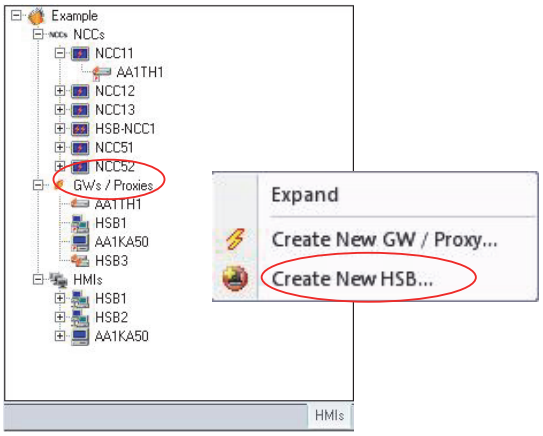
1. Open the HMI tab, select the HMIs node and click on an existing HMI or Hot Standby (HSB) HMI device
2. Drag and drop the HMI device (e.g. HSB1) into the GWs / Proxies node



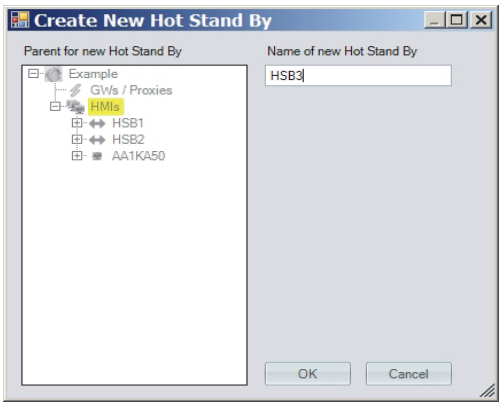
3. The HSB1 device is now defined as a HSB gateway

### 12.2.3 Create HSB Gateway / Proxy

1. Open the HMI tab and select the GWs / Proxies node, right click to open the context menu → select Create new HSB…




2. Specify the name of the new HSB → click OK




3. The new HSB GW is now shown under the GW node

Next, we need to define which two GW devices belong to the HSB configuration.

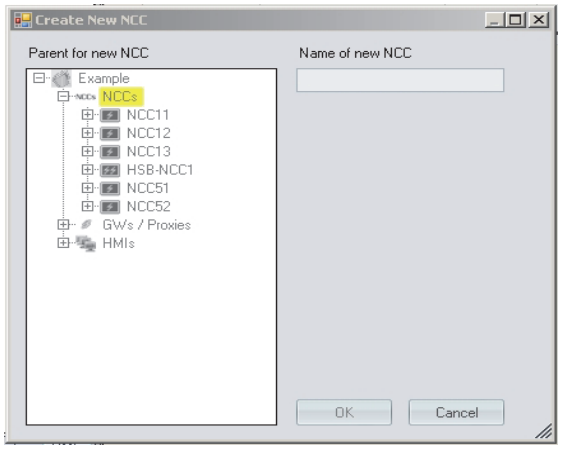
4. Create two standalone GW devices as described in the previous sections
5. Select the first GW device → drag and drop it into the HSB node

6. Repeat step 5 for the second GW

## 12.3 Create NCC

### 12.3.1 Create single NCC

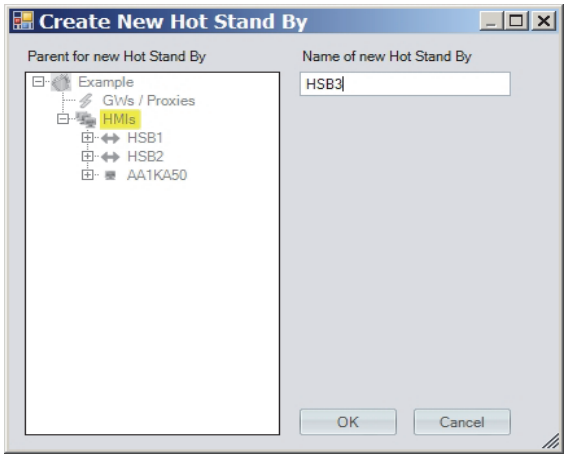
1. Open the HMI tab. Select the NCCs node, right click and select Create New NCC…
2. Enter the NCC name in the dialog and click OK




3. the NCC device is now created under the NCCs node

### 12.3.2 Create HSB NCC

1. Open the HMI tab. Select the NCCs node, right click and select Create New HSB…
2. Enter the HSB name in the dialog and click OK



3. the HSB device is now created under the NCCs node

Next, we need to define which two NCCs belong to the HSB configuration.

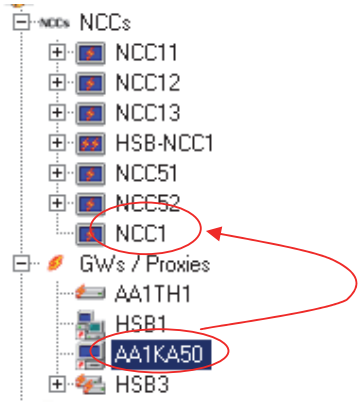
4. Create two stand-alone NCC devices as described in the previous section
5. Select the first NCC → drag and drop it into the HSB node


6. Repeat step 5 for the second NCC

## 12.4 Connect Gateways / Proxy to NCCs

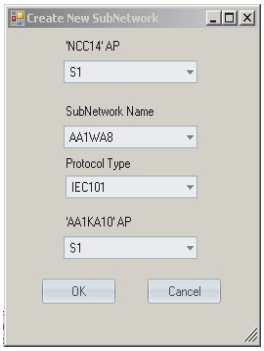
### 12.4.1 Connect single Gateway / Proxy to single NCC

1. Select a GW / Proxy node → drag and drop it into the NCC node




For this simple case, IET600 provides support to specify the physical connection properties of the link between the GW/Proxy access point and the NCC access point. This can also be defined in the communication pane for each device.

2. The Create New SubNetwork dialog is opened and specify the physiccal connection properties → click OK




Note that the Subnetwork Name and Access Points (AP) is used to assign the GW to a Subnetwork in the Communication navigation tab.

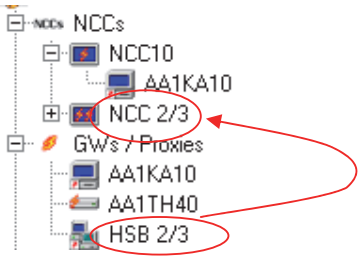
3. The GW / Proxies node now appears under the NCC node


Repeat step 1 to 3 to map another single GW / Proxy to the NCC.

## 12.4.2 Connect multiple Gateways / Proxys to multiple NCCs

There are many more combinations to connect multiple GWs/Proxys to one or multiple NCCs than described in the previous section. Therefore in the HMI pane, connecting multiple GWs/Proxys to the NCC node defines only the logical connections to the data available to each NCC. The physical connection of the individual GW/Proxy access points and each NCC access point must be done separately in the Communication pane, as discussed in the chapter Communication Engineering.

4. Select a single or HSB GW / Proxy node → drag and drop it into a single or HSB NCC node




5. The GW/Proxy node now appears under the NCC node


## 12.5 Select Signals for NCC

### 12.5.1 Overview

Once the GW/Proxy and NCC devices nodes have been created, one is ready to configure the specific signals to the remote client. The Signal Clients editor provides the functionality to map data signals from the station to the Gateway / Proxy and finally to the destination NCC.

### 12.5.2 Signal Clients Editor

The Signal Client Editor is available when the HMI Navigation pane is open. The editor provides a filtered view of the station signals which are available to the GW / Proxy and connected NCCs.

Note that the Signal Clients Editor provides the “server” view of the GW / Proxy. It shows:

- All of the station signals which are available in the station
- Signal Selection columns showing the GW / Proxy and its associated NCC clients

The editor shows on the left hand side the station signals available and its properties. On the right hand side, the editor shows a matrix for the Signal Selection or “marshalling” to the available GWs / Proxies and its allocated NCCs. The following sub-headings are available in the Signal Selection section:

- Sel (Select) Place an “x” to assigned a signal to a GW / Proxy and NCC.
- Grp (Group) Place a number to group signals. Signals with the same group number in the same NCC belong to the same group.
- GrpI (Group Index) An automatically generated incremental number for each member of the group. A positive number indicates a downlink (indication) and a negative number indicates an uplink (command) group.

Station signals and its properties

Signal selection or “marshalling”

| | | | | | AA1TH1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | NCC11 | | | NCC12 | | | NCC13 | | |
| StationText | EventTextLanguage1 | StatusTextL | LN | IX | Sel | Grp | GrpI | Sel | Grp | GrpI | Sel | Grp | GrpI |
| | Command from NCC NCC51 | | BNCC 00001 | 10 | | | | | | | | | |
| | Command from NCC NCC51 | | BNCC 00002 | 10 | | | | | | | | | |
| | Command from NCC NCC51 | | BNCC 00003 | 10 | | | | | | | | | |
| | Command from NCC NCC51 | | BNCC 00004 | 10 | | | | | | | | | |
| | Command from NCC NCC51 | | BNCC 00005 | 10 | | | | | | | | | |
| | Command from NCC NCC51 | | BNCC 00006 | 10 | | | | | | | | | |
| | Command from NCC NCC51 | | BNCC 00007 | 10 | | | | | | | | | |
| | Command from NCC NCC52 | | BNCC 00008 | 10 | | | | | | | | | |
| | Command from NCC NCC52 | | BNCC 00009 | 10 | | | | | | | | | |
| | Command from NCC NCC52 | | BNCC 00010 | 10 | | | | | | | | | |
| | Command from NCC NCC52 | | BNCC 00011 | 10 | | | | | | | | | |
| Sils | Group Alarm 1 for NCC NCC51 | | BNCC0 GRP | 1 | | | | | | | | | |
| Sils | Group Alarm 2 for NCC NCC51 | | BNCC0 GRP | 2 | | | | | | | | | |
| Sils.Aarau | Binary input (spare) 01 | Off/On | C1Q01 | 265 | x | 1 | 1 | | | | | | |
| Sils.Aarau | Binary input (spare) 02 | Off/On | C1Q01 | 266 | x | 1 | 2 | | | | | | |
| Sils.Aarau | Binary input (spare) 03 | Off/On | C1Q01 | 267 | x | 1 | 3 | | | | | | |
| Sils.Aarau | Binary input (spare) 04 | Off/On | C1Q01 | 268 | x | 2 | 1 | | | | | | |
| Sils.Aarau | Binary input (spare) 05 | Off/On | C1Q01 | 269 | x | 2 | 2 | | | | | | |
| Sils.Aarau | Binary input (spare) 06 | Off/On | C1Q01 | 270 | x | 2 | 3 | x | 1 | 1 | | | |
| Sils.Aarau | Binary input (spare) 07 | Off/On | C1Q01 | 271 | | | | x | 1 | 2 | | | |
| Sils.Aarau | Binary input (spare) 08 | Off/On | C1Q01 | 272 | | | | x | 1 | 3 | | | |
| Sils.Aarau | Binary input (spare) 09 | Off/On | C1Q01 | 273 | | | | x | 1 | 4 | | | |
| Sils.Aarau | Binary input (spare) 10 | Off/On | C1Q01 | 274 | | | | | | | | | |
| Sils.Aarau | Binary input (spare) 11 | Off/On | C1Q01 | 275 | | | | | | | | | |
| Sils.Aarau | Binary input (spare) 12 | Off/On | C1Q01 | 276 | | | | | | | | | |

Signal Clients

Note that the Signal Clients editor is empty when an NCC object node is selected in the HMI navigation pane. This is because the Signal Clients editor provides a “server” view and the NCC acts only as a “client” to the data. To view the data signals available at the NCC node, open the NCC Data editor or click on the GW / Proxy node.

## 12.5.3 Select Signals to be Sent to NCC

1. Select the GW node in the HMI Navigation tab → select the Signal Clients editor
2. Place an “x” in the matrix under the column heading “Sel” to assign a station signal to a GW / Proxy and NCC
3. These signals are now marshalled to the respective GW / Proxy and NCC

## 12.6 Group Signals

### 12.6.1 Overview

To group signals to the NCC, one must distinguish between:

- Indication signals – signals which are only read from the GW
- Command signals – signals which require both read and write from the GW to NCC

### 12.6.2 Group Indication Signals

Indication signals are grouped in the Signal Clients editor under the “Grp” column. Signals which are assigned to the same group are given the same number.

1. Enter a group name or number under “Grp” column (e.g. AI1) to identify which signals belong to the same group. The grouped signals are automatically assigned a “GrpI” number to indentify the group index of the signal within the group

| StationText | EventTextLanguage1 | StatusTextL | LN | IX | AA1KA50 | AA1KA50 NCC51 Sel | AA1KA50 NCC51 Grp | AA1KA50 NCC51 GrpI | AA1KA50 NCC52 Sel | AA1KA50 NCC52 Grp | AA1KA50 NCC52 GrpI |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Sils | Group Alarm 1 for NCC NCC51 | | BNCC0_GRP | 1 | | x | AI1 | -1 | | | |
| Sils | Group Alarm 2 for NCC NCC51 | | BNCC0_GRP | 2 | | x | AI2 | -1 | | | |
| Sils.Aarau | Binary input (spare) 01 | Off/On | C1Q01 | 265 | | | | | | | |
| Sils.Aarau | Binary input (spare) 02 | Off/On | C1Q01 | 266 | x | | AI1 | 1 | | | |
| Sils.Aarau | Binary input (spare) 03 | Off/On | C1Q01 | 267 | x | | AI1 | 2 | | | |
| Sils.Aarau | Binary input (spare) 04 | Off/On | C1Q01 | 268 | x | | AI1 | 3 | | | |
| Sils.Aarau | Binary input (spare) 05 | Off/On | C1Q01 | 269 | x | | AI1 | 4 | | | |
| Sils.Aarau | Binary input (spare) 06 | Off/On | C1Q01 | 270 | x | | AI1 | 5 | | | |
| Sils.Aarau | Binary input (spare) 07 | Off/On | C1Q01 | 271 | x | | AI1 | 6 | | | |
| Sils.Aarau | Binary input (spare) 08 | Off/On | C1Q01 | 272 | x | | AI1 | 7 | | | |
| Sils.Aarau | Binary input (spare) 09 | Off/On | C1Q01 | 273 | x | | AI2 | 1 | | | |
| Sils.Aarau | Binary input (spare) 10 | Off/On | C1Q01 | 274 | x | | AI2 | 2 | | | |
| Sils.Aarau | Binary input (spare) 11 | Off/On | C1Q01 | 275 | x | | AI2 | 3 | | | |
| Sils.Aarau | Binary input (spare) 12 | Off/On | C1Q01 | 276 | x | | AI2 | 4 | | | |
| Sils.Aarau | Binary input (spare) 13 | Off/On | C1Q01 | 277 | x | | AI2 | 5 | | | |
| Sils.Aarau | Binary input (spare) 14 | Off/On | C1Q01 | 278 | x | | AI2 | 6 | | | |
| Sils.Aarau | Binary input (spare) 15 | Off/On | C1Q01 | 279 | x | | AI2 | 7 | | | |
| Sils.Aarau | Binary input (spare) 16 | Off/On | C1Q01 | 280 | x | | AI2 | 8 | | | |

2. A Group Alarm signal (e.g. Group Alarm 1 for NCC NCC51) is automatically generated by the tool
3. Repeat steps 1-2 for the next group using another “Grp” number (e.g. AI2) to indicate a new group

Note that the “Sel” column does not need to be selected to group the signal. Selecting the “Sel” column indicates that the signal shall be sent to the NCC individually.

### 12.6.3 Group Command Signals

A special group of signals is needed to execute a remote command from the NCC, where all of these signals are required to execute a command successfully. These Group Command signals include:

- Position indication
- Command select open
- Command execute open
- Command select close
- Command execute close
- Command cancel

When one or more of the above signals is selected, IET600 will automatically group the entire set of signals as a Group Command.

1. Select one of the signal types listed above (e.g. Position indication) and enter a group name in the “Grp” column (e.g. C1)

| | | | | | | | AA1KA50 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | NCC51 | | | NCC52 | | |
| StationText | EventTextLanguage1 | StatusTextL | LN | IX | AA1KA50 | Sel | Grp | Grpl | Sel | Grp | Grpl |
| Sils.Aarau | Position indication | Intermediate/ | C1Q01QA1 | 10 | x | x | C1 | 6 | | C1 | 6 |
| Sils.Aarau | Command select open | (empty)/Select | C1Q01QA1 | 11 | x | | C1 | 1 | | C1 | 1 |
| Sils.Aarau | Command execute close | (empty)/Execu | C1Q01QA1 | 14 | x | | C1 | 4 | | C1 | 4 |
| Sils.Aarau | Command cancel | (empty)/Cance | C1Q01QA1 | 25 | x | | C1 | 5 | | C1 | 5 |
| Sils.Aarau | Command select close | (empty)/Select | C1Q01QA1 | 12 | x | | C1 | 2 | | C1 | 2 |
| Sils.Aarau | Command execute open | (empty)/Execu | C1Q01QA1 | 13 | x | | C1 | 3 | | C1 | 3 |
| | Command from NCC NCC51 | | BNCC_00001 | 10 | | x | C1 | -1 | | | |

2. IET600 will automatically find the remaining signals in the group and assign these also to the same group (e.g. C1).

## 12.7 Configure NCC Data

### 12.7.1 NCC Data Editor

The NCC Data Editor is available from the HMI Navigation pane. The editor provides a filtered view of the station signals which are available to the specified NCC only.

Note that the NCC Data Editor provides the “client” view to the GW / Proxy. It shows the signals which are mapped to the specific NCC only.

- The editor shows on the left hand side the station signals which have been selected or “marshalled” in the Signal Clients editor. On the right hand side, the editor shows a matrix for configuration of the NCC signals.

Station signals and its properties

NCC signal properties

| | StationText | EventTextLanguage1 | StatusTextLanguage1 | LN | IX | NCC11 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Address | Signal Handling | Command Type | Transmission Class | Scale Name | Send Trigger |
| ▶ | | | | | | | | | | | OnC |
| | | | | | | | | | | | OnC |
| | | | | | | | | | | | |
| | Sils.Aarau | Binary input (spare) 01 | Off/On | C1Q01 | 265 | | | | | | OnC |
| | Sils.Aarau | Binary input (spare) 02 | Off/On | C1Q01 | 266 | | | | | | OnC |
| | Sils.Aarau | Binary input (spare) 03 | Off/On | C1Q01 | 267 | | | | | | OnC |
| | Sils.Aarau | Binary input (spare) 04 | Off/On | C1Q01 | 268 | | | | | | OnC |
| | Sils.Aarau | Binary input (spare) 05 | Off/On | C1Q01 | 269 | | | | | | OnC |
| | Sils.Aarau | Binary input (spare) 06 | Off/On | C1Q01 | 270 | | | | | | OnC |
| | Sils.Aarau | Binary input (spare) 07 | Off/On | C1Q01 | 271 | | | | | | |
| | Sils.Aarau | Binary input (spare) 08 | Off/On | C1Q01 | 272 | | | | | | |
| | Sils.Aarau | Binary input (spare) 09 | Off/On | C1Q01 | 273 | | | | | | |

NCC Data

## 12.7.2 Configure NCC signal properties

The following sub-headings are available on the NCC Signal Properties section:

- Address
- Signal Handling
- Command Type
- Transmission Class
- Scale Name
- Send Trigger (default: OnChange, currently not used)

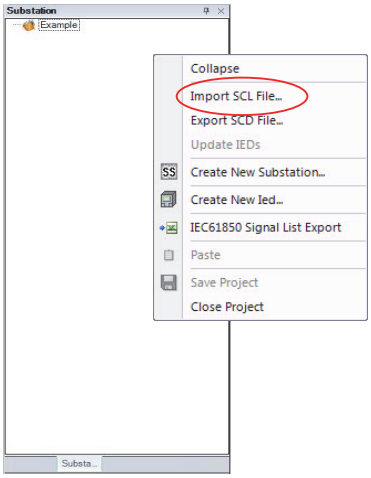
Note that only the white cells in the NCC signal properties columns are editable fields, meaning that one can manually specify the attribute in the field. The grey cells are automatically defined by the tool.

# 13 Data Import/Export

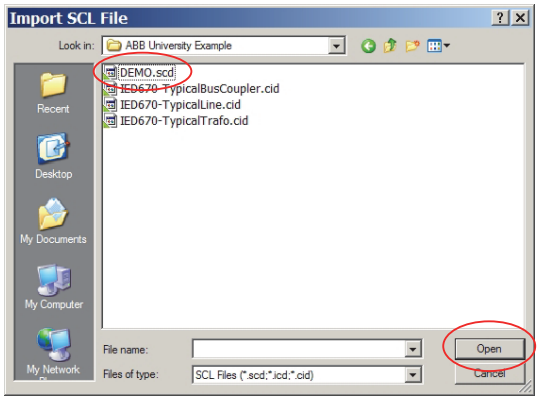
## 13.1 SCL files

### 13.1.1 Import an SCD File to Create a Project

1. Open a new project
2. Right-click on the empty Project




3. Click on “Import SCL File”
4. Choose “Import SCL File” -> the normal dialog for selecting a file appears.




Select an appropriate SCD file and click “Open”.

5. The contents of the file will be imported without further confirmation.
6. Check the contents of the project after the import
   - If you choose the wrong SCD file or you are not satisfied otherwise, close the project without saving. You will not be able to import another SCD file in this

project, therefore it would be no use to attempt to delete e.g. the Substation structure in the hope to import it again from another SCD file.

- If you choose an ICD/IID file (probably by accident; an ICD/IID file will not make much sense in this situation), you will probably find only one or several IEDs, but no Substation structure. Again, close the project without saving; do not try to attempt any other way to correct it.

### 13.1.2 Import an SCD File to Update IEDs

1. Right-click on a Project, Substation, Voltage Level, Bay or IED -> the context menu appears:
2. Choose “Update IED” (from an IED context menu) or “Update IEDs” (from all other context menus).
3. Proceed as described in chapter 7. 3: Update IEDs

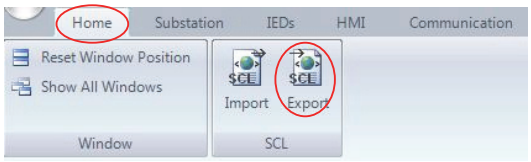
### 13.1.3 Import an ICD/IID File to Update IEDs

This is identical to the import of an SCD file as described above (chapter 13. 1. 2).

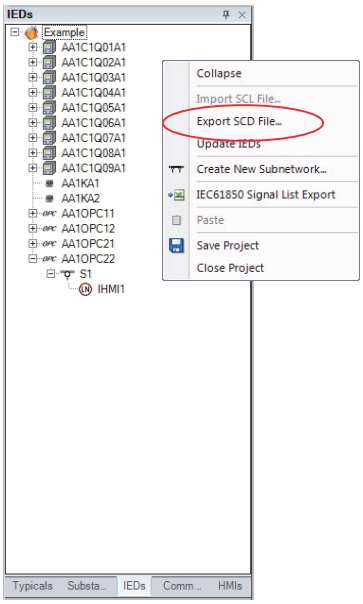
### 13.1.4 Export an SCD File

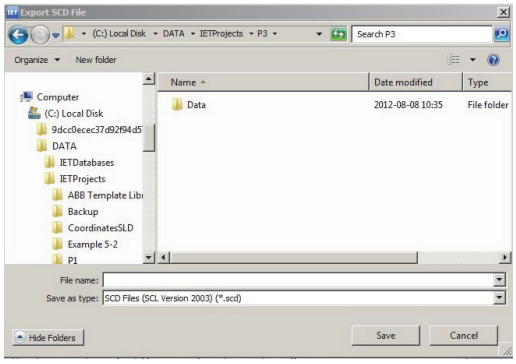
An SCD file can be exported either from the Home menu or from the navigation tab.

1. Select the “Home” menu tab and click on “Export”




Alternatively, right-click on the Project node in the Substation, IED or Communication tab and select “Export SCD File” from the context menu:



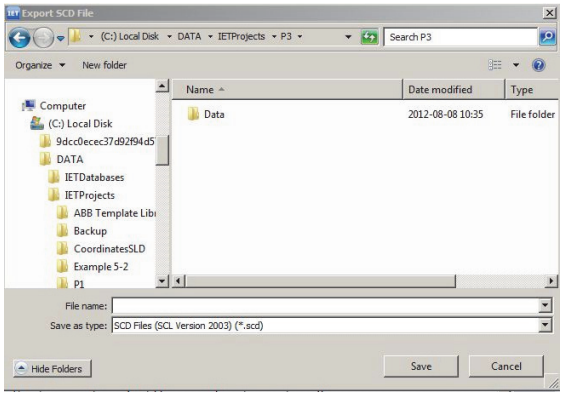




Note that the SCD file formats are available in two SCL versions:

- SCL version 2003
- SCL version 2007 (default)

By default the SCD file is exported as SCL in 2007 format because it contains more information. The 2003 format is optionally offered for backward compatibility reasons in case 3rd party tools cannot import the 2007 SCL format.

2. A normal “File save” dialog appears. Select the directory where you want to save the file and give it an appropriate name.




3. (If any SPAZC40x Gateways exits in the Project, their Loadfiles will automatically be exported on an SCD file export, they will be found in a subfolder “SPAZC40x” to the folder the SCD file is exported to.)

## 13.2 HMI Status Texts Exchange

### 13.2.1 Format of Status Text files

Status Texts can be exchanged with other tools via text files in CSV-format. Translations into different languages are kept in different files, so each file contains texts of only one single language.

The file name must have the format [filename].[language code].csv with the following specifications:

1. [filename] can currently be freely chosen; it is recommended to use “StatusTexts” to distinguish it from other files following a similar naming pattern (e.g. “SignalTexts”).
2. [language code] must be a two-letter language code according to ISO 639-1. The tool will recognize capital letters “EN” instead of “en” although it is not according to ISO 639-1.
3. “csv” as extension.

One line of the file must contain the following information:

1. A key to identify a set of Status Texts.
2. The type of the signal (AI for “analogue input”, AO for analogue output etc.)
3. A set of process values. They can be specified in two different ways:
   - a plain number represents a set of continuous values from 0 to the number minus one, e.g. a double binary value will have a “4”, representing the consecutive values 0, 1, 2 and 3
   - a set of numbers in vector format to represent an irregular sequence of numbers, e.g. “(0, 5, 6, 10, 100)”
4. a sequence of status texts, their count must match the count of the process values (e.g. if the process values are specified by a number “4”, there must be 4 status texts “intermediate”, “on”, “off”, “faulty”.
5. Items 1, 2, 3, and each text under 4 must be separated by a “;”. The line must **not** be terminated with a “;”
6. Texts must not start or end with delimiters such as “ or ‘ (such characters will be interpreted as belonging to the texts, not as delimiters)

The file must also contain a header with defined information. Any file not following the above specifications cannot be imported.

A default file with English status texts, as well as the command procedures scripts to import and export these texts into MicroSCADA, are provided with the IET600 baseline installation. These files can be found at:

*C:\Program Files\ABB\IET 5.2.x\SADesigner\ApplicationTemplates\HMI\StatusTexts*

### 13.2.2 Import Status Texts

Status Texts can only be imported from the Main Menu

1. Select the “HMI” tab in the Main Menu.

2. Click “Status Texts” in the “Import” group -> the “Import Status Texts” dialog appears (the default path is the subdirectory “Import” in your local project path).
3. Select one or more Status Text files (typically named “StatusText.[language code].csv.
   - The part [language code].csv is mandatory; a file with a different name cannot be imported.
   - You can select several files in different languages to import Status Texts in several languages at the same time.
4. Click “Open” -> the files are imported.
5. If the import succeeds, you will not get any feedback directly. If something goes wrong, an error message will appear.
6. If you have HMI signals without Status Texts but with valid Status Text keys (e.g. EH attribute for MicroSCADA signals), Status Texts will be added in the HMI Signal Editor immediately after import.

A Re-Import of a Status Texts file will add new Status Texts and will overwrite Status Texts that have changed, but it will not delete existing Status Texts that may not exist in the imported file.

### 13.2.3 Export Status Texts

Status Texts can only be imported from the Main Menu

1. Select the “HMI” tab in the Main Menu.
2. Click “Status Texts” in the “Export” group -> an “Export Status Texts” dialog appears (the default export path is the subdirectory “Export” in your local project path).
3. Choose a basic name (“StatusTexts” is recommended and may become mandatory at some point in the future).

   If you have status texts in different languages, they will be exported as “StatusTexts.en.csv”, “StatusTexts.de.csv”, “StatusTexts.fi.csv” etc. Files with those names in the export directory will be overwritten,
4. Click “Save” -> the files are exported.
5. If the export succeeds, you will not get any feedback directly. If something goes wrong, an error message will appear.

A Re-Export will overwrite files in the export directory.

## 13.3 HMI Signal Information Exchange

### 13.3.1 Overview

There is no standard describing how to exchange Signal Texts and other HMI-related information via SCD file. So the tendency is to exchange these data in additional files, with references to IEC-61850 objects.

Currently, HMI information can only be exchanged in a MicroSCADA-specific format (LOF file or CSV file).

## 13.3.2 Import MicroSCADA Database file (LOF or CSV file)

The MicroSCADA Database file import function utilizes a LOF file or CSV file exported from an existing MicroSCADA system or a previous IET project. On a MicroSCADA file import, the tool will use the OPC references (MicroSCADA ON attribute) to map the imported signals to IEC 61850 data. If this reference does not exist or is wrong, a file import may not be of very much help and indeed upset things.

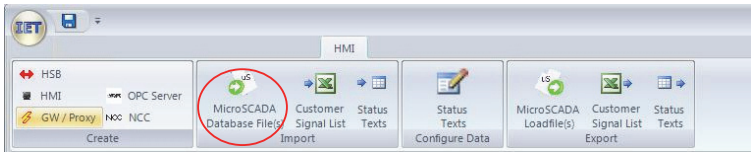
Alternatively, the tool uses the MicroSCADA OI attribute for the signal mapping.

A MicroSCADA file import is therefore recommended only in two specific situations:

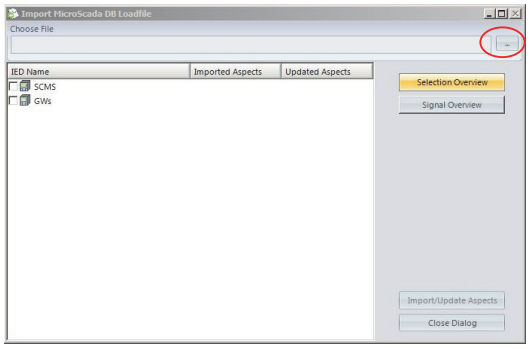
1. At the start of a project, if you do not have an IET600 project, but only a running MicroSCADA 9.x (or an up-to-date LOF/CSV file) with or without an SCD. Before the import, you should have configured your IEDs and done the LN mapping (this will be used by IET600 to automatically map signals on import).
2. To exchange data with MicroSCADA 8.x. In this case you are sure to have valid ON attributes for the mapping of the data.

LOF files can only be imported from the Main Menu.

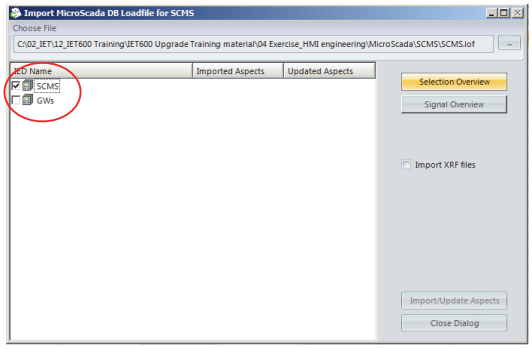
1. Select the “HMI” menu tab -> the Navigation Panel switches to the “HMIs” tab.
2. Click “MicroSCADA Database File(s)” in the “Import” group of the Main Menu.



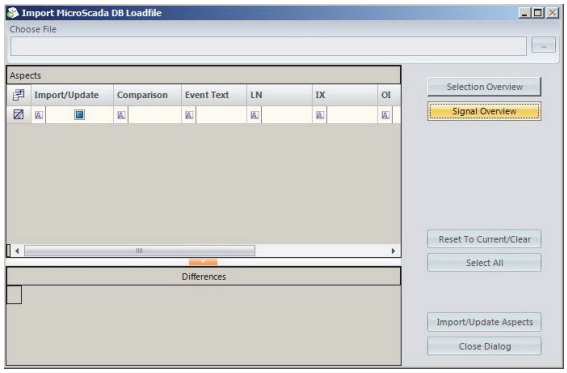

3. The “Import MicroSCADA Database File(s)” dialog appears. Click on the button to the right to choose a LOF files for import:



4. A “File Open” dialog appears. Select a suitable LOF or CSV file.
5. Once you have selected a file, you can select the destination HMIs from a list of available HMIs (i.e. the HMIs configured in your project). You can import the same signals to several HMIs in one import.




6. To preview the signals which will be imported, click the “Signal Overview” button

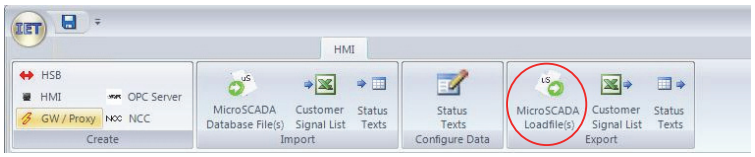



7. Click “Import/Update Aspects” to start the file import.
8. The dialog will inform you how many signals have been imported to each HMI.
9. Repeat 3, 4, 5 and 6 for more imports from other files, if needed.
10. Click “Close” to close the dialog.

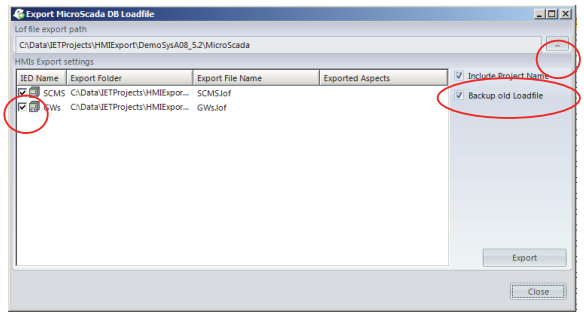
## 13.3.3 Export MicroSCADA Loadfiles (LOF file)

LOF files can only be exported from the Main Menu.

1. Select the “HMI” menu tab -> the Navigation Panel switches to the “HMIs” tab.
2. Click “MicroSCADA LOF file(s)” in the “Export” group of the Main Menu.




3. The “Export MicroSCADA Loadfile(s)” dialog appears:




Select a directory where to export the LOF files.

Select the HMIs whose HMI signals will be exported. For each standalone or each Hot Standby System, one LOF file will be created.

Check “Backup old Loadfile” if you want IET600 to create backups of older LOF files with the same name.

### 13.3.4 Export MicroSCADA CSV file with Process Object Data

This is not possible. To transfer data from IET600 to MicroSCADA, only LOF files are available.

## 13.4 NCC Gateway Signal Information Exchange

### 13.4.1 Overview

Configuration exchange with the NCC Gateway is only supported for MicroSCADA COM500 or SYS600C gateways. The following files are exchanged between IET600 and the COM500 configuration tool:

- COM_XRIND.XRF
- COM_XCMD1.TXT

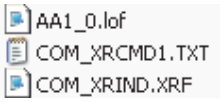
The above files are imported and exported together with the MicroSCADA Loadfile(s), as described in the previous section.

### 13.4.2 Import COM500 Configuration File (XRF)

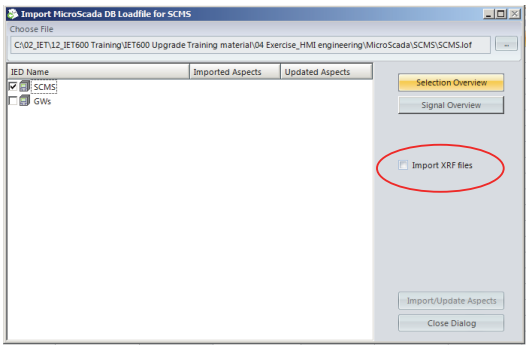
The COM500 configuration files are always imported with the corresponding MicroSCADA loadfiles.

An important task to be done before importing the COM500 files is to prepare the NCC nodes in the HMI navigation tab. The NCC nodes shall be defined in the order of the NCC line numbers in the COM500 file. Furthermore, the UN numbers of the NCCs must also be specified according to the file. IET will use these two references to correctly map the COM500 configuration to the correct NCC during the import.

Note that the COM500 files must be stored in the same folder as the MicroSCADA loadfiles, for example, in the folder Microscada:




To import the MicroSCADA loadfiles, follow the steps described in Section 13.3.2 “Import MicroSCADA Loadfile”. In the import dialog, select the check box “Import XRF files”:




### 13.4.3 Export COM500 Configuration File (XRF)

To export the COM500 signal configuration files, select Export MicroSCADA Loadfile(s) from the HMI menu, as described in the Section 13.3.3: Export MicroSCADA Loadfile. The COM500 files will be exported automatically to the same folder as the MicroSCADA loadfiles.

### 13.4.4 Export Configuration Data for other NCC Gateways

Currently, only the SCD file is available for other NCC Gateways. No specific signal configuration files are available for other gateways at this time.

## 13.5 Customer Signal Lists

### 13.5.1 Overview

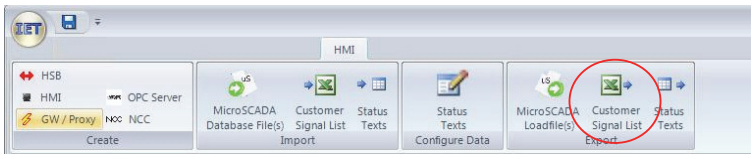
To facilitate exchange of data with the customer, and to offer the customer a possibility to edit these data on his own, a special Excel file, the “Customer Signal List” can be exported and re-imported.

Currently, this signal list contains much more information than the customer can edit. It is therefore only of limited use for its above purpose. Additions are planned.

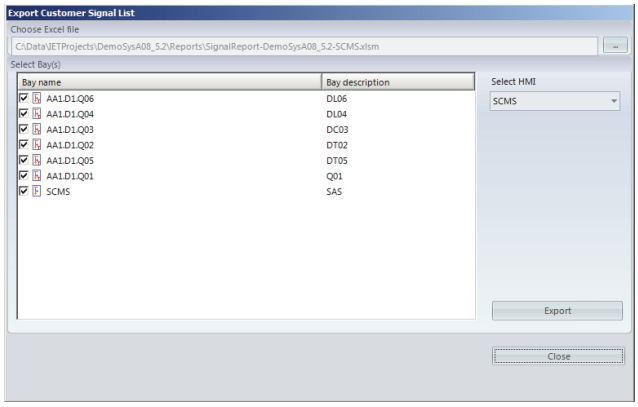
### 13.5.2 Export a Customer Signal List

A Customer Signal List can only be exported from the Main Menu.

1. Select the “HMI” menu tab -> the Navigation Panel switches to the “HMIs” tab.
2. Click “Customer Signal List” in the “Export” group of the Main Menu.




3. The “Export Customer Signal List” dialog appears:

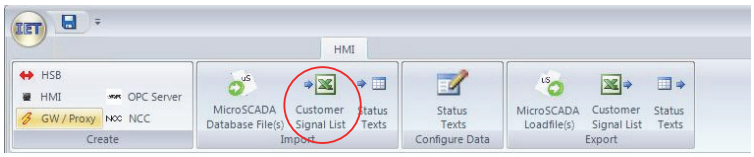



4. Select the HMI or Hot Standby System whose data you want to export (for each HMI, one Excel file will be generated).
5. Depending on the selected HMI, IET600 proposes a filename. You may change it according to your needs by clicking on the “…” button.
6. Select the Bays whose data will be exported. For each Bay, one Excel Sheet will be created in the file. As these sheets will be named after the customer name, it is mandatory that you have given Customer names to the Bays you want to export, otherwise they will be marked red and an attempted export will fail with an error dialog.
7. Click on “Export” to finally export the data of the selected HMI.
8. Repeat steps 4, 5 and 6 for additional HMIs, if needed.
9. Click “Close” to finally close the dialog.

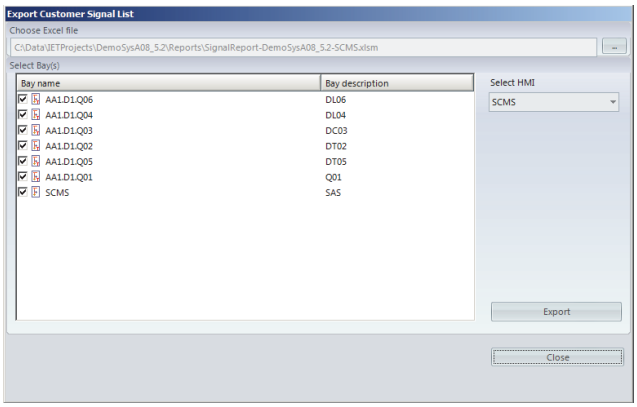
### 13.5.3 Import a Customer Signal List

A Customer Signal List can only be imported from the Main Menu.

1. Select the “HMI” menu tab -> the Navigation Panel switches to the “HMIs” tab.
2. Click “Customer Signal List” in the “Import” group of the Main Menu.




3. The “Import Customer Signal List” dialog appears. Click the “…” button to choose a Customer Signal List for import.




4. The contents of the file determine to which HMI the data will be imported. Select the Bays for which you want to import the data. Use the button “Select All” to select all of them.
5. Select “Import” to finally import the data.
6. Repeat steps 3, 4 and 5 to import data from other files.
7. Click “Close” to finally close the dialog.

# 14 Annex A: Regular Expressions

## 14.1 Introduction

A Regular Expression, in short RegEx, is a formalized way to describe search patterns.

The main advantage of RegExes is that it allows formulating complicate search patterns in a compact way.

The main disadvantage of RegExes is that they are not very intuitive. It takes some experience to be sure that they mean what they are intended to mean.

In our context, RegExes are used to find objects in IET600 which match the strings defined by the RegEx. E.g. if the RegEx for Data Object Instances (DOIs) is set to “^Pos$”, the search will find all DOIs with exactly “Pos”

## 14.2 Characters in Context of a RegEx

### 14.2.1 Definition

Characters in the context of a RegEx include everything defined by the Unicode character set.

### 14.2.2 Characters in Unicode

IET600 is using Unicode for Encoding. Also SCL-files are encoded in Unicode, IEC 61850 allows the use of Unicode Characters e.g. in the names of Substation tree elements.

Regular Expressions allow for powerful abbreviations such as /d for all digits. Be aware, that in Unicode, the meaning of such abbreviations may be different from what you intend. While in ASCII /d equals [0-9], in Unicode there are also Arabic and Indic characters that represent digits and are included in /d, but not in [0-9].

In the IET600 context, RegExes are currently only used for objects that contain a limited set of ASCII characters (A-Z, a-z, 0-9, _), so no special caution is needed.

### 14.2.3 Regular Expression Metacharacters

Metacharacters in RegExes are characters that do not have their literal meaning, but have a special and defined meaning in Regular Expressions.

| | |
|---|---|
| . | A dot matches for any single character (except if within brackets and except a newline character). |
| ? | A question mark matches zero or one occurrences of the preceding character. E.g. “values?” would match “value” and “values”. |

| | |
|---|---|
| * | A star matches zero or more occurrences of the preceding character. |
| + | A plus sign matches one or more occurrences of the preceding character. |
| ^ | A caret matches the beginning of a line (except if within brackets). |
| $ | A dollar sign matches the end of a line. |
| [ ] | A bracket expression matches one single character equal to any characters contained within the brackets. E.g. ^[A-Z].* matches any line starting with a capital Latin letter. |
| [^ ] | A caret within a bracket expression matches one single character equal to any characters not contained within the brackets. E.g. ^[^0-9].* matches any string not starting with a Latin digit. |
| \ | “Escape” character, forces the RegEx to take the next character literally. E.g. “^Pos$” matches exactly Pos, but “^\$” matches a string starting with a dollar sign. |

Alternatives:

| | |
|---|---|
| (\|) | This expression matches alternatives. E.g. ^(SPS\|DPS\|INS)$ matches any string that is either SPS or DPS or INS. |

Some more special patterns which may be useful in circumstances:

| | |
|---|---|
| a(?=b) | positive look-ahead assertion, match a only, if it is followed by b. |
| a(?!b) | negative look-ahead assertion, match a only, if it is not followed by b. |
| (?<=b)a | negative look-behind assertion, match a only, if it is preceded by b. |
| (?<!b)a | negative look-behind assertion, match a only, if it is not preceded by b. |

### 14.2.4 Substation Automation Examples

Some examples of match criteria:

| | |
|---|---|
| ^Pos$ | matches exact string “Pos” |
| ^(stVal\|q)$ | matches exact strings “stVal” or “q” |
| ^M | matches all strings starting with “M” |
| ^(?!PTRC)P | matches all strings starting with “P” except those starting with “PTRC” |

### 14.2.5 References

http://en.wikipedia.org/wiki/Regex offers a good overview of Regular Expressions.

# Contact us:

**ABB AB**
**Substation Automation Products**
SE-721 59 Västerås
Sweden
Tel. +46 (0) 21 32 50 00
Fax +46 (0) 21 14 69 18

http://www.abb.com/substationautomation