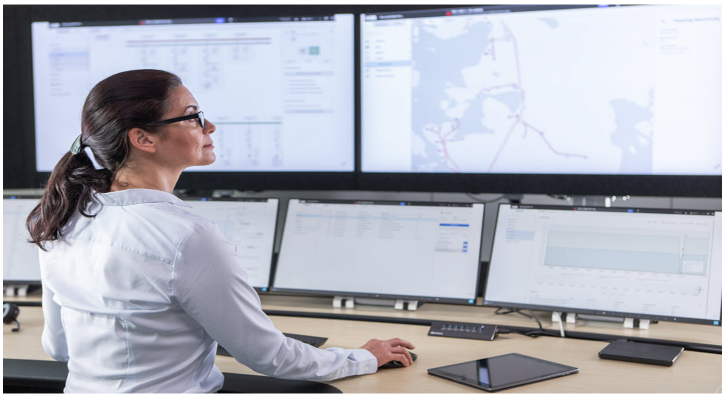
GRID AUTOMATION PRODUCTS

MicroSCADA X SYS600 10.2

Application Design



Document ID: 1MRK 511 466-UEN
Issued: March 2021
Revision: A
Product version: 10.2

Table of contents

Section 1 Copyrights

Trademarks

ABB is a registered trademark of ABB Asea Brown Boveri Ltd. Manufactured by/for a Hitachi Power Grids company. All other brand or product names mentioned in this document may be trademarks or registered trademarks of their respective holders.

Guarantee

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

Third Party Copyright Notices

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

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

Section 2 Introduction

2.1 This manual

This manual provides information for application engineers who build up the SYS600 applications using the Monitor Pro user interface.

The steps for application engineering are:

- Creating the process database by using the IET Data Loader Tool, SCL Importer or Object Navigator
- Building the Process Displays by dragging and dropping complete objects from Object Browser to the Process Display
- Customizing the Process Display
- Customizing the Alarm, Event, Blocking, Trend and Measurement Report Displays
- Customizing the menus and toolbars
- Localizing the application texts

.

2.2 Use of symbols

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

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

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

Information icon alerts the reader to relevant factors and conditions.

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

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

2.3 Related documents

Name of the manual	Document ID
SYS600 10.2 Operation Manual	1MRK 511 499-UEN
SYS600 10.2 Operation Manual for Workplace X	1MRK 511 500-UEN
SYS600 10.2 Process Display Design 1	1MRK 511 478-UEN
SYS600 10.2 Workplace X Process Picture Design	1MRK 511 505-UEN
SYS600 10.2 Installation and Administration Manual	1MRK 511 496-UEN
SYS600 10.2 System Objects	1MRK 511 482-UEN
SYS600 10.2 Application Objects	1MRK 511 467-UEN
SYS600 10.2 Visual SCIL Application Design	1MRK 511 483-UEN
SYS600 10.2 Workplace X View Writer's Guide 1	1MRK 511 506-UEN
SYS600 10.2 Programming Language SCIL	1MRK 511 479-UEN
SYS600 10.2 System Configuration	1MRK 511 481-UEN
PCM600 Installation and Commissioning Manual	1MRS755552
PCM600 Help	

2.4 Document conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a dialog, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for file names.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the CTRL key. Although the Enter and Shift keys are not labeled they are written in capital letters, e.g. press ENTER.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the space bar, comma key and so on.
- Press CTRL+C indicates that the user must hold down the CTRL key while pressing the C key (in this case, to copy a selected object).
- Press ALT E C indicates that the user presses and releases each key in sequence (in this case, to copy a selected object).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
    - The following convention is used for menu operations: **Menu Name/Menu Item/Cascaded Menu Item**. For example: select **File/Open/New Project**.
    - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.
- System prompts/messages and user responses/input are shown in the Courier font. For example, if the user enters a value that is out of range, the following message is displayed: `Entered value is not valid`.
  The user may be told to enter the string MIF349 in a field. The string is shown as follows in the procedure: `MIF349`
- Variables are shown using lowercase letters: sequence name

2.5 Document revisions

Revision	Version number	Date	History
A	10.2	31.03.2021	New document for SYS600 10.2

Section 3 Creating a SYS600 application

3.1 Preparing application

The application is prepared for software modules with Application Initialization Tool. If the first login to an application is done with Classic Monitor or Monitor Pro (+), the application is automatically prepared for SA_LIB. When the first login happens with SYS600 Tool Launcher, SA_LIB has to be prepared as well.

The tool is opened from the Monitor Pro menu **Tools/Engineering Tools/Tool Manager** and **Application Objects** sheet or **SYS600 Tool Launcher/Application Configuration** sheet.

If you are using SYS600 Workplace X for process controlling, the application **must** be prepared for Base Backbone and Base Process Library.

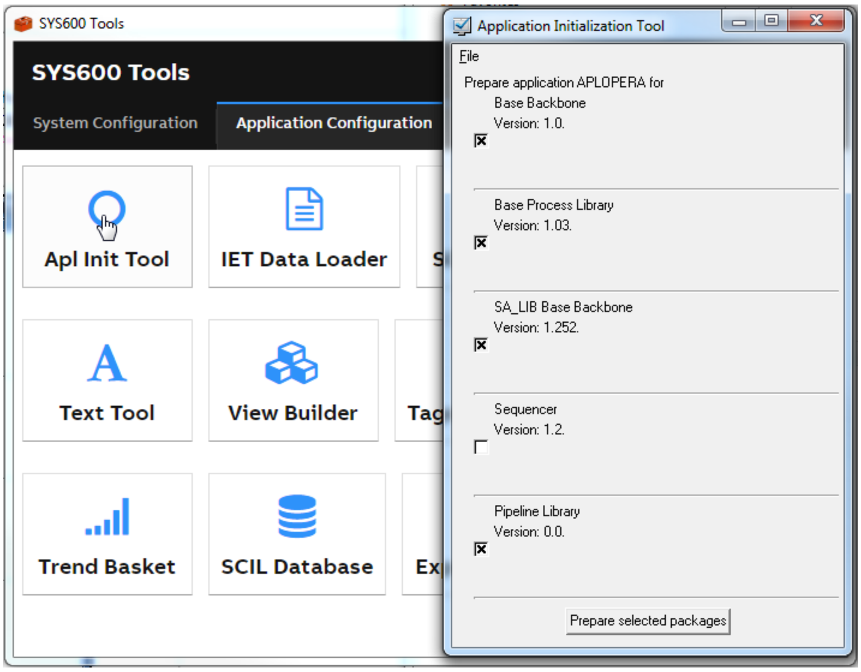


Figure 1: Application Initialization Tool

There are dependencies between different modules. Base Process Library requires the Base Backbone module. Select the modules and click **Prepare selected packages**.

3.2 Engineering scenarios

The engineering of a SYS600 application can be divided in different approaches:

- IET Data Loader
- Object Navigator engineering
- SCL Importer

From the different engineering approaches, the IET Data Loader is preferred.

3.2.1 IET Data Loader

The IET Data Loader tool is used when Engineering SYS600 with configuration data coming from IET600. This approach is primarily used for IEC 61850 engineering. For more information, see MicroSCADA X SYS600 10.0 IEC 61850 System Design.

3.2.2 Object Navigator engineering

These objects are installed during Process Display engineering by drag and drop operation. In this operation, all the connections between display elements and the process database are automatically configured. The following figure describes the Object Navigator engineering approach.

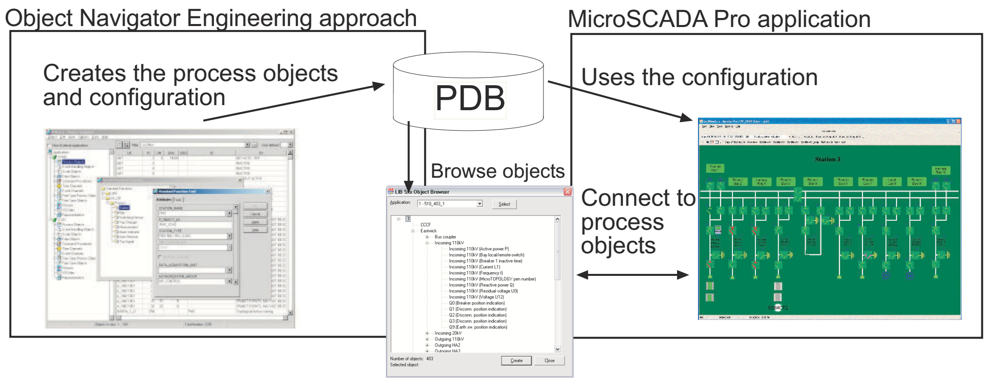


Figure 2: Object Navigator engineering approach

3.2.3 Object Navigator engineering

The function of an application is to fulfil the requirements to monitor and control the process devices related to certain application area. The application area may be electricity distribution, heat distribution or some other related application.

The application requires its own connections to the process equipment in the form of system and application objects. System objects describe how the system is configured. Application objects are more related to signal flow, when indications are received in the application database or process devices, which are controlled in the application. For more information, see SYS600 Application Objects and SYS600 System Objects.

3.2.3.1 Installing standard functions

Object Navigator is capable of creating application objects or standard functions. In the case of standard functions, a set of application objects is created during the standard function installation. Object Navigator can be used for application engineering of objects with regard to the following operations:

- Add
- Delete
- Modify
- Export
- Import
- Document

The functionality of the Object Navigator to create and engineer certain types of application objects is described in more detail in the Application Objects manual. The following section describes how Object Navigator can be used for creating and engineering standard functions.

Standard functions of LIB 5xx and Power Process Library

The Power Process Library is included in SYS600. This library contains generic functions for building process displays for distribution/transmission network. These functions can be used during application engineering to model application related processes. Each standard function contains a set of configurable attributes, which specify how the signal flow is passed between the process devices and the application database.

The following standard functions are included in the Power Process Library:

- Station
- Bay
- Switching Device
- Tap Changer
- Measurement
- Alarm Indicator
- Auto Reclose
- Trip Signal
- Generator
- Line Indicator

If the SYS600 system also contains Standard Application Library LIB 5xx products, they are included in the list of standard functions as well.

For more information on symbols, see Section 6.

Installing and configuring standard functions

In the Object Navigator, installing of standard functions is possible when **Standard Function/ Install** is selected from the menu bar. The **Install Standard Function** dialog is then opened, see Figure 3.

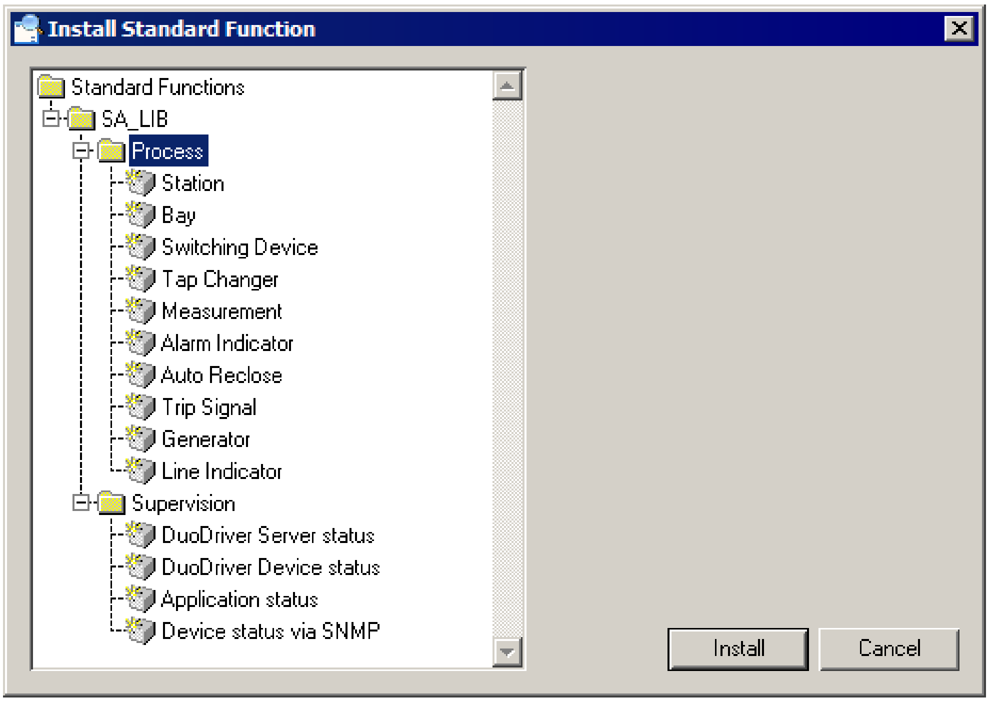


Figure 3: Install Standard Function

Expand the SA_LIB Process node, select a standard function, then click **Install** or press ENTER. The Standard Function Tool is now opened, see Figure 4.

Standard Function Tool contains the **Attributes**, **Programs**, and **Tools** tabs, see Figure 4.

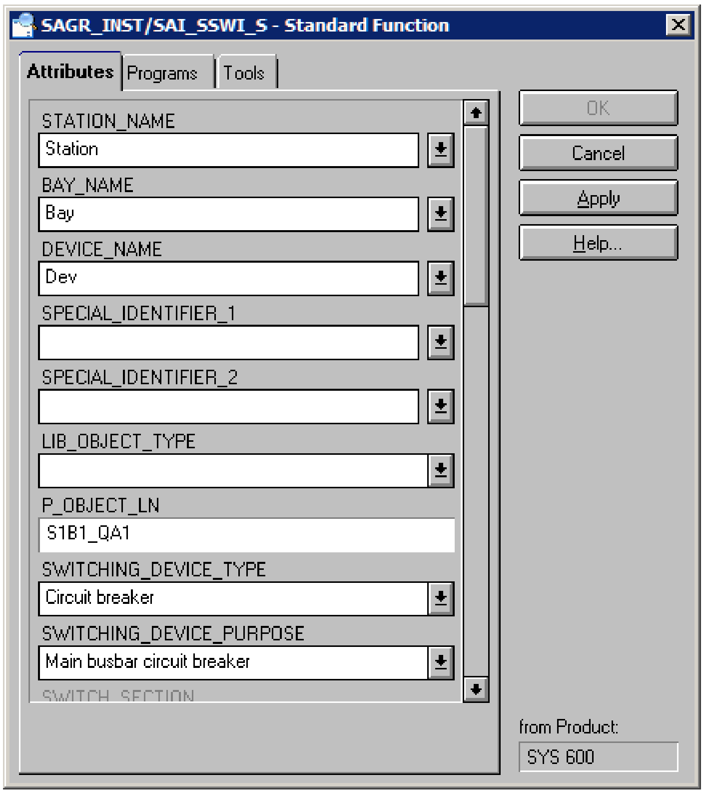


Figure 4: Standard Function Tool

In Standard Function Tool, the standard function related configurable attributes are displayed and configured in the **Attributes** tab. In the **Programs** tab, select the program that is to be edited. The standard function specific tool is found on the Tools tab.

When Standard Function Tool is opened, the default values are assigned in the attribute list. During application engineering, these attributes are engineered to the values required by the application in question. When **Apply** is clicked, the configuration for these attributes is created. The next step is to start the Process Object Tool from the Tools tab. This can be done either by double-clicking the Process Object Tool in the list or by selecting the Process Object Tool and clicking **Open**.

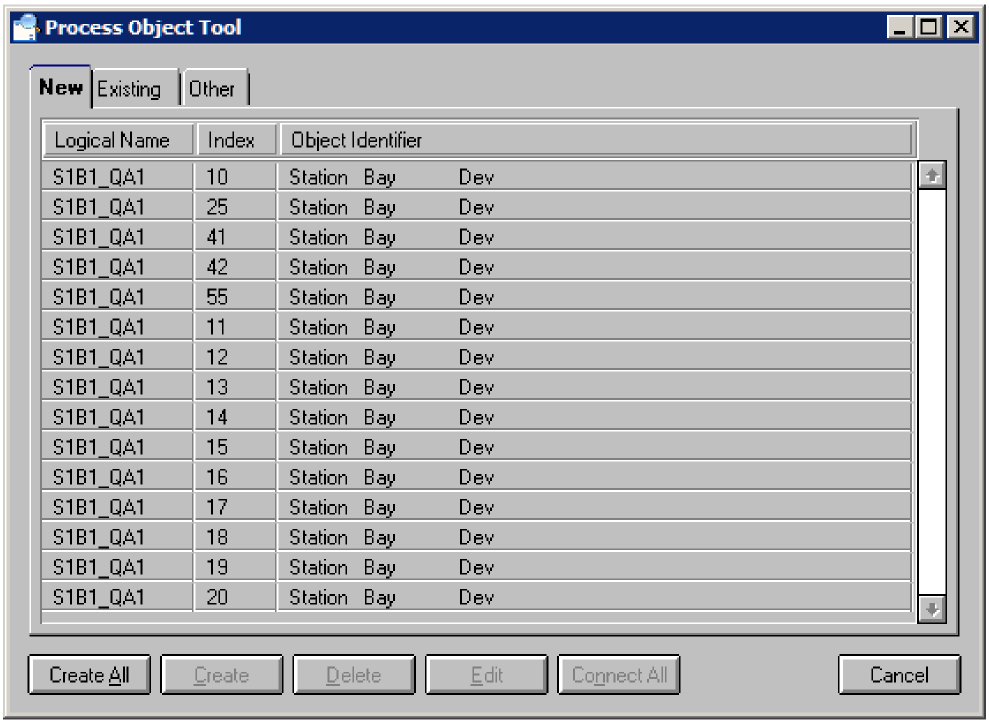


Figure 5: Process Object Tool

There are three tabs in the Process Object Tool:

- **New**
- **Existing**
- **Other**

On the **New** tab, the standard function related process objects can be created one by one or in small groups with the **Create** button located on the bottom row of the tool. The **Create All** button creates all the process objects on the **New** tab. After the process objects have been created, they appear on the **Existing** tab.

The **Existing** tab of the Process Object Tool includes functionalities to directly edit the process object attribute values usually modified in the Process Object Form of the Object Navigator. The editable process object attributes are:

IU = In Use

SS = Switch State

UN = Unit Number

OA = Object Address

OB = Object Bit Address

The OA and OB attributes are displayed and handled as process object type dependent encoded values, like in the Process Object Form of Object Navigator, thus they are not necessarily the same values as stored in the SYS600 process database. The titles for OA and OB fields are surrounded with brackets to indicate this encoding and slight address naming inaccuracy. Both the OA and OB attributes are displayed as decimal numbers, ignoring the Output Type attribute definition.

A separate edit area (bottom of the **Existing** tab) is used for editing purposes in this tab. Read only columns for the attributes are also shown in the process object list of the **Existing** tab.

When an existing process object is selected in the Process Object Tool, the current attribute values for IU, SS, UN, OA and OB are displayed in the edit area together with their corresponding dialog commands. When multiple process objects are selected, the attribute values of the first selected object are shown in the edit area. If an attribute value is modified, it affects the attribute value for all the selected process objects.

However, if multiple process objects are selected, then the OA and OB attribute fields are disabled. The UN attribute is the only address attribute that can be changed alone. If either OA or OB attribute is changed, all the three/two address attributes must be defined before applying the change. If all address attributes UN, OA and OB have their initial values (UN=0, OA=0 and OB=16), they are shown as empty fields.

The change in IU or SS attribute value is executed immediately after modifying the value of the edit control, except for multiple selected objects. After this, a confirmation prompt appears. The UN, OA and OB attribute changes are executed only after the user clicks the **Enter** button on the right hand side of the attribute edit areas, or presses the ENTER key on the keyboard.

The **Other** tab shows indexes that are not recognized by a standard function. This happens when the user creates new indexes for an existing process object group in the Object Navigator. For more information on the Object Navigator, see SYS600 Application Objects.

3.2.3.2 Connecting standard functions to display

Object Browser can be used for creating display symbols based on existing process objects in the process database. The browser contains the insertable objects structured in tree view. Object Browser is opened by selecting **Object Browser** in the **Actions** menu of the Display Builder.

Object Browser displays the object structure for all applications. An application can be selected from the drop-down list in the upper part of the dialog. Select an application from the list and click the **Select** button to update the view.

Symbols can be inserted in the display by dragging them to the drawing area. Station and bay symbols are dragged from the corresponding node. As an example, dragging an Incoming 110kV object in Figure 6 creates a bay symbol in the display. Dragging an Eastwick object creates a station symbol.

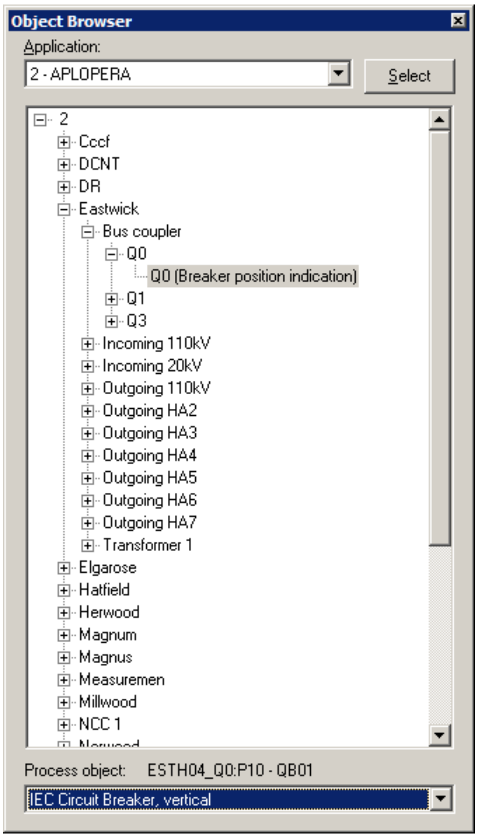


Figure 6: View of the Object Browser

3.2.4 SCL Importer

With SCL import function, process objects from CID (Configured IED Description) or SCD (System Configuration Description) files can be created. In addition, it is possible to create External OPC DA Client configuration file(s). SCL Importer can be used with IEC 61850 Protocol.

The substation section (template) is needed also in CID files.

The workflow when using SCL Importer as well as the different options that can be used are described in the following sections.

3.2.4.1 Basic workflow

The SCL Importer tool is opened from the Tool Manager's Application Objects tab.

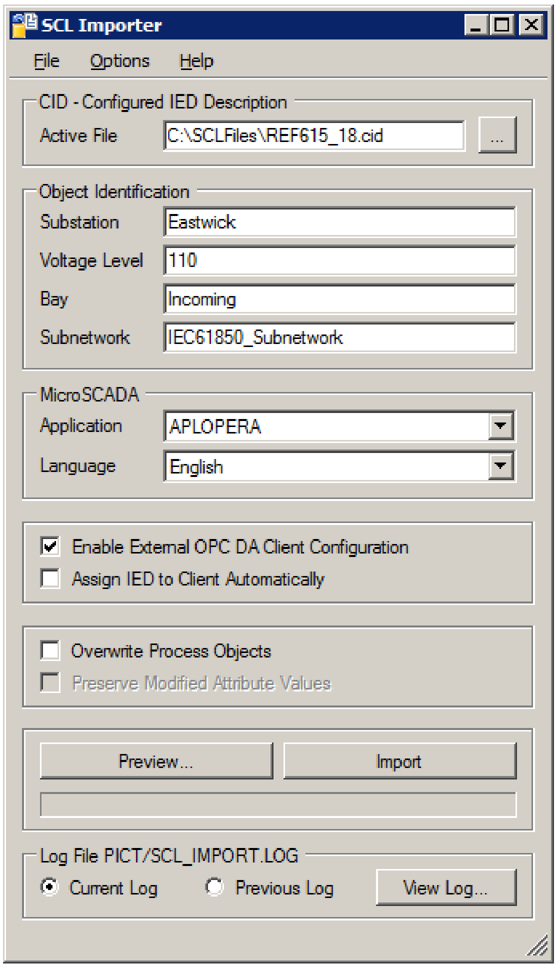
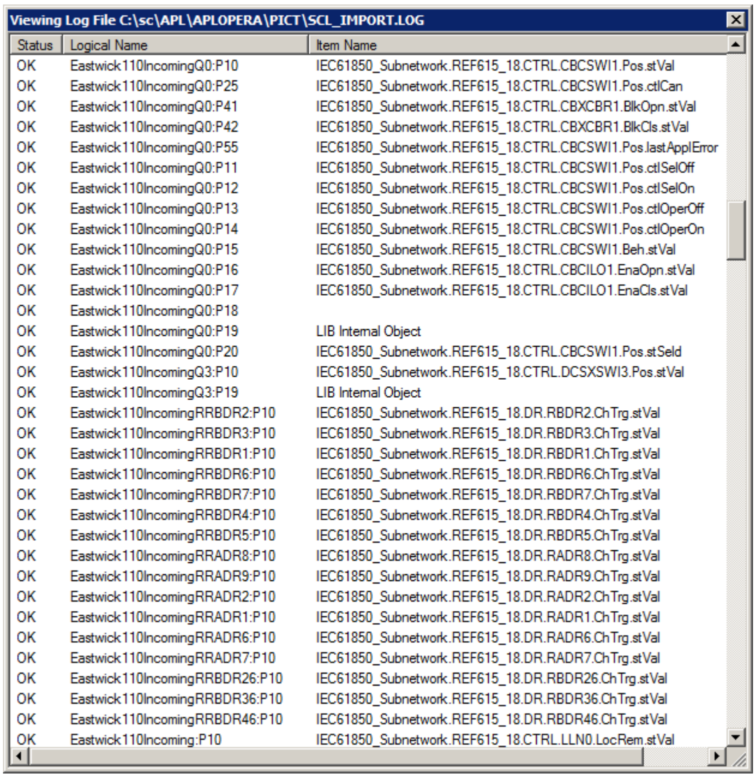


Figure 7: SCL Importer dialog

1. In the Main view, browse the SCD/CID file to be imported. If the selected file is of CID type, Object Identification needs to be filled in.
2. Select the **Application** and **Language** from the drop-down list.
3. Define the additional options found from the Main view and from the **Options** menu.
4. Click **Preview** to see details of the data being imported.
5. Click **Import**.
6. View the created process objects and their indexes by clicking **View Log**, which is activated after importing.



Viewing Log File C:\sc\APL\APLOPERA\PICT\SCL_IMPORT.LOG

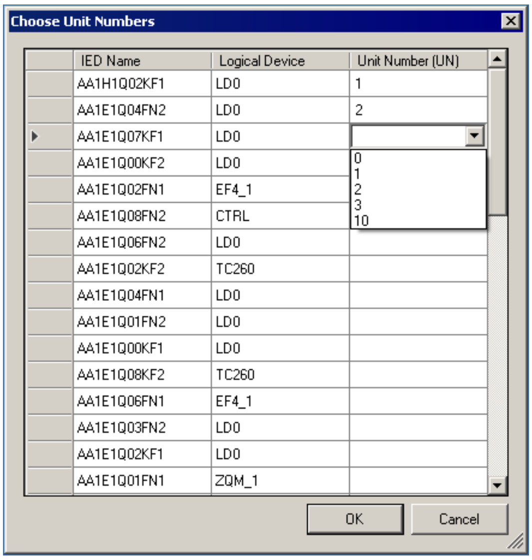
Status	Logical Name	Item Name
OK	Eastwick110IncomingQ0:P10	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Pos.stVal
OK	Eastwick110IncomingQ0:P25	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Pos.ctlCan
OK	Eastwick110IncomingQ0:P41	IEC61850_Subnetwork.REF615_18.CTRL.CBXCBR1.BlkOpn.stVal
OK	Eastwick110IncomingQ0:P42	IEC61850_Subnetwork.REF615_18.CTRL.CBXCBR1.BlkCls.stVal
OK	Eastwick110IncomingQ0:P55	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Pos.lastApplError
OK	Eastwick110IncomingQ0:P11	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Pos.ctlSelOff
OK	Eastwick110IncomingQ0:P12	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Pos.ctlSelOn
OK	Eastwick110IncomingQ0:P13	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Pos.ctlOperOff
OK	Eastwick110IncomingQ0:P14	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Pos.ctlOperOn
OK	Eastwick110IncomingQ0:P15	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Beh.stVal
OK	Eastwick110IncomingQ0:P16	IEC61850_Subnetwork.REF615_18.CTRL.CBCILO1.EnaOpn.stVal
OK	Eastwick110IncomingQ0:P17	IEC61850_Subnetwork.REF615_18.CTRL.CBCILO1.EnaCls.stVal
OK	Eastwick110IncomingQ0:P18	
OK	Eastwick110IncomingQ0:P19	LIB Internal Object
OK	Eastwick110IncomingQ0:P20	IEC61850_Subnetwork.REF615_18.CTRL.CBCSWI1.Pos.stSeld
OK	Eastwick110IncomingQ3:P10	IEC61850_Subnetwork.REF615_18.CTRL.DCSXSWI3.Pos.stVal
OK	Eastwick110IncomingQ3:P19	LIB Internal Object
OK	Eastwick110IncomingRRBDR2:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR2.ChTrg.stVal
OK	Eastwick110IncomingRRBDR3:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR3.ChTrg.stVal
OK	Eastwick110IncomingRRBDR1:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR1.ChTrg.stVal
OK	Eastwick110IncomingRRBDR6:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR6.ChTrg.stVal
OK	Eastwick110IncomingRRBDR7:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR7.ChTrg.stVal
OK	Eastwick110IncomingRRBDR4:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR4.ChTrg.stVal
OK	Eastwick110IncomingRRBDR5:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR5.ChTrg.stVal
OK	Eastwick110IncomingRRADR8:P10	IEC61850_Subnetwork.REF615_18.DR.RADR8.ChTrg.stVal
OK	Eastwick110IncomingRRADR9:P10	IEC61850_Subnetwork.REF615_18.DR.RADR9.ChTrg.stVal
OK	Eastwick110IncomingRRADR2:P10	IEC61850_Subnetwork.REF615_18.DR.RADR2.ChTrg.stVal
OK	Eastwick110IncomingRRADR1:P10	IEC61850_Subnetwork.REF615_18.DR.RADR1.ChTrg.stVal
OK	Eastwick110IncomingRRADR6:P10	IEC61850_Subnetwork.REF615_18.DR.RADR6.ChTrg.stVal
OK	Eastwick110IncomingRRADR7:P10	IEC61850_Subnetwork.REF615_18.DR.RADR7.ChTrg.stVal
OK	Eastwick110IncomingRRBDR26:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR26.ChTrg.stVal
OK	Eastwick110IncomingRRBDR36:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR36.ChTrg.stVal
OK	Eastwick110IncomingRRBDR46:P10	IEC61850_Subnetwork.REF615_18.DR.RBDR46.ChTrg.stVal
OK	Eastwick110Incoming:P10	IEC61850_Subnetwork.REF615_18.CTRL.LLN0.LocRem.stVal

Figure 8: Viewing log file

3.2.4.2 Import options in Main view

External OPC DA Client Configuration

If **Enable External OPC DA Client Configuration** is checked in the Main view, a dialog asking for the unit number(s), i.e. station number of MicroSCADA base system STA object is shown when importing is started. The Unit number can be selected from the drop down list or written directly into the field.



Choose Unit Numbers

IED Name	Logical Device	Unit Number (UN)
AA1H1Q02KF1	LD0	1
AA1E1Q04FN2	LD0	2
AA1E1Q07KF1	LD0	
AA1E1Q00KF2	LD0	
AA1E1Q02FN1	EF4_1	
AA1E1Q08FN2	CTRL	
AA1E1Q06FN2	LD0	
AA1E1Q02KF2	TC260	
AA1E1Q04FN1	LD0	
AA1E1Q01FN2	LD0	
AA1E1Q00KF1	LD0	
AA1E1Q08KF2	TC260	
AA1E1Q06FN1	EF4_1	
AA1E1Q03FN2	LD0	
AA1E1Q02KF1	LD0	
AA1E1Q01FN1	ZQM_1	

Figure 9: Select unit number

In case of SCD files, unit numbers are given at this phase for all IEDs found from the file.

Once the Unit Number(s) are defined, a dialog for assigning properties for External DA Client Configuration file is shown.

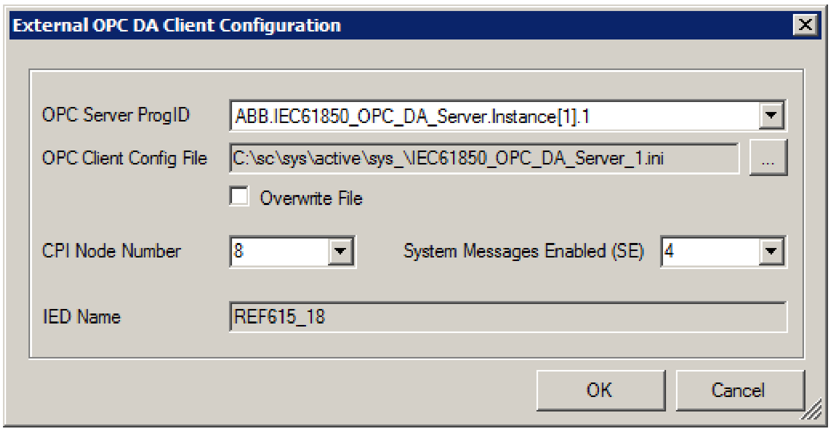


Figure 10: External OPC DA Client Configuration

OPC Server ProgID lists available OPC DA Servers. If none is found, then option <Unknown IEC 61850 Server> is available. In this case, the External OPC DA Client configuration file can be created, but the correct IEC 61850 OPC server information has to be filled in later on to the OPC Client Config file.

After the IEC 61850 OPC server configuration is done, server information can be defined with External OPC Client Configuration Tool.

If **Assign IED to Client Automatically** is selected in the Main view, MicroSCADA SCL Importer resolves the correct OPC Server ProgID for the IED.

Algorithm for resolving the correct OPC servers assumes that IED name is unique within the IEC 61850 OPC server configuration.

OPC Client Config File defines the name and location of the configuration file.

If **Override File** is checked, a new file for External OPC DA Client is created. This file will contain only the configuration defined in imported SCD/CID file. Backup is taken from the existing one.

CPI Node Number is the node number of the External DA Client in MicroSCADA configuration.

System Messages Enabled (SE) defines how system messages (used for System Self Supervision purposes) are updated to MicroSCADA process database. When 1 is selected, analog status points are updated. When 4 is selected, both analog and binary status points are updated.

As the status for the IEDs is received as binary information, 4 should be used.

In the **IED Name** field, the name of the IED being configured is shown.

If **External OPC DA Client Configuration** is selected, MicroSCADA base system NOD and STA objects should be created, and IEC 61850 OPC Server configuration should be made before importing.

Override Process Objects

When **Overwrite Process Objects** is selected in the Main view, process object that has the same Logical Name (LN) that is being created by MicroSCADA SCL Importer is deleted and then re-created.

Preserve Modified Attribute Values
If **Preserve Modified Attribute Values** is selected, a copy is taken from the existing process object before it is deleted, and the attributes of the copied object and a new object are merged. This way, for example, if a certain Event Handling object was connected to a process object, the same configuration is also found from the new process object.

3.2.4.3 Options menu

Import settings
In **Options/Import Settings**, there is a possibility to affect to the amount of process objects that are created, and to enable or disable the possibility to affect the logical name of process objects.

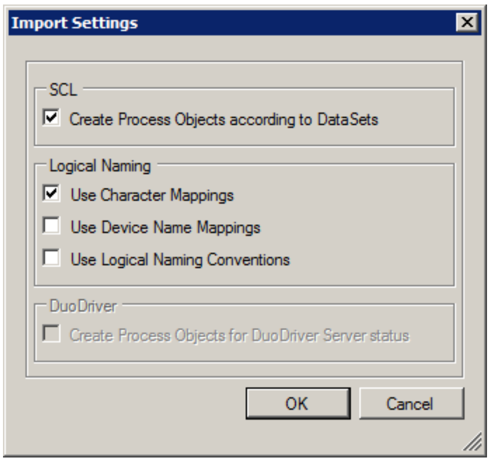


Figure 11: Import Settings

When **Create process Objects according to Datasets** is selected, the program checks that a Logical Node (for example, CSWI) is found both from substation section and data set in CID/SCD file. If both conditions are fulfilled, process objects are created for the Logical Node in question.

If **Create process Objects according to Datasets** is not selected, process objects are created for all Logical Nodes found from the substation section. This means that some of the process objects may not ever receive event based data from process.

When **Use Character Mappings** is selected, the mappings made in the **Options/Character Mapping** dialog are taken into account when creating process objects.

When **Use Device Name Mappings** is selected, the mappings made in the **Preview/Device Name Mapping** dialog are taken into account when creating process objects.

When **Use Logical Naming Conventions** is selected, the rules defined in the **Options/Logical Naming Conventions** dialog are taken into account when creating process objects.

Create Process Objects for DuoDriver Server status is enabled only if DuoDriver is installed. When this function is used, process objects for DuoDriver server supervision are created during the SCL import. Preferably dedicated Unit numbers should be assigned for DuoDrivers, since they do not belong to any IED.

Character mapping
During import, the logical name of a process objects is formed from Substation, Voltage Level and Bay defined in the SCL Importer Main view. Also, this information is set to the object identification (OI attribute).

For process objects, the allowed characters are the letters A-Z, all digits and the underscore (_). However, there can be cases when some special characters are wanted to be used in object identification e.g. "ä". In those cases, the characters that are not allowed in the logical name of process objects can be converted into allowed characters with the help of the **Character Mapping** dialog.

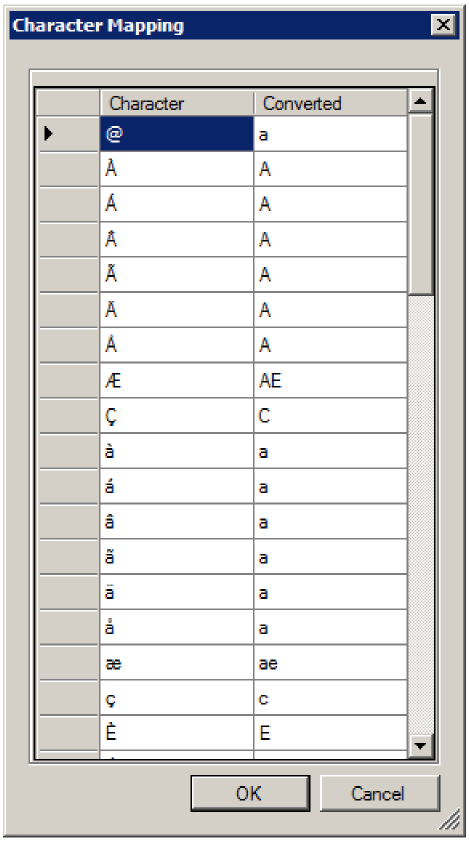


Figure 12: Character Mapping

By default, some character conversion is shown. In order to add a new character conversion, click the last shell in the **Character** column and add the character to be converted. Then, click the corresponding **Converted** column field and fill in the converted character.

Logical naming conventions

With the Logical Naming Conventions dialog, it is possible to affect the length of the logical name of the process objects.

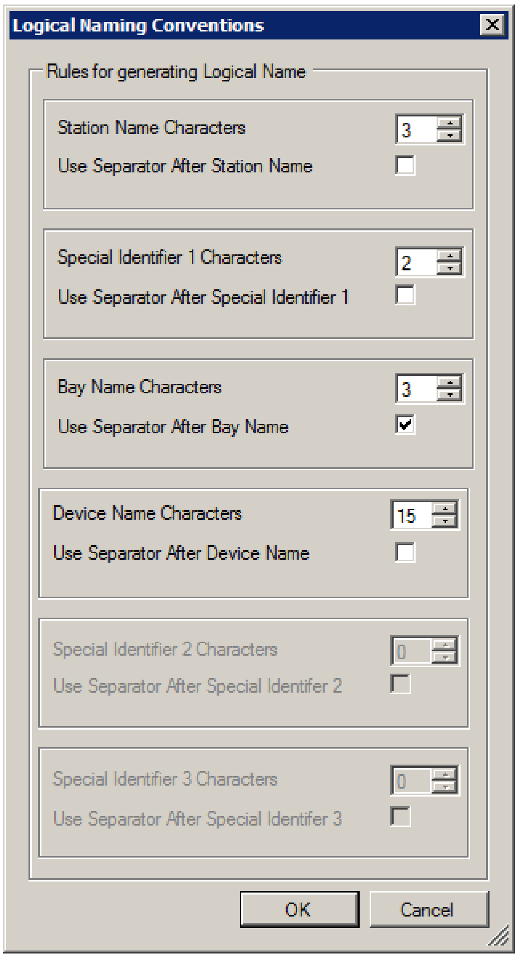


Figure 13: Logical Naming Conventions dialog

The number of characters taken to the process object name from Station, Voltage Level, Bay and Device in Main view can be defined.

For example, if Station Name Characters is 3, the three first letters from a station name is used in process object name and so on.

It is also possible to define a separator. If this option is selected, the underscore (_) character will be used as a separator.

Device Name Mapping

With Device Name Mapping, it is possible to affect the device name part of object identification and the logical name of the process objects. By default, the IEC61850 Logical Node information found from the CID/SCD file is used for this part, or a device name is created by internal algorithms in MicroSCADA SCL Importer.

Device Name Mapping dialog can be opened from the **Import Preview** dialog. All device names found from the CID/SCD file are shown on the left side, and the converted value can be defined on the right side.

In order to change the device name, click the field in the **Converted** column and type in a new device name.

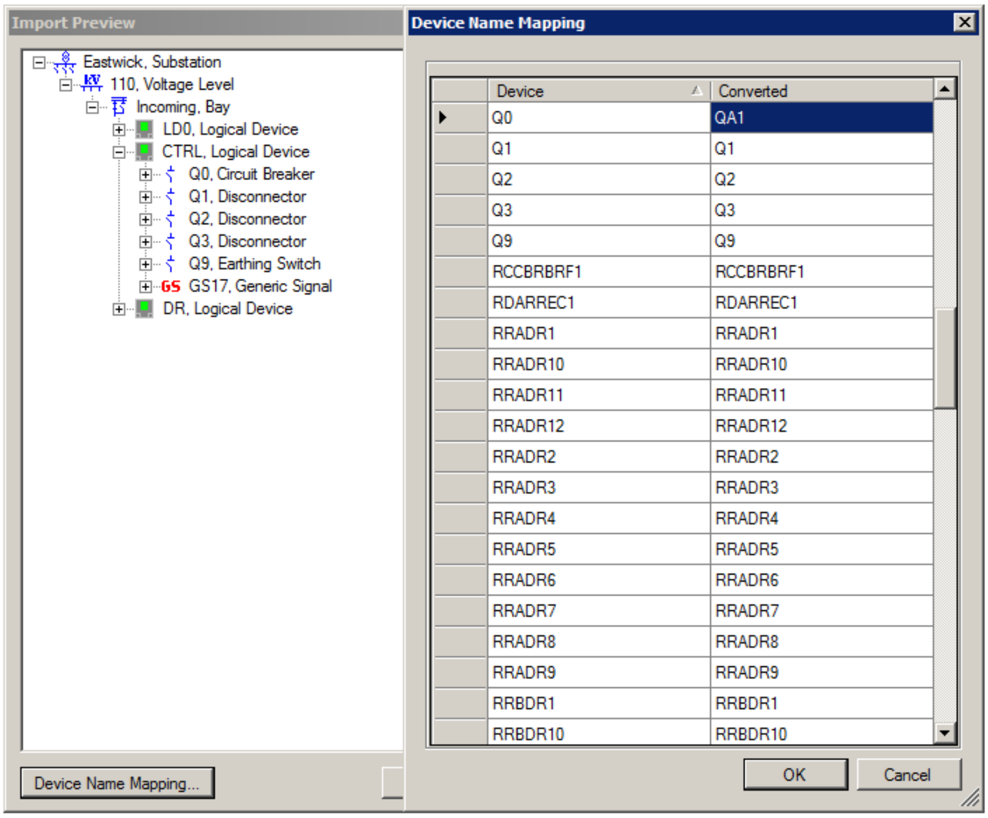


Figure 14: Device Name mapping

3.2.5 Using Input Objects

Input Object is an object responding to the user's action, for example a mouse click. It writes data to the MicroSCADA process database and can have several appearances, such as a button or a slider.

When the data is written from the Input Object to the process database, it is done in two phases, which are always required to write data to the process database:

1. First, the Input Object sets the data variable to a certain value, for example, when the mouse moves above the Input Object slider. The data variable value is changed accordingly, but the value is not yet written into the process database. The data variable for the set operation can be defined by selecting the data variable in the **Input Object Properties** dialog.
2. The data variable is written to the data source (MicroSCADA OPC server). This happens when the user clicks on the object. The data variable for writing can be defined in the **Rules** tab of the Input Objects Properties dialog, see Figure 18.

There are two ways to create an Input Object and connect it to the process database:

- Set a data variable to a certain value and write the value to a data source in the same object, as described in the example below.
- Do it in separate objects. For example, there can be two buttons: the first button sets a variable1 to value x, and the second button writes the value of variable1 (x) to variable2.

3.2.5.1 Creating an Input Object and connecting it to process database:

1. Click the **Input Object** icon on the Objects toolbar of Display Builder. See Figure 15.

Figure 15: Input Object button

2. 2. Select the location for the Input Object by dragging a rectangle on the drawing area. Now you can see the Input Object on the drawing area, see Figure 16.

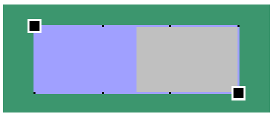


Figure 16: Input Object on the drawing area

3. Right-click the Input Object and select **Input Object Properties** to open the Input Object Properties dialog. The data variable that is mapped to the object is displayed in the Variables field, see Figure 17.

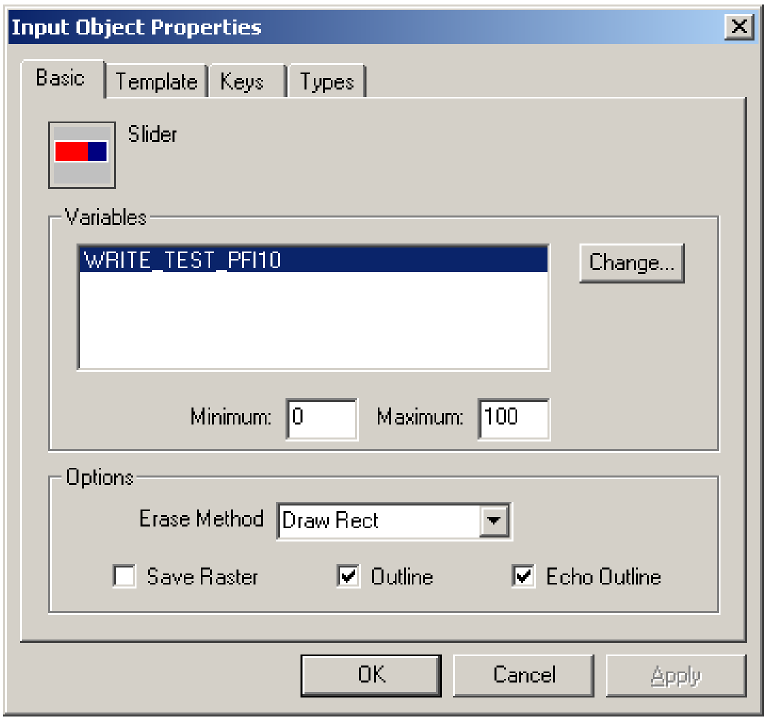


Figure 17: Mapped data variable

4. To change the mapped data variable, click Change and select the data variable.
5. Click **OK**.
6. Right-click the Input Object and select **Properties**.
7. Click the **Create a New Rule** button on the Rules tab Figure 18.

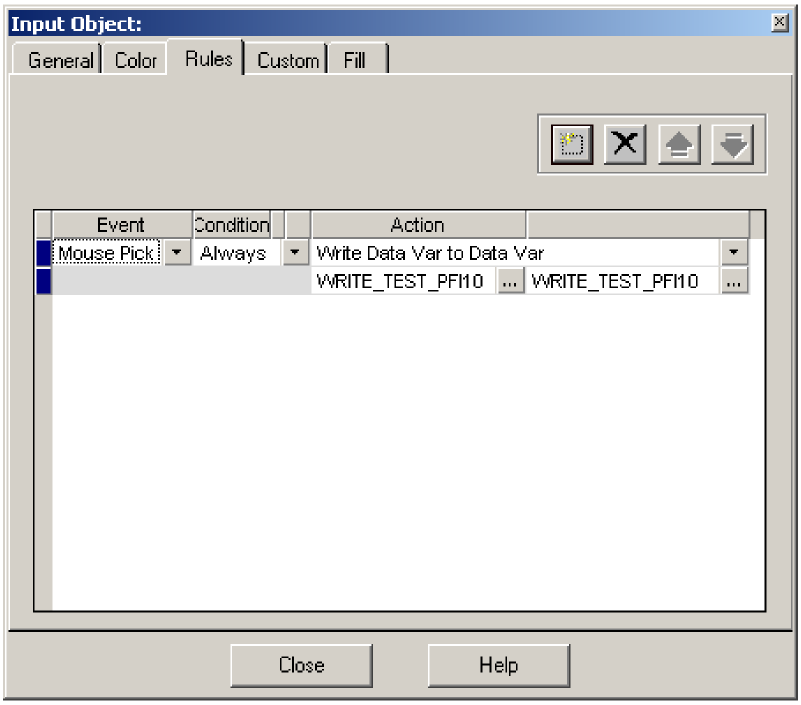


Figure 18: Rules tab of the Input Object dialog

8. Select **Write Data Var to Data Var** from the drop-down list in the Action column to open the Data selection for the Writable Vars dialog.
9. Select the data variable you want to write to. You can select the same variable that was defined in the **Input Object Properties** dialog.
10. The **Data Selection** dialog is displayed. Select the variable from where the data is read in this dialog. This can be the same data variable that was used as the target data variable.

As default, Input Object is shown as a slider. To change its layout, right-click the Input Object and select the **Types** tab in the Input Object Properties dialog and select another layout.

The different types of Input Objects may require a different count of mapped data variables.

Section 4 Defining application options

To define application options, select **Settings/Application Settings...**. The user can specify application specific options, such as:

- System name
- System location
- Startup display, first display shown after login
- Application language
- Daylight Saving time handling
- Process controlling dialog settings
- Hard disk space limit supervision

To modify these options, authorization level must be Engineering (2). For more information see, Section 24.

4.1 Application settings

Application settings can be modified on the **Application Settings** tab.

The application name is shown in the title bar of the **Application Settings** dialog. The name of the application cannot be changed in the **Application Settings** tab, see Figure 19.

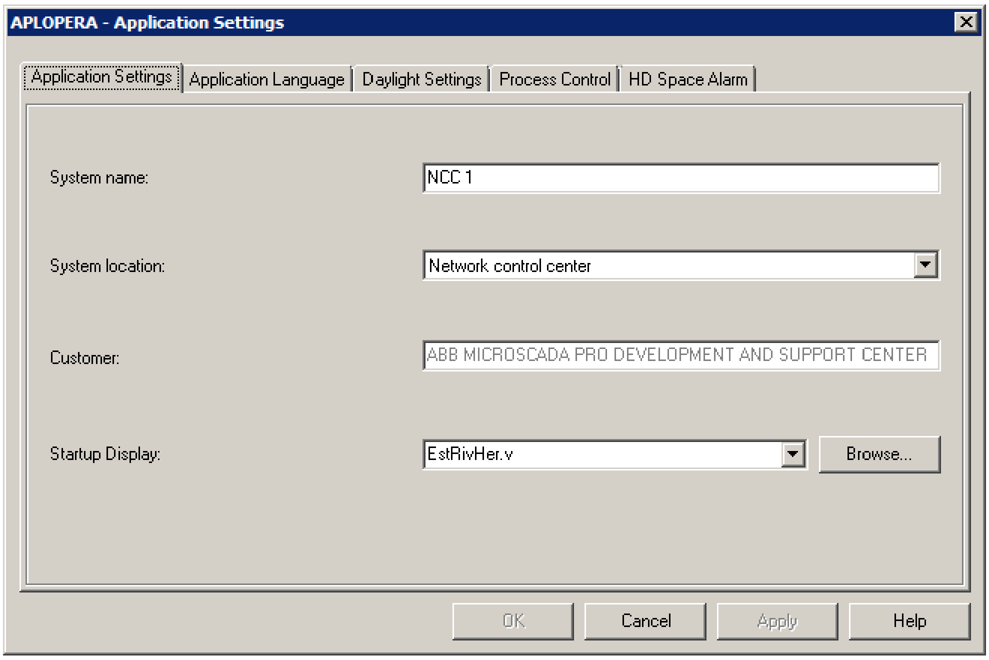


Figure 19: Application settings

A certain Process Display can be defined to be shown after logging by browsing the Process Display to the **Startup Display** box. The Process Display can be any .v or .sd type. It is also possible to define other displays by clicking the **Startup Display** box. Select the Display and add the existing preconfiguration name after the display, for example Event Display MyPreconfiguration.

System location is an attribute that needs to be configured. This defines whether SYS600 is running as Network Control Center (NCC) or Substation Control System (SCS).

To be able to make any changes in the application settings, the user's authorization level must be Engineering (2) in the authorization group PRO_OPTIONS_HANDLING. For more information about authorization, see Section 21.

To save the new settings, click **OK** or **Apply**. Clicking **Cancel** returns the previous settings.

4.2 Application Language

Language settings can be modified on the **Application Language** tab. The default language is English and the following customized languages (if any) are defined as local languages.

When printing to an event printer, the application language is used. The application language is the same for all the users within the same application. The application language does not have any effect on the language shown in the monitors.

Application language can be selected from the drop-down list, see Figure 20. To save the new settings, click **OK** or **Apply**. Clicking **Cancel** returns the previous settings.

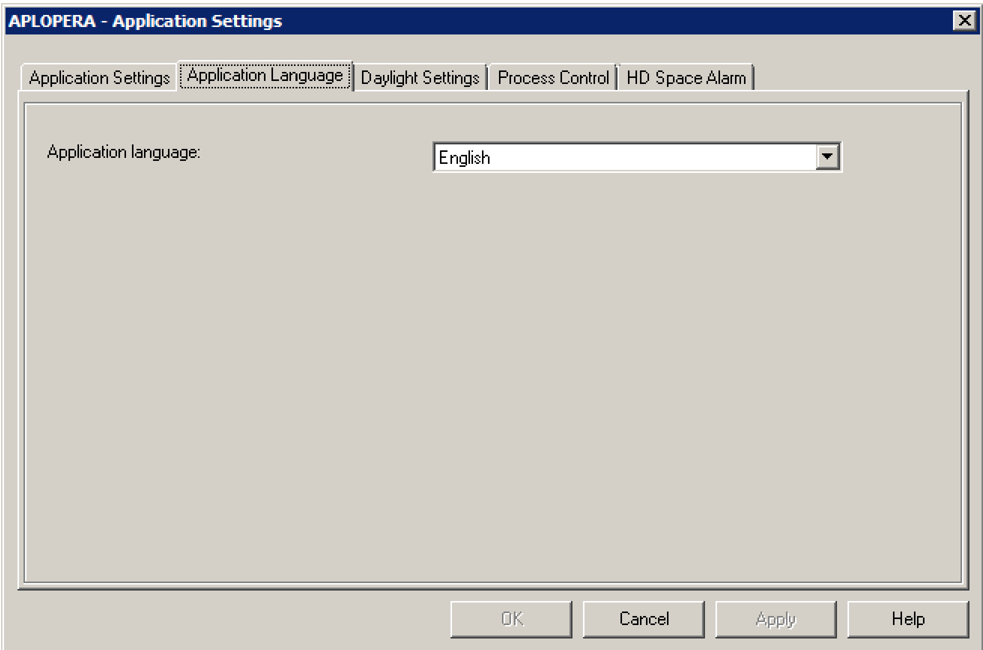


Figure 20: Application Language tab

4.3 Daylight settings

If the daylight saving time function is set to be in use in the operating system, the system time will be automatically adjusted to normal time and to daylight saving time. It is possible to do application specific actions in MicroSCADA X during the time change if **Automatically adjust applications for daylight saving time changes** is enabled in Application Settings, see Figure 21.

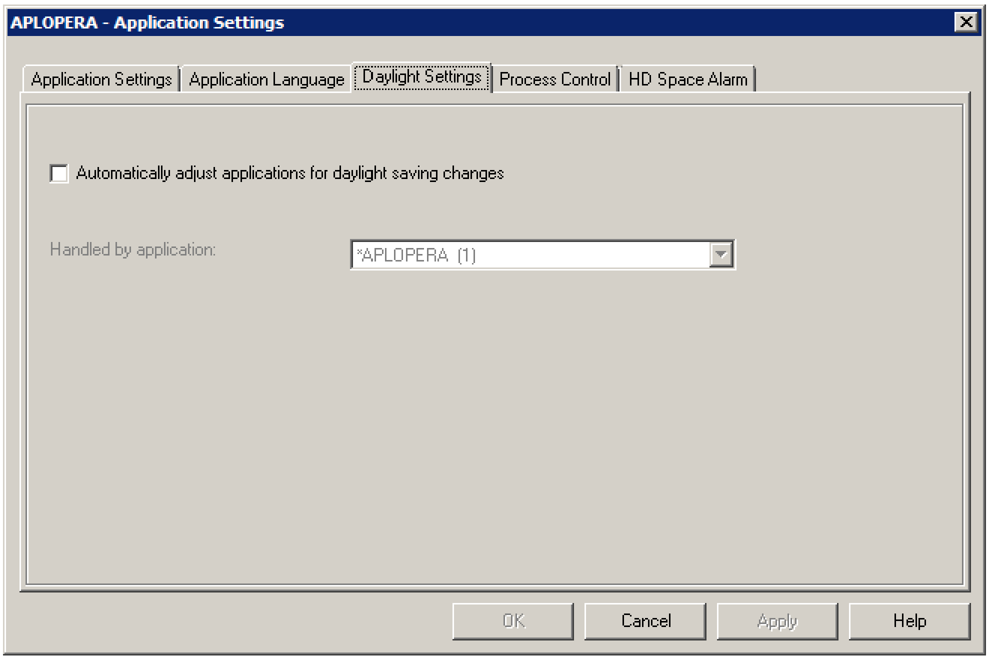


Figure 21: Daylight Settings tab

When **Automatically adjust applications for daylight saving time changes** option is enabled, application specific actions can be defined in the following command procedures:

- BGU_TIMEFO:C, executed when the clock is moved forwards (Daylight saving time activated).
- BGU_TIMEBA:C, executed when the clock is moved backwards (Daylight saving time deactivated).

The point of execution time is 20 seconds after the change. Command procedures are executed in each found LOCAL application according to the application mapping.

When the **Automatically adjust applications for daylight saving time changes** option is enabled, the application handling the actions can be selected from the drop-down list. To save the new settings, click **OK** or **Apply**. Clicking **Cancel** returns the previous settings.

4.4 Process control settings

The process control settings can be defined on the **Process Control** tab. The user can set the behavior for process control when operations are executed, see Figure 22. When the Close Control Dialog after execution option checkbox is checked, the control dialogs, for example the breaker dialogs, are closed after the operation is done.

Control timeout defines the timeout for control confirmation dialog used to execute or cancel operations. With Dialog timeout, it is possible to define the user inactivity time in the desktop before the Control Dialog expires.

Horizontal offset and Vertical offset values configure the position of the Control Dialog.

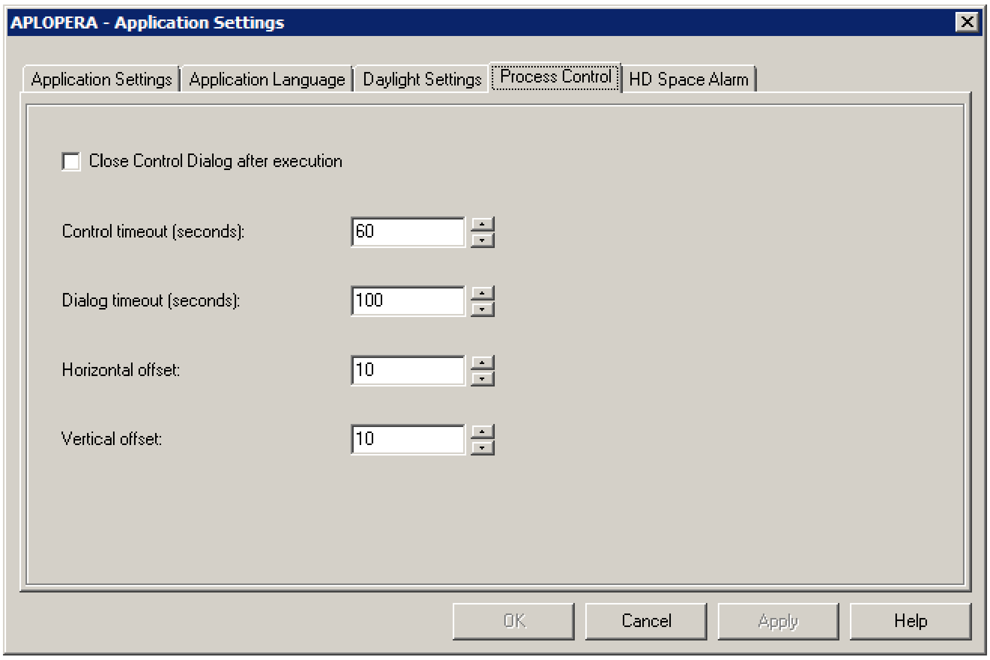


Figure 22: Process Control tab

To activate or deactivate option, select or clear the check box. To save the new settings, click **OK** or **Apply**. Clicking **Cancel** returns the previous settings.

4.5 HD space alarm

The hard disk supervision function can be defined on the **HD Space Alarm** tab to supervise the free hard disk space. With this function, the user receives an early warning if there is a risk of running out of space on the hard disk, see Figure 23.

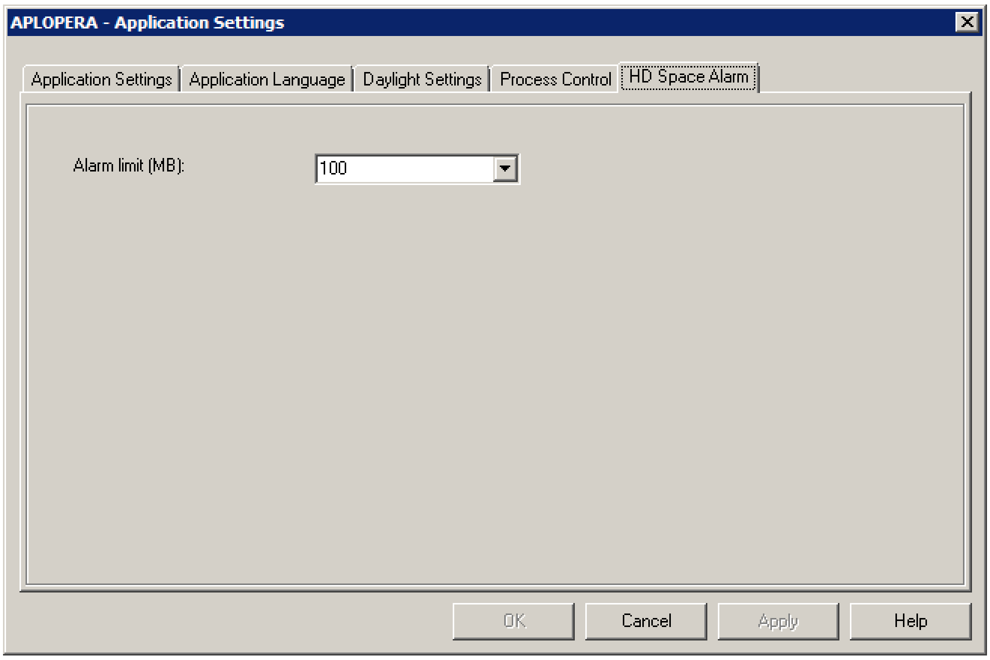


Figure 23: HD space alarm dialog

The user can give a lower alarm limit for disk space by selecting the alarm limit from the **Alarm limit** drop-down list. The required limit can also be typed in. The limit is presented as megabytes.

If the alarm limit is set to zero (0), the hard disk supervision is stopped. A warning limit is calculated from the alarm limit. The warning limit is 10% higher than the low alarm limit.

To save the new settings, click **OK** or **Apply**. Clicking **Cancel** returns the previous settings.

4.6 Customizing time formats

Date format for Alarm and Event Display, Alarm Row and time fields in Monitor Pro can be customized individually for each user.

Date format is defined by the key FreeDateTimeField in the DAYFORMAT section of the FrameWindow.ini configuration file:

[DAYFORMAT]

;Datetime format can be, for example "%Y-%m-%d %H:%M:%S" (note the quotation marks). If not defined, the format is received from MicroSCADA (when logged in).

FreeDateTimeField=

The application specific configuration file is located at \sc\apl\<application name>\PAR\APL \FrameWindow.ini.

The user specific configuration file is located at \sc\apl\<application name>\PAR\<user name>\FrameWindow.ini.

Quotation marks must be used in the date format string.

If FreeDateTimeField key is empty (default), the time format is taken from the SYS:BTF (Time Format) attribute:

- If SYS:BTF is 0, then format "%Y-%m-%d %H:%M:%S" is used.
- If SYS:BTF is 1, then format "%m-%Y-%d %H:%M:%S" is used.
- If SYS:BTF is 2, then format "%m-%d-%Y- %H:%M:%S" is used.

If the %S specifier is used and milliseconds are available for the attribute in question, milliseconds are automatically added to the formatted string in Event Display and Alarm Display. Milliseconds are displayed immediately after %S, separated with period (.).

Customize dialog can be user for changing the user specific configuration:

1. Select **Settings/Customize...** to open the **Customize** dialog.
2. Right click the time on Status Bar, see Figure 24.

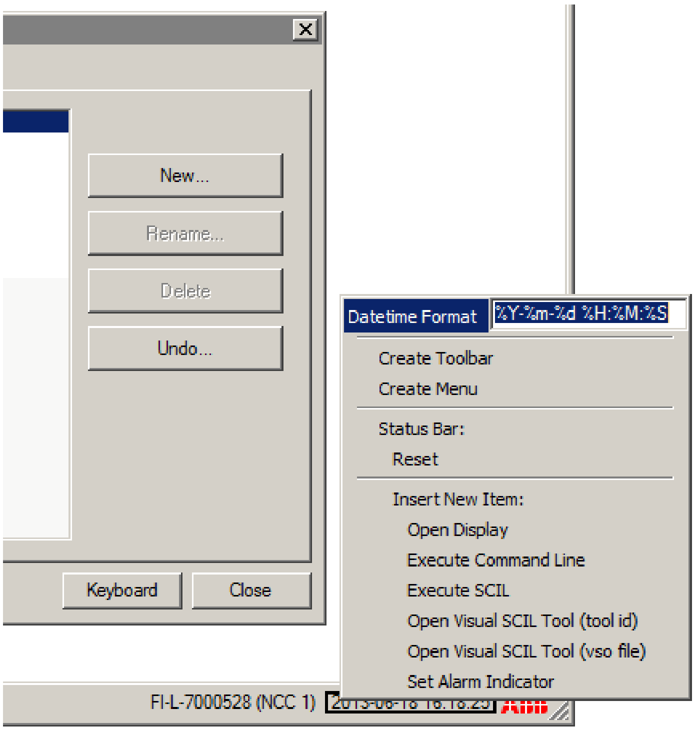


Figure 24: Customizing user specific time format

3. Type in the desired time format.
4. The time format is set to the configured one.
5. Click **Close** to exit the **Customize** dialog.

The following format specifiers can be used in format string. Notice that month and weekday names are translated according to the operating system's user locale setting.

Table 1: Custom time format specifiers

Specifier	Description
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%d	Day of month as decimal number (01–31)
%H	Hour in 24-hour format (00–23)
%I	Hour in 12-hour format (01–12)
%m	Month as decimal number (01–12)
%M	Minute as decimal number (00–59)
%p	Current locale's A.M./P.M. indicator for 12-hour clock
%S	Second as decimal number (00–59)
%y	Year without century, as decimal number (00–99)
%Y	Year with century, as decimal number

Any other characters are added to resulting string as given.

Section 5 SQL Exporter

SQL Exporter is used to get an HSI (human-system interface) for selecting process objects to be sent to an SQL database. With SQL Exporter, a process object can be connected to a history database, a measurement database (cyclic) or to both.

To set up the connection between the SQL server and SYS600:

1. Create a database.
2. Create tables for cyclical measurement recording and history recording.
3. Create user accounts.
4. Enable communication between the SQL database and SYS600.
5. Edit command procedures.
6. Test the user accounts and database.

5.1 Using SQL Exporter

To open SQL Exporter, click on **SQL Exporter** in the **Application Objects** tab of **Tool Manager**. The left pane of the SQL Exporter dialog contains process objects in the SYS600 database, see Figure 25. The process objects are filtered according to the options specified in the filter options below. The right pane contains the process objects that are connected to the SQL database, also filtered according to the filter options below. In the middle of the dialog, there are buttons for adding and removing objects to/from the SQL database.

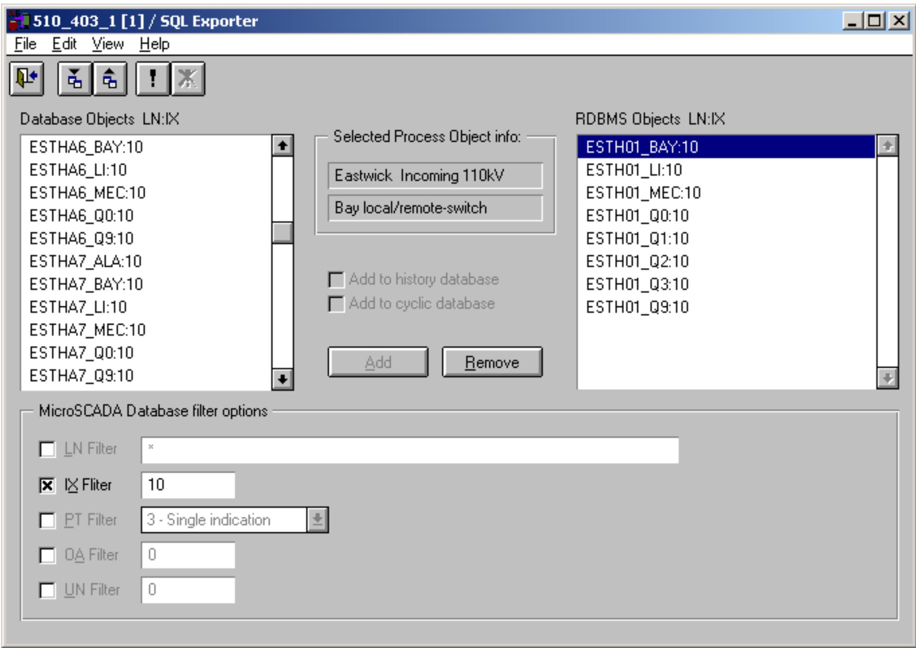


Figure 25: SQL Exporter

5.1.1 Menu bar

The main menu bar contains four menus: **File**, **Edit**, **View**, and **Help**.

- **File** contains three menu commands:
  - **Exit**
  - **Import from file**
  - **Export to file**

For more information about the import and export functions, see Section 5.9 and Section 5.12.

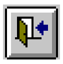
- **Edit** has three menu commands:
  - **Clean RDBMS Objects**
  - **SQL Logging enabled** and **SQL Logging disabled**

For more information about the functions, see Section 5.13 and Section 5.5.1.

- **View** contains one menu command:
  - **Toolbar**, which hides the main toolbar.
- **Help** contains one menu command
  - **About SQL Exporter**, which provides information about SQL Exporter and the SYS600 system, such as version numbers and license information.

5.1.2 Toolbar

Table 2: Toolbar buttons

Toolbar button	Description
	Use **Exit** for exiting the program. The same command can be found in **File/Exit**.
	Use **Import** for importing the CSV files (comma delimited). The same command can be found in **File/Import from file**. For more information about importing files, see Section 5.12.
	Use **Export** for exporting the CSV files. The same command can be found in **File/Export to file**. For more information about exporting files, see Section 5.9.
	Use **Enable** to enable the SQL logging. The same command can be found in **Edit/SQL Logging enabled**.
	Use **Disable** to disable the SQL logging. The same command can be found in **Edit/SQL Logging disabled**.

5.1.3 Dialog selections

- Database Objects
  Process objects in the database are filtered according to the database filter options. When an object is selected, the object's OI (Object Identifier) and OX (Object Text) are shown in the Selected Process Object field. If the object can be added to the SQL system, **Add** is activated.
- Selected Process Object info
  When a process object in the Database Objects or the objects list is clicked, the object is displayed in this field. If several objects are selected, the field is empty.
- RDBMS objects
  The process objects connected to the SQL database are listed under the objects. When a process object is selected, information about the object is displayed in the Selected Process Object info field and **Remove** is activated.
- Database selection
  To choose whether the object should be connected to the history database, the cyclic database or both, select one of the check boxes or both. SQL Exporter suggests the

history database for indications and alarms and the cyclic database for analog values. If more than one object is selected, no suggestions are made.

- The **Add** and **Remove** buttons
  To add or remove objects that are to be connected to the SQL system, use the **Add** or **Remove** buttons. These buttons are enabled and disabled according to what is allowed to do with the selected process objects. For more information about adding or removing process objects, see Section 5.8.1 and Section 5.8.2.
- Process object filter options
  The five optional filters can be used one at a time or together. The selected filters apply to both the process objects and to the SQL process objects lists. For more information about the filter options, see Section 5.8.4.

5.2 Creating database

To create a database:

1. Install Microsoft SQL 2000 Server.
2. Select **Start/Programs/Microsoft SQL Server/Enterprise Manager** to start the Enterprise Manager of the SQL Server.
3. Create a new database by right-clicking the server and selecting **New/Database**, see Figure 26.

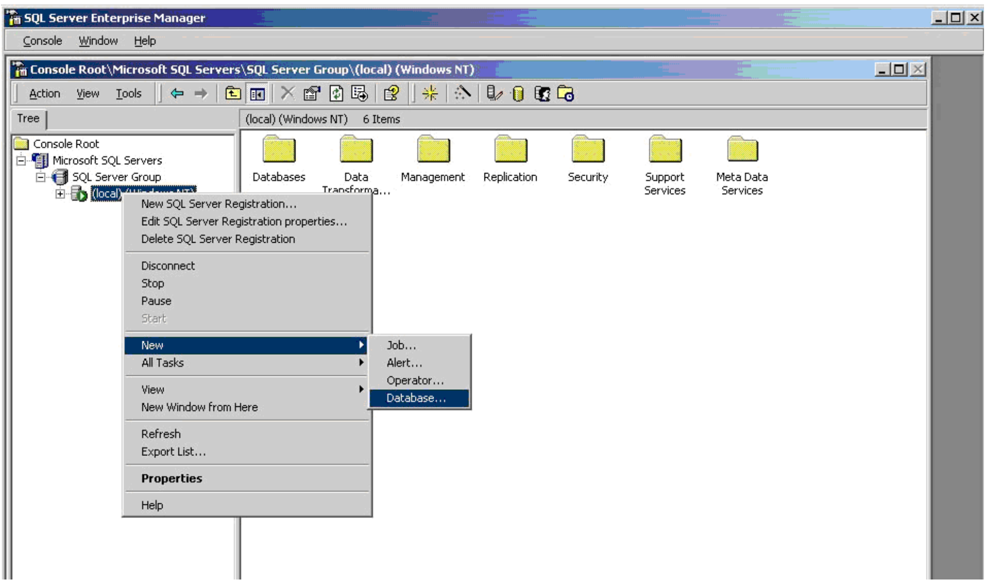


Figure 26: Creating a new database

4. Name the database, for example SCADA.

5.3 Creating cyclical measurement recording table

The cyclically sent values are saved in the process object table (MSPDB). Typically, values are measurements. The SYS600 command procedure updates the values every 15 minutes and a new value is inserted to a new row in the table. If SYS600 does not send the process objects, the process objects are no longer updated. The old process object is not removed.

A unique ID consists of:

- LN
- IX
- Moment
- Moment_IX

If several signals from a process object have the same LN and IX, the values are indexed with the Moment_IX time stamp.

To create a table:

1. Open the Enterprise Manager and create a new table.
2. Name the table columns for the database as shown in Figure 27.

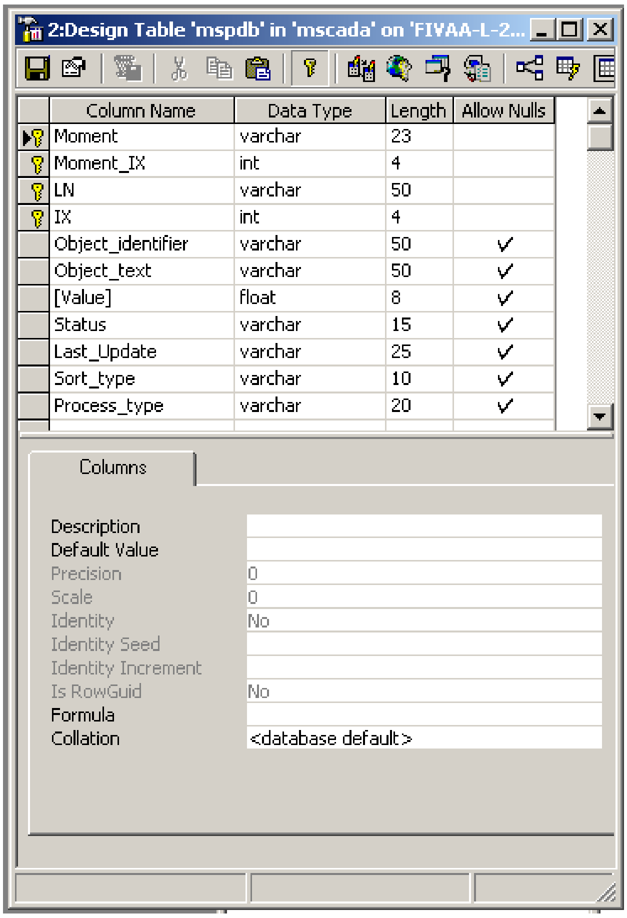


Figure 27: Defining table for the cyclical recording

3. Save the table as a MSPDB.
4. Run the following code to create a table. The table can be created with, for example, the Query Analyzer.

```
CREATE TABLE [mspdb] (
  [LN] [varchar] (50) NOT NULL ,
  [IX] [int] NOT NULL ,
  [Moment] [varchar] (23) NULL ,
  [Object_identifier] [varchar] (50) NULL ,
  [Object_text] [varchar] (50) NULL ,
  [Value] [float] NULL ,
  [Status] [varchar] (15) NULL ,
  [Last_Update] [varchar] (25) NULL ,
  [Sort_type] [varchar] (10) NULL ,
  [Process_type] [varchar] (20) NULL
) ON [PRIMARY]
```

The user must have engineering rights to run the code.

Figure 28 describes how an update of a process object's value is distributed to the cyclic database, that is, to the MSPDB SQL table.

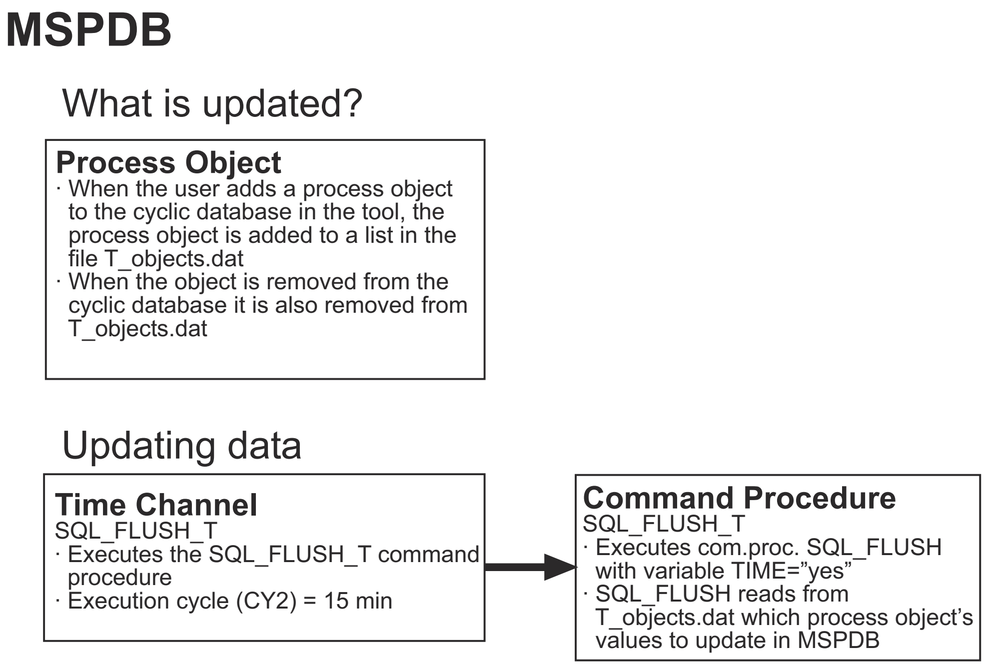


Figure 28: Information flow to MSPDB

The SQL table registers changes only. If the states of the process objects are static, nothing is updated in the tables.

5.4 Creating history recording table

The history and event values are saved in the process object table (MSPDB_hist). Typically, values are indications or alarms. The values are saved in the SYS600 queue file and sent to the history table every five seconds. Every value from the queue file is added to the table. Therefore, there can be several entries from the same process object. The new rows are added until the table size reaches the limit of the SQL system.

A unique ID consists of:

- LN
- IX
- RT
- RT_IX

If several signals from a process object have the same LN and IX, the values are indexed with the RT_IX time stamp.

To create a table:

1. Open the Enterprise Manager and create a new table.
2. Name the table columns for the database as shown in Figure 29.

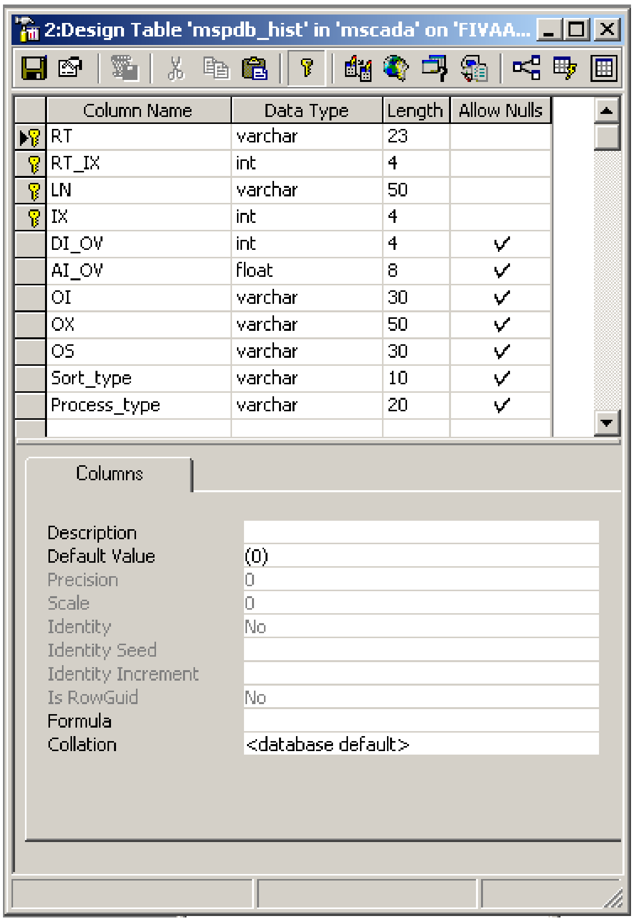


Figure 29: Defining table for the history recording

3. Save the table as a MSPDB_hist.
4. Run the following code to create a table. The table can be created, for example, with the Query Analyzer.

```
CREATE TABLE [mspdb_hist] (
  [LN] [varchar] (50) NOT NULL ,
  [IX] [int] NOT NULL ,
  [DI_OV] [int] NULL ,
  [AI_OV] [float] NULL ,
  [RT] [varchar] (23) NULL ,
  [OI] [varchar] (30) NULL ,
  [OX] [varchar] (50) NULL ,
  [OS] [varchar] (30) NULL ,
  [Sort_type] [varchar] (10) NULL ,
  [Process_type] [varchar] (20) NULL
ON [PRIMARY]
```

The user must have engineering rights to run the code.

When the user adds a process object to the history database (MSPDB_Hist SQL table) in SQL Exporter, the process object is connected to the SQL_WRITE_DB_EVENT event channel. Figure 30 describes how, for example, a change in an indication object is distributed to the history database.

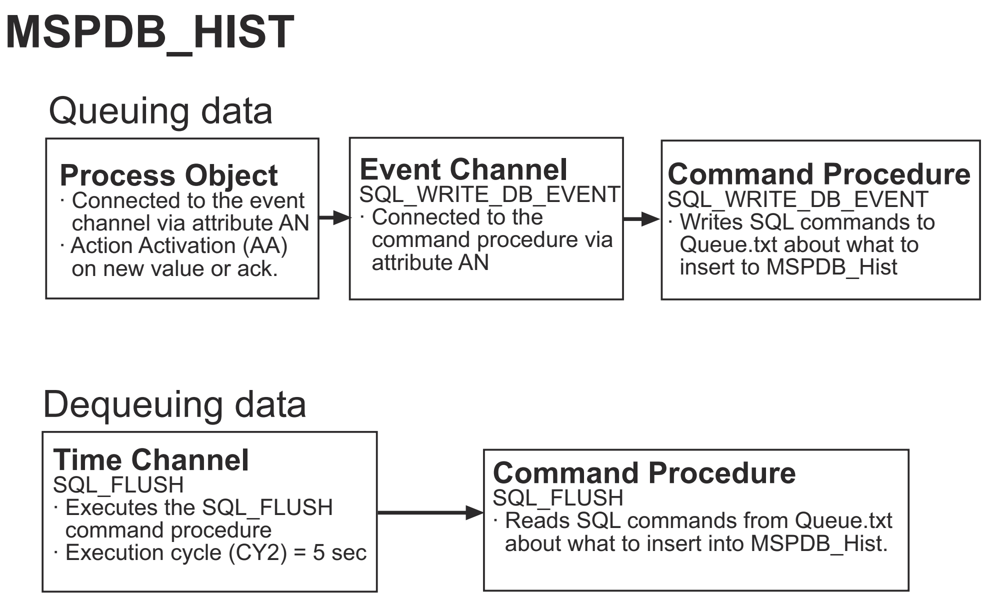


Figure 30: Information flow to MSPDB_Hist

The SQL tables register changes only. If the states of the process objects are static, nothing is updated in the tables.

5.5 Creating user accounts

To create a new user:

1. Select **<new>** from the login name drop-down list to create a new user account, see Figure 31.

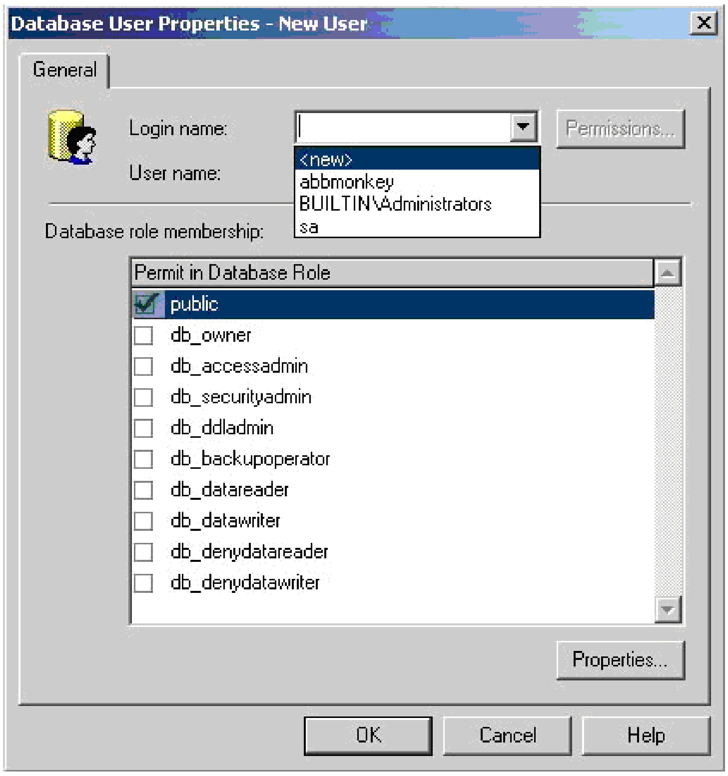


Figure 31: Creating a new user

2. Define options for the new SQL user, see Figure 32.
3. Type the user name in the **Name** field.
4. Select the SQL Server Authentication option from the **Authentication** field.
5. Type the default password in the **Password** field.
6. Select a database created before, for example SCADA, from the **Database** drop-down list.
7. Select the **Database Access** tab.

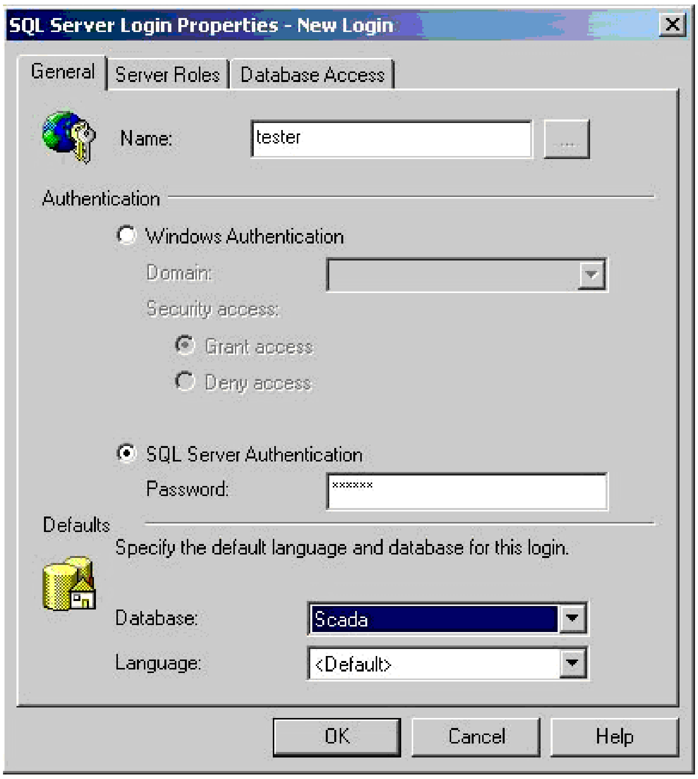


Figure 32: Defining a new SQL user options

8. Define the database that the user can access by selecting the database defined before, see [Figure 33](#).
9. Select the new user from the **Database Role** field.

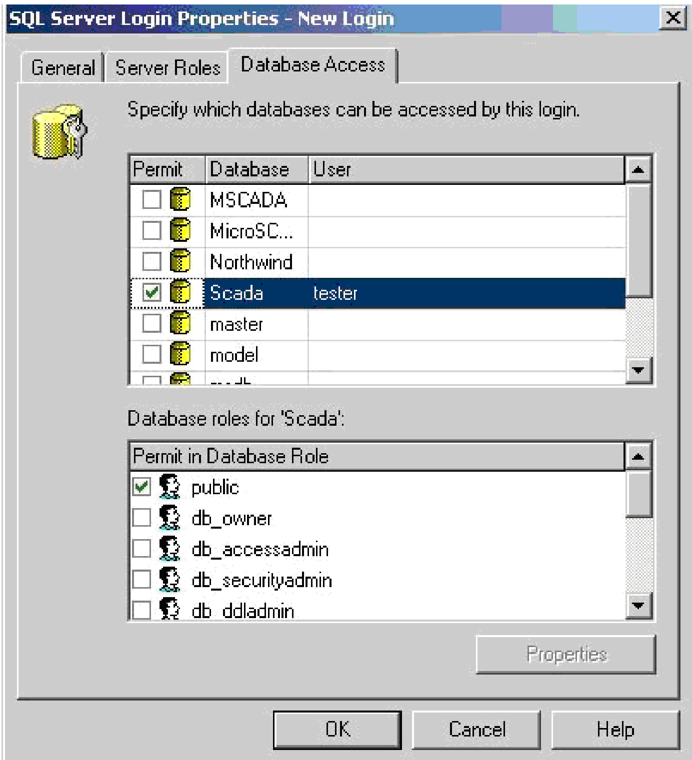


Figure 33: Defining database access

10. Click **Properties** to open the **Database User Properties** dialog, see Figure 34.
11. Define the needed permissions.

The user must have at least write/delete permissions.

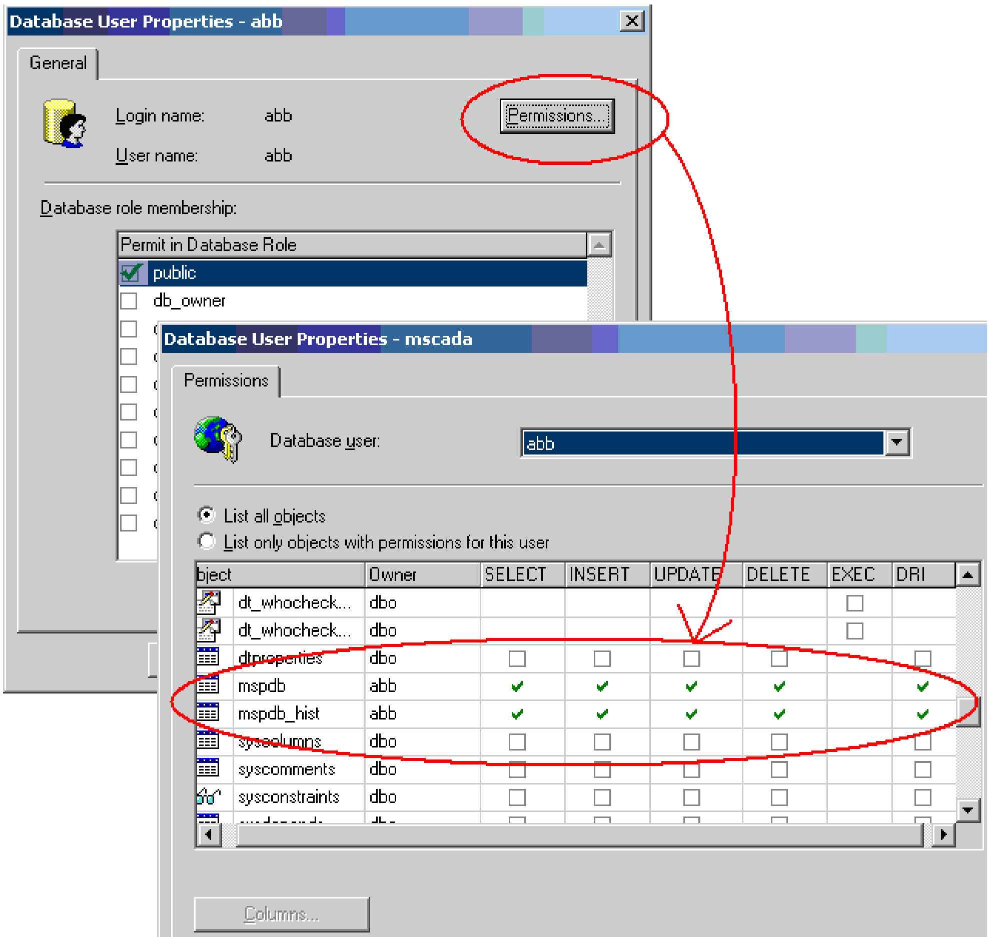


Figure 34: Defining access rights

12. Click **OK** to save the settings.

5.5.1 Enabling and disabling SQL logging

Select **Edit/SQL enabled/SQL disabled** to turn on or off the logging to SQL. The function is useful when working with the application.

5.6 Enabling communication between SQL database and SYS600

To enable communication between the SQL database and SYS600, define the ODBS/DDE connection settings.

1. Select **Start/Settings/Control Panel/Administrative tools/DataSources (ODBC)**.
2. Select **Add** in the System DSN tab, see Figure 35.

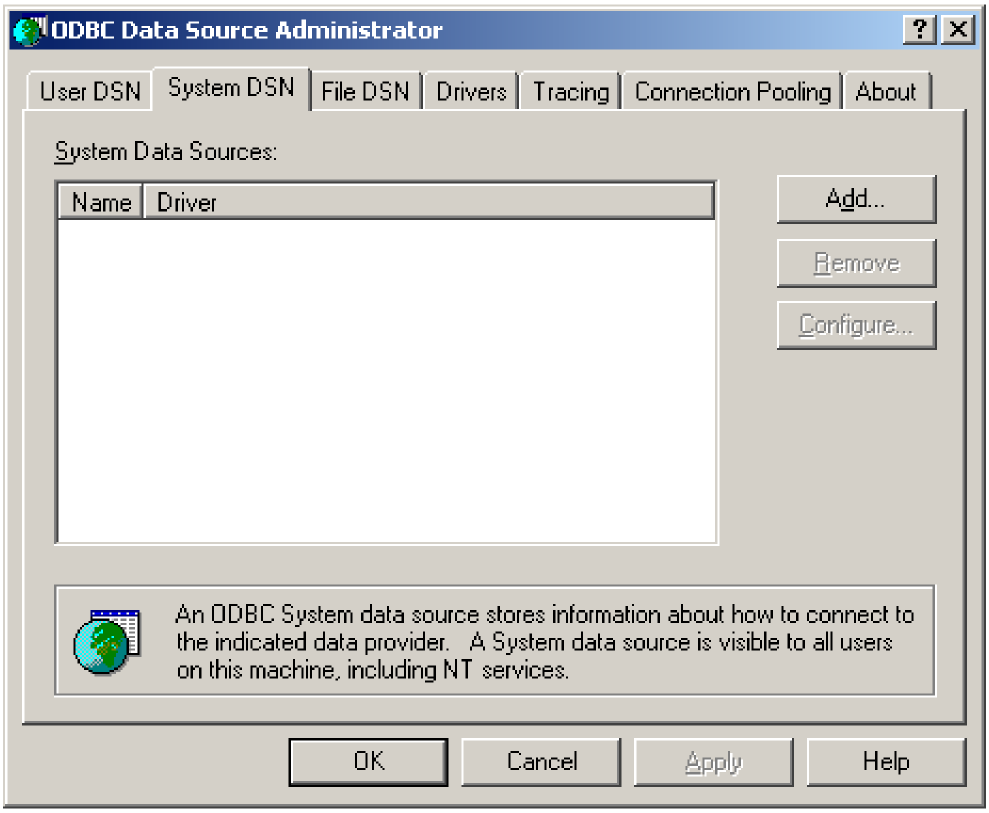


Figure 35: Adding SQL Server

3. Select the SQL Server in the **Create New Data Source** dialog, see Figure 36.

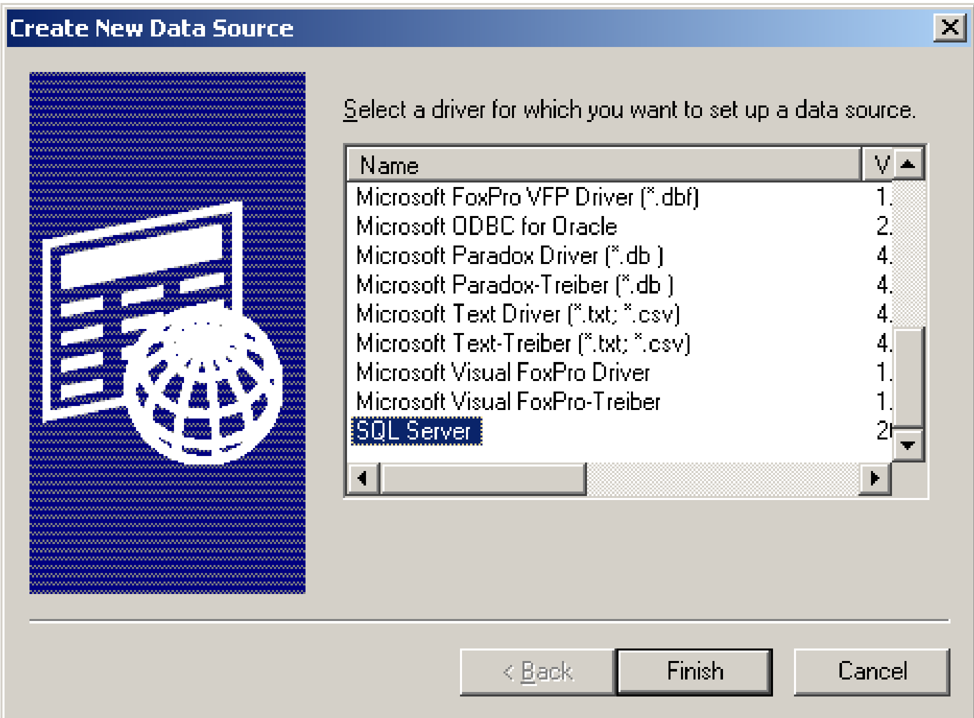


Figure 36: Selecting a database

4. Click **Finish** to display the SQL server configuration dialog.
5. Enter the name of the data source in the **Name** field. The name is used to connect to the database through SYS600, see Figure 37.

6. Enter the description of the data source in the **Description** field.
7. Select the server that SYS600 is connected to. If the server and the client are on the same computer, it is possible to use localhost as a SQL server.

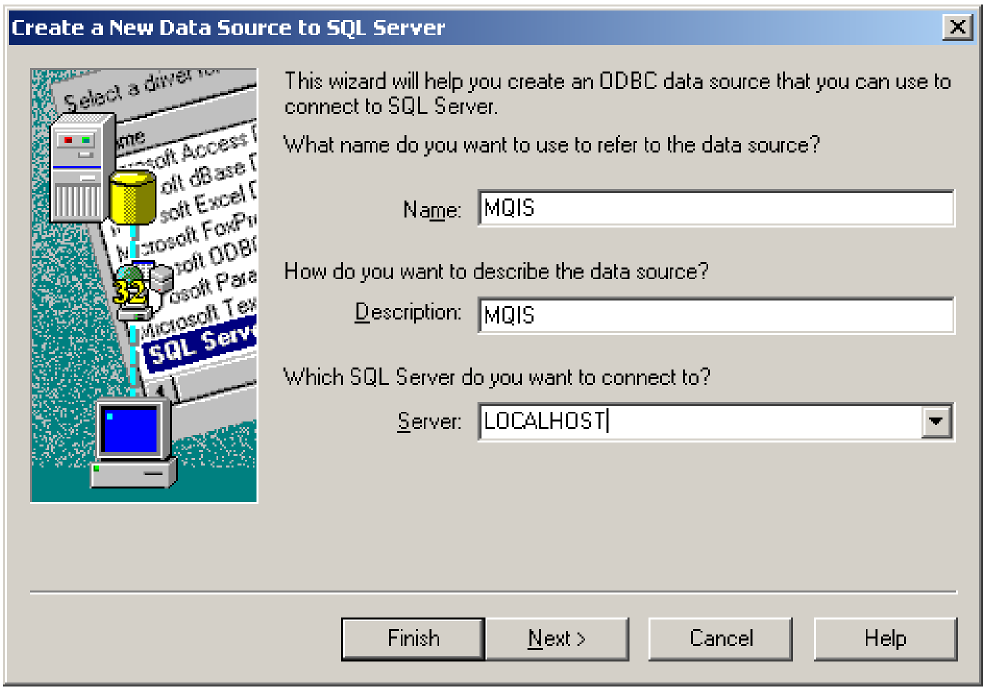


Figure 37: Creating a new data source

8. Click **Next**.
9. Select the SQL server authentication and type the user name and password in the **Login ID** and **Password** fields, see Figure 38. Use the same user name as created in Section 5.5.

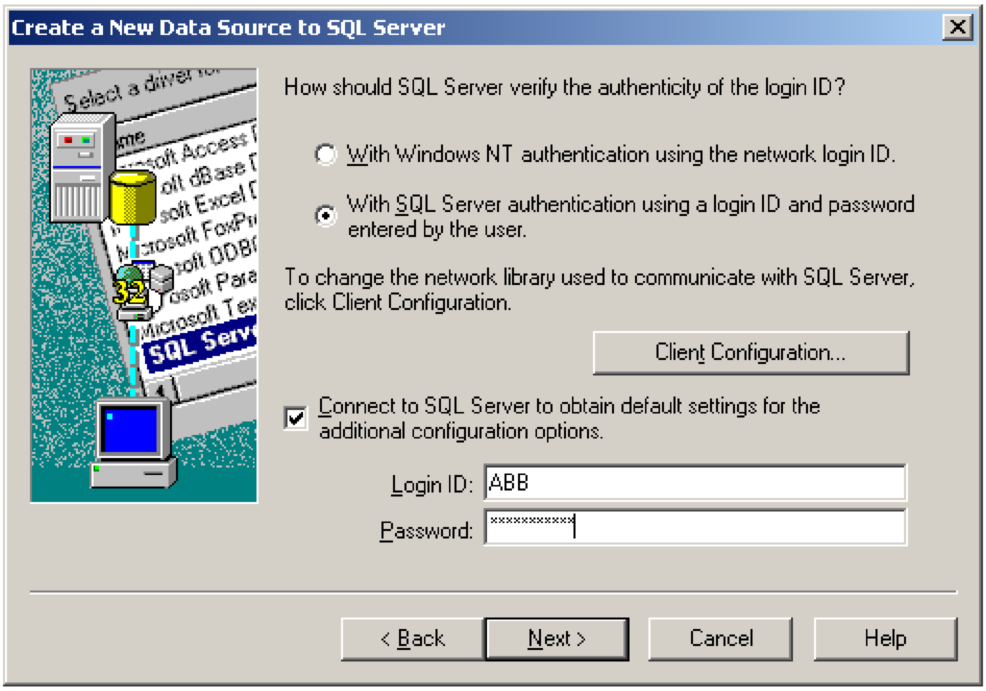


Figure 38: Defining authentication when logging on

10. Click **Next**.
11. Select a default database, for example SCADA, see Figure 39. Select the same database as created in Section 5.2.

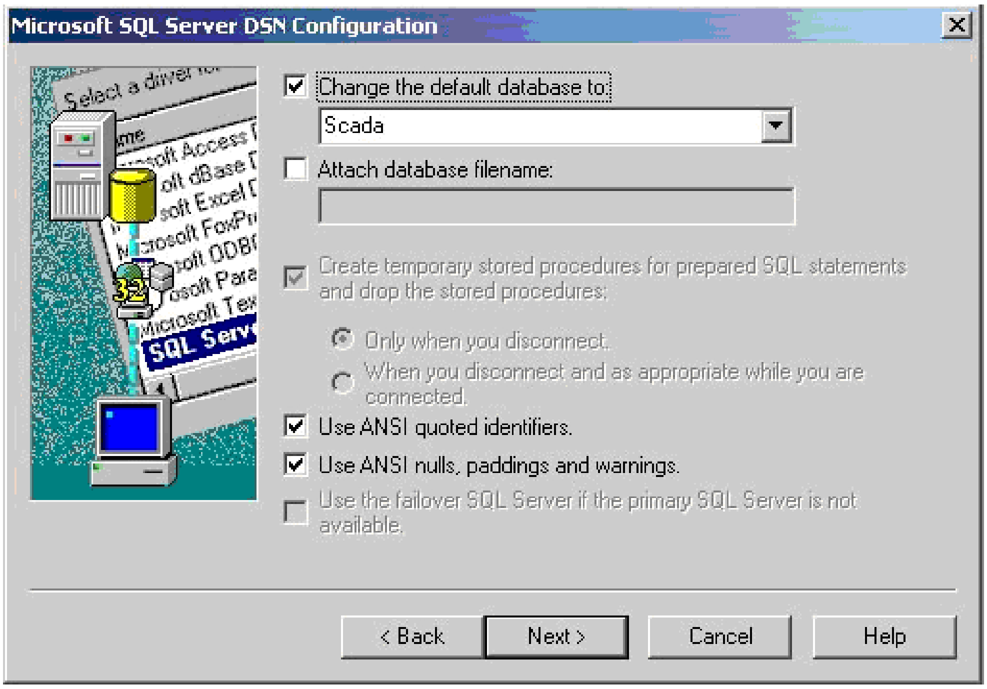


Figure 39: Selecting the default database

12. Click **Next**.
13. Select the **Perform translation for character data** check box, see Figure 40.

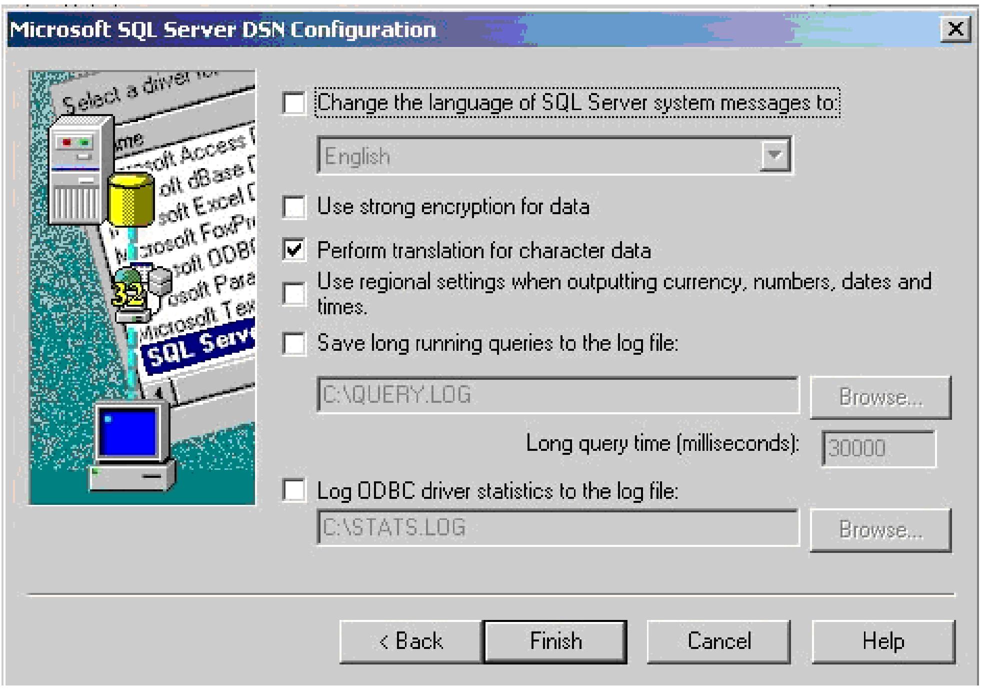


Figure 40: Selecting the Perform translation for character

14. Click **Finish**. After creating the database, test it as described in Section 5.7.

5.6.1 Editing command procedures

To enable SQL Exporter to log on to the SQL database, edit the SQL_FLUSH command procedure in SYS600, see Table 3.

Table 3: Command procedures to be edited

SQL_DATABASE	Rename the value with the same name as used to refer to the data source in Section 5.6.
SQL_USER	Rename the value with the same name as created in Section 5.5.
SQL_PASSWD	Rename the value with the same name as created in Section 5.5.

```
;*************** CONFIGURATION ***************

@SQL_DATABASE="MQIS"

@SQL_USER="Administrator"

@SQL_PASSWD="RSSlabra"

@CONSOLE_LOGGING="on" ;set "on" to log, "off" to disable

;*******************************************
```

Do not change the rest of the configuration.

Connect the RDBMS event channel (SQL_WRITE_DB_EVENT) to a Process Object with SQL Exporter in SYS600.

5.7 Testing user accounts and databases

Test the connection to the SQL Server with the SQL Query Analyzer.

1. Select **Start/Programs/Microsoft SQL Server/Query Analyzer** to start the SQL Query Analyzer.
2. Type the required information in the **Connect to SQL Server** dialog.
3. Try a simple query to see the content of the database, for example, see Figure 41.

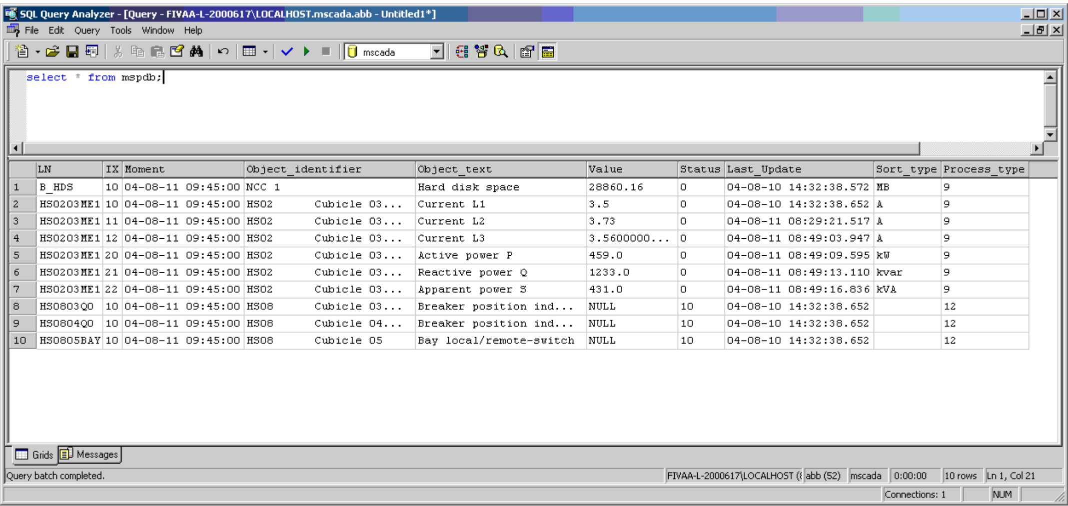


Figure 41: Testing the database connection

It is possible to type several queries at a time, for example:

```
Select * from mspdb

Order by mspdb.moment
```

```
Select * from mspdb_hist

Order by mspdb_hist.rt
```

5.8 Managing process objects

5.8.1 Adding process object

To add a process object to the SQL system:

1. Open SQL Exporter.
2. Select a process object from the **Database Objects** list.
3. Select whether to add the process object to the history or to the cyclic database or both.
4. Click **Add**.

The process object is displayed in the objects list.

To add several process objects to the SQL system at once:

1. Hold down the CTRL key and select all the needed process objects or drag the cursor over the process objects to select them.
2. Select whether to add the process objects to the history or to the cyclic database or both.
3. Click **Add**.

In general, measurements are added to the cyclic database while indications, alarms, and so on are added to the history database.

5.8.2 Removing process objects

To remove a process object from the SQL system:

1. Open SQL Exporter.
2. Select a process object from the objects list.
3. Click **Remove**.

The process object disappears from the objects list.

To remove several process objects from the SQL system at once:

1. Press CTRL and click all the process objects you would like to select or drag the cursor over the process objects to select them.
2. Click **Remove**.

Note that this action does not actually remove anything from the history database with the SQL Exporter. This only stops the process object's value from being registered in the history database when it is removed from the objects list. Clicking the **Remove** button does not remove any rows from the cyclic database. It only stops the object's value from being updated in the SQL database.

5.8.3 Selecting process objects

Excel can be used to create a list where the user has selected which process objects are to be connected to the SQL database. Create a list in Excel with four columns:

1. In the first column, add logical names (LN).
2. In the second column, add an index (IX).
3. In the third column, add an X if the process object should be connected to the history database.
4. In the fourth column, add an X if the process object should be connected to the measurement database.

The column headers do not necessarily need to be included.

The import function in the SQL Exporter starts reading process objects only from the second row, so if a process object is on the first row, it is ignored. Make sure that the columns are separated with commas or semicolons and not with any other character. Otherwise, the SQL Exporter cannot read the file correctly. For more information about saving the Excel list in a correct format, see Section 5.12.

An easy way to get a pre-made list of the process objects in the current application is to use the export function in SQL Exporter. For more information about exporting, see Section 5.9. After exporting, import the list to Excel and add the X:s. For more information about importing, see Section 5.10.

5.8.4 Using database filter options

The process objects in the SYS600 and SQL databases can be filtered according to their attributes LN, IX, PT, OA and UN. To use a filter, select the check box of the desired filter. It is possible to use the filters independently or several filters combined.

Note that both the **Database Objects** and **RDMS Objects** lists in the SQL Exporter are filtered using the same filters.

Filters:

- LN (Logical Name): Use an asterisk (*) as a wildcard character at the end of the filter string. The filter is case sensitive.
- IX (Index): Enter an index number. It is only possible to enter numbers, wildcard characters are not allowed.
- PT (Process Object Type): Select the desired type of objects list to be visible:
  - 3 – Single indication
  - 6 – Digital input
  - 9 – Analog input
  - 12 – Double indication
- OA (Object Address): Enter an object address number. It is possible to enter numbers only, wildcard characters are not allowed.
- UN (Unit Number): Enter a unit number. It is possible to enter numbers only, wildcard characters are not allowed.

5.9 Exporting list

To export the Database Objects list to a CSV file (comma delimited file), there are three alternatives:

- Click **Export**
- Select **File/Export to file**.
- Use the shortcut CTRL+E.

A list with the filtered process objects in the **Database Objects** list is written to a CSV file. The list in the file has 4 columns separated with commas. The column headers are:

- LN
- IX
- Send to history database
- Send to measurement database

Mark the process objects that are connected to the history or the measurement database respectively with an X in the corresponding column.

The CSV file can be imported to, for example, Excel. For more information about opening the CSV file in Excel, see Section 5.10.

5.10 Opening list in Excel

To import the list to Excel:

1. Open the text file created with SQL Exporter in Excel.
2. Text Import Wizard opens, see Figure 42.
3. Select the **Delimited** option in the **Original data type** field.
4. Click **Next**.

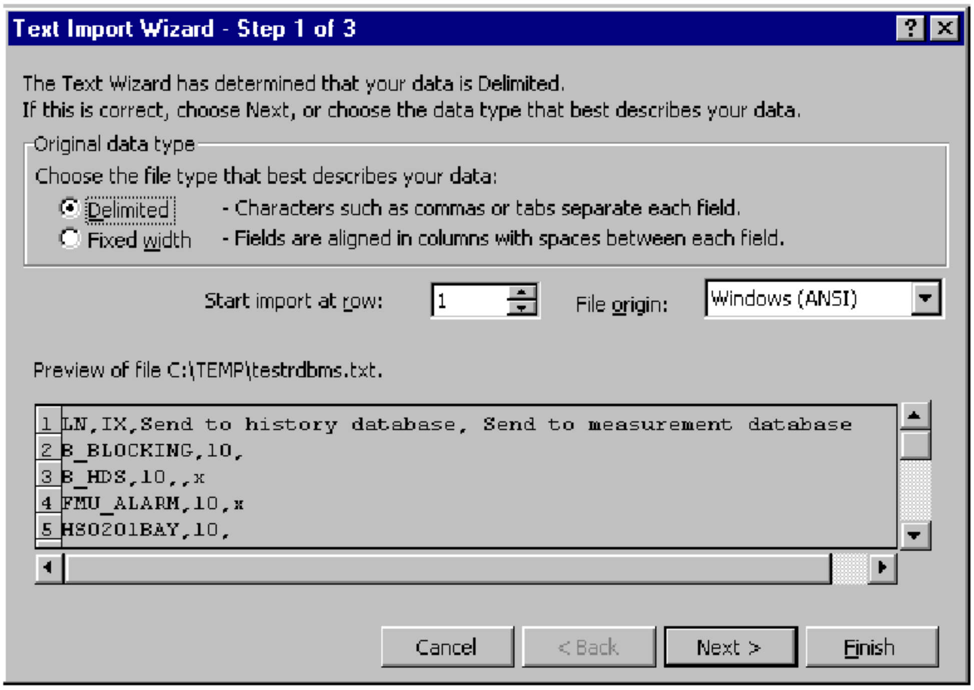


Figure 42: Excel Text Import Wizard, step 1

5. Select **Comma** in the Delimiters field, see Figure 43.

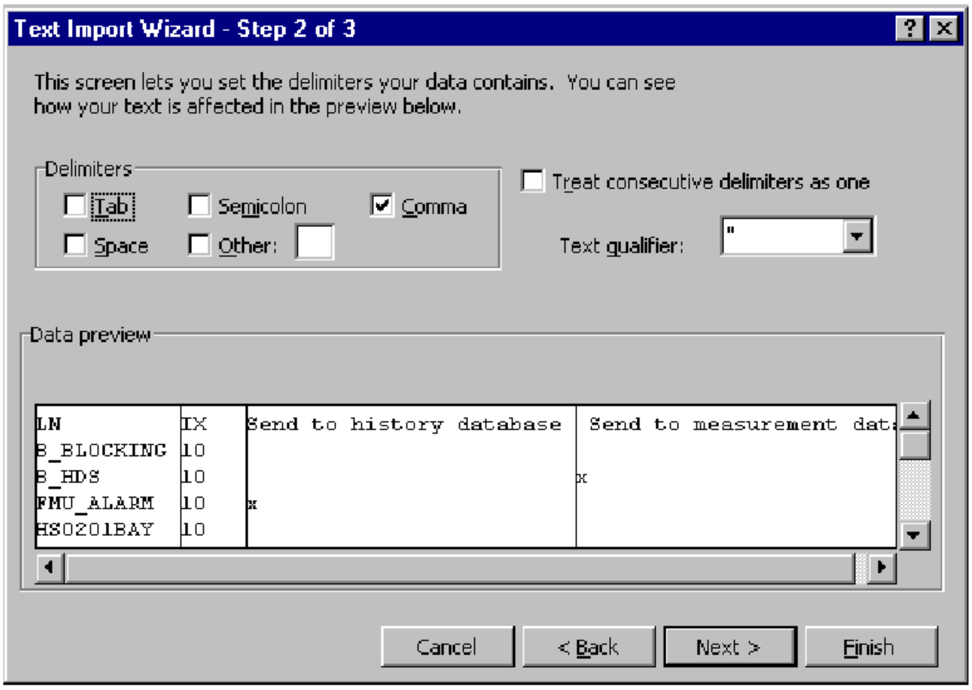


Figure 43: Excel Text Import Wizard, step 2

6. Preview the results of the import operation in **Data preview**.
7. Click **Finish** to import the list into an Excel worksheet.

5.11 Opening list in refreshable Excel range

Use this procedure to refresh the data in Excel whenever the original text file changes. If the data should be copied into Excel without maintaining a connection to the original file, open the text file in Excel. Once the list is imported, it can be saved as a normal Excel file.

To create a refreshable Excel:

1. Click the cell where the data from the text file should be placed.
2. Select **Data/Data/Import External Data/Import Data**.
3. Select the text file that should be imported as an external data range.
4. To specify how to divide the text into columns, follow the instructions in Text Import Wizard.
5. Click **Finish**.
6. Click **Properties** in the **Import Data** dialog to set formatting and layout options for how the external data range is imported into Microsoft Excel.
7. In the **Import Data** dialog, do one of the following:
   - To place the data to the existing worksheet, select **Existing worksheet**.
   - To place the data to a new worksheet, select **New worksheet**. Microsoft Excel adds a new worksheet to the workbook and automatically puts the external data range in the upper left corner of the new worksheet.
8. Click **OK**.

To refresh the external data range when the original text file changes:

1. Click **Refresh Data** on the External Data toolbar.
2. Select the text file in the **Import Text File** dialog.
3. Click **Import**.

To change the settings made in the Text Import Wizard:

1. Click **Edit Text Import** on the External Data toolbar.
2. Select the text file in the **Import Text File** dialog.
3. Click **Import**.
4. Make the desired changes in Text Import Wizard.

To import only part of the data in a text file as a refreshable data range, a query can be created to retrieve the data.

5.12 Importing list

The import function concerns only the process objects in the objects list. This function enables the user to import a list of process objects that are to be connected to the SQL database. The file should be the same file type as in the export function (a CSV file).

To import objects to the objects list from a CSV file (or another comma delimited text file), there are three alternatives:

- Click **Import**.
- Select **Import to file** from the **File** menu.
- Use the shortcut CTRL+I.

In the import list, the column headers are:

- LN
- IX
- Send to history database
- Send to measurement database

The list contains a process object in the LN column and the index in the IX column. Mark the process objects that will be connected to the SQL database respectively with an X in the corresponding column, see Figure 46.

Currently, it is possible to add process objects only by using the import function. If an X is removed and the list is imported, the process object is not removed from the objects list.

The CSV file can be generated, for example, with Excel or Notepad. Save the file as a CSV file. The most convenient way to create a list to be imported to SQL Exporter is to use Excel on an existing process object list.

To import a list by using Excel:

1. Select **File/Save As**.
2. Save the file as a CSV file.
3. Answer **Yes** if asked whether to keep the workbook in this format, see Figure 44.

Figure 44: Save the workbook in CSV format

4. The file is saved as a text file with semicolons separating the columns and each row ending with a carriage return.

Make sure that the columns are separated with commas or semicolons, and not with any other character. Check this by opening the file with, for example Notepad, see Figure 45. Otherwise, SQL Exporter cannot read the text file correctly.

If any other character than semicolon or comma is used as a delimiter, replace the character with a semicolon or comma. To replace delimiters in Notepad, select **Edit/Replace**.

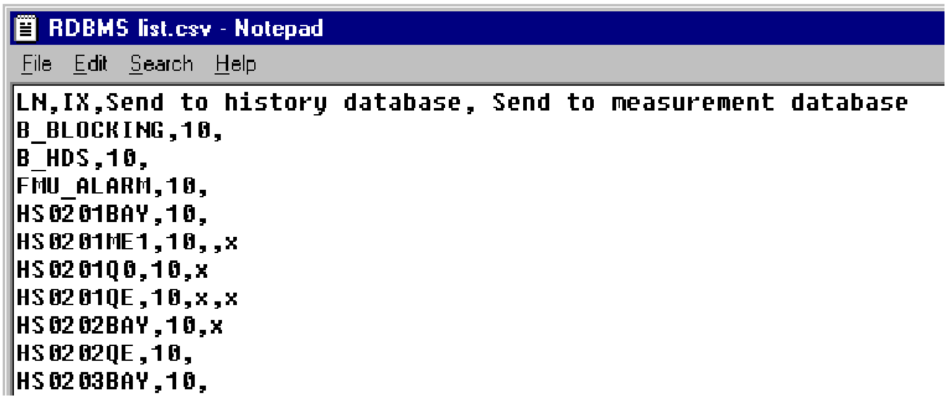


Figure 45: Comma delimited process object list in Notepad

Example:

In Figure 46, the process object FMU_ALARM is added to the history database and B_HDS is added to the measurement database.

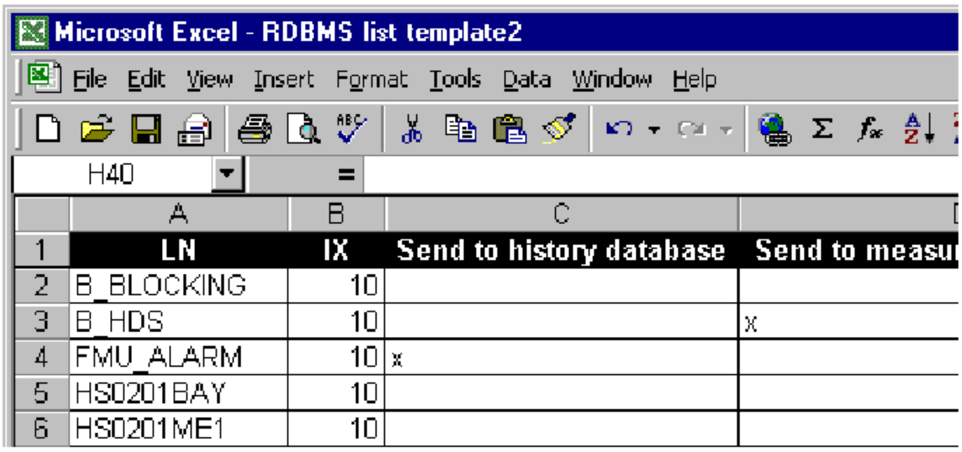


Figure 46: Creating a signal list in Excel for importing

All process objects available in the current application do not necessarily have to be included in the imported list. Including only the desired objects to the SQL system is enough.

An error log is created:

- If the process object list contains a process object that does not exist in the user's current application.
- If the user tries to add a process object that already is connected to an event channel to the history database.

The error log is located in the same folder as the file that is being imported (filename: <imported file>_ERROR.LOG).

5.13 Cleaning objects

When building a SYS600 application, process objects are added and removed. Sometimes the objects are left in the SQL_OBJECTS command procedure.

The cleaning function removes non-existing process objects from the command procedure (SQL_OBJECTS:C). Select **Edit/Clean RDBMS Objects** to clean the command procedure.

Section 6 Standard functions of Power Process Library

6.1 Preliminary work

Before the configuration of the standard functions can start, some preparatory configurations are required (that is, configurations that are required only once). The usage of the object identification (OI) attribute has to be settled, the naming convention of process object has to be decided and the translation of signal texts into the local language has to be performed.

Some of the symbols in this section have different presentations, for example the vertical and horizontal presentation. For more information on the different presentations, see the SYS600 Process Display Design manual.

6.2 Procedures for use with standard functions

Procedure U_OTHER_OBJECTS

This procedure can be used to define other process objects that are needed by the standard function. This procedure can be used for optimizing the performance of the control dialog. When these logical names are defined, the control dialog does not need to make query to process database to find the objects, but it can directly refer to the objects with their given logical names. For STATION_LN and BAY_LN, value NONE can be used to define that the corresponding object does not prohibit operations in the control dialog. When STATION_LN or BAY_LN is "", the corresponding object state is checked and it is searched from process database by object identification (OI).

Default contents of the procedure is following:

```
@U_OTHER_OBJECTS = LIST(-
STATION_LN = "",-
BAY_LN = "",-
AR_LN = "",-
TRIP_LN = "")
#RETURN %U_OTHER_OBJECTS
```

STATION_LN defines the logical name of station standard function.

Table 4: Attributes for Procedure U_OTHER_OBJECTS

Attribute	Description
STATION_LN	TEXT, logical name of station standard function.
BAY_LN	TEXT, logical name of bay standard function.
AR_LN	TEXT, logical name of auto-reclosure standard function.
TRIP_LN	TEXT, logical name of trip tag standard function.

Procedure U_INTERLOCKING

This procedure can be used for defining custom interlocking conditions for the control dialog. The procedure is executed at control dialog startup. The procedure must return a LIST with the following attributes:

Table 5: Attributes for Procedure U_INTERLOCKING

Attribute	Description
OPEN_INTERLOCKED	BOOLEAN, open interlocking state. If TRUE, the **Open** button is disabled in the control dialog.
CLOSE_INTERLOCKED	BOOLEAN, close interlocking state. If TRUE, the **Close** button is disabled in the control dialog.
INTERLOCKING_CAUSE	TEXT, reason of the interlocking to be displayed in control dialog info field.

Default contents of the procedure is:

```
#RETURN LIST(-
OPEN_INTERLOCKED=%OPEN_INTERLOCKED, -
CLOSE_INTERLOCKED=%CLOSE_INTERLOCKED, -
INTERLOCKING_CAUSE=%INTERLOCKING_CAUSE)
```

The procedure is executed only at dialog startup. If the prerequisites of the interlocking state change while the dialog is open, the changes are not dynamically updated in the dialog.

Procedure U_LR_VALUE_CONVENTION

This procedure can be used for defining the bay L/R switch value convention. The value of the Bay L/R indication can be mapped to control locations. One value can be mapped to more than one location. The following keywords are used in the mapping.

- OFF, control is not allowed from any place
- LOCAL, control is in the front panel of the IED
- STATION, control is on the station level
- NCC, control is in the Network Control Center
- All, control allowed from all operator places

Switch device and Tap changer dialogs can read the convention from the bay.

Table 6: Attributes for Procedure U_LR_VALUE_CONVENTION

Attribute	Description
THIS_SYSTEM_NAME	TEXT, optional system name. If the name of the system is not given, the system location is read from the definition done in the Application Settings (SCS/NCC).
SIGNALS	VECTOR, each element of the vector is a LIST with attributes described below.
VALUE	INTEGER, the value to which the control locations are mapped.
CONTROL_PLACES	VECTOR, the control locations for the value.
CONTROL_POSITION	TEXT, text that is shown in the control dialog.
CONTROL_VALUE	INTEGER, optional. If the L/R switch is remotely controllable this value is used to control to this position.
CONTROL_LN	TEXT, optional. The logical name of the process object where the CONTROL_VALUE is written.
CONTROL_INDEX	INTEGER, optional. The index of the process object where the CONTROL_VALUE is written.

Example of the usage of the U_LR_VALUE_CONVENTION:

```
#RETURN LIST(-
THIS_SYSTEM_NAME = "",-
   SIGNALS = VECTOR(-
```

```
      LIST(VALUE = 0, CONTROL_PLACES = VECTOR("OFF")),-
      LIST(VALUE = 1, CONTROL_PLACES = VECTOR("LOCAL")),-
      LIST(VALUE = 3, CONTROL_PLACES = VECTOR("STATION")),-
      LIST(VALUE = 5, CONTROL_PLACES = VECTOR("LOCAL", "STATION")),-
      LIST(VALUE = 6, CONTROL_PLACES = VECTOR("ALL")),-
      LIST(VALUE = 7, CONTROL_PLACES = VECTOR("STATION", "NCC")),-
      LIST(CONTROL_POSITION = translation("Remote=0"), CONTROL_VALUE = 0,
CONTROL_INDEX = 23),-
      LIST(CONTROL_POSITION = "Station=1", CONTROL_VALUE = 1,
CONTROL_INDEX = 23, CONTROL_LN = "")-
    ))
```

When U_LR_VALUE_CONVENTION is used, the Bay control dialog reads the current L/R switch position directly from the SX attribute of the indication process object. The Event Handling object connected to the indication should thus follow the convention defined by this program.
If the CONTROL_POSITION is not defined custom convention is indication only.

For Relion 615 and 670 there is a default (automatic) custom convention defined in the text file BGU_U_LR_VALUE_CONVENTION.txt. This is applicable when the IEC 61850 protocol is in use. In this file the operator place handling is decided based on the data object name found in the IN attribute.

6.3 Communication support

The following protocols are supported in Power Process Library:

- SPA
- ANSI
- LON
- RP-570
- RP-570 with FTABs
- IEC 60870-5-101/104
- IEC 60870-5-103
- IEC 61850-8
- DNP 3.0
- MODBUS RTU/ASCII/TCP
- PROCOL

6.4 Station

6.4.1 Standard function installation

This section describes the installation of station standard function from the Power Process Library. The standard function for the station is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Process/Station). The standard function can be configured by using the Object Navigator tool.

6.4.1.1 Symbol installation

The Power Process symbol for station standard function is installed by using Display Builder functions:

1. Object Browser
2. Palette

For more information, see the SYS600 Process Display Design manual.

Table 7: Power Process symbol for station standard function

File name	Symbol
Station BI.sd	

6.4.2 Standard function configuration

6.4.2.1 Configuration with a tool

The station standard function has the following attributes to be configured with the Standard Function Tool:

Table 8: Configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
P_OBJECT_LN	Logical name of database process objects	-
STATION_TYPE	The type of control device & protocol	LON
CMD_PARAMETER	The type of control command for	N/A
OUTPUT_STATUS	Process objects for output status with DNP 3.0 protocol	N/A
AUTHORIZATION_GROUP	The name of the authorization group	MV_CONTROL
STATION_LR_IN_USE	Station local/remote switch used for control authority checking	Yes
IND_DOUBLE_BINARY	Indication type double binary/ single binary	Double binary
LR_REMOTELY_CONTROLLABLE	Station local/remote switch remotely controllable/ manually operated	Yes
EVENT_RECORDING	Process objects for event recording with RP-570	No process objects
OPC_ITEM_PREFIX	Specifies the OPC Item Prefix used as an IEC 61850 instance information	"IEC61850 Subnetwork.IED1.LD1"
OPC_LN_INSTANCES	Specifies the OPC Logical Node names used as IEC 61850 instance information	VECTOR("LLN0")

For more detailed description on the configurable attributes, see Section 6.15.

6.4.2.2 Configuration with the Process Object Tool

Depending on the configuration of the station standard function, the Process Object Tool creates a certain set of the process objects into the database. Process objects that will be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

The Process Object Tool creates process objects based on the following list:

Table 9: *Process objects created by the Process Object Tool*

Index	Explanation	Purpose
10	Station local/ remote switch indication	The station local/remote switch is used to determine whether the control of station objects is allowed from the panel (local), from the substation control system, from the remote control system or whether station authority checking is out of use.
11	Station local/ remote switch to Local command or Station/Remote command	The station local/remote switch to Local command object is an output object which is designed to set the station control to local (panel) when the attribute IND_DOUBLE_BINARY is checked. If IND_DOUBLE_BINARY is not checked, output object is used for setting the switch either to Station or Remote.
12	Station local/ remote switch to Station command	The station local/remote switch to Station command object is an output object, which is designed to set the station control to substation control system (station).
13	Station local/ remote switch to Remote command	The station local/remote switch to Remote command object is an output object, which is designed to set the station control to network control system (remote).
14	Station local/ remote switch to Out of use command	The station local/remote switch to Out of use command object is an output object, which is designed to set the station authority checking out of use.
15	Station blockings (optional)	The indication for station blockings is an internal object, which is meant for generating events and printouts of the station blockings.
16	Station selected on monitor (optional)	The indication for station selected on monitor is an internal object and it is used to inform other users that the station dialog has been opened by other user(s).
110	Station local/ remote-switch indication	Created only for ANSI and RTU type stations. For ANSI it is used for converting the indication from analog value to a double binary value. For RTU, it is used as an event recording object.
111-114	Indication for command termination (optional)	Indicates with IEC 60870-5-101/104 and -103 whether the issued command was successful or not.

6.4.2.3 Example of a station configuration

Table 10: *Example of a station standard function configuration*

Attribute	Value
STATION_NAME	Eastwick
P_OBJECT_LN	EST
STATION_TYPE	LON
AUTHORIZATION_GROUP	MV_CONTROL
STATION_LR_IN_USE	Yes
IND_DOUBLE_BINARY	Yes
LR_REMOTELY_CONTROLLABLE	No
EVENT_RECORDING	N/A

6.4.3 Application engineering information

6.4.3.1 Structure of the station standard function

This section describes the structure of the station standard function. All subdrawing files form pictures, help and other text files, as well as database objects are included. The station is a part of the standard functions of Power Process Library and has a directory.

6.4.3.2 Files

The table below lists all of the station standard function related files and their functionality.

Table 11: Station standard function related files

File	Functionality	
Station BI.sd Station.sd	Power Process symbol for station standard function	/PROG/GRAPHICSENGINE/PALETTE/04-SA_Indication
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

6.4.3.3 Language text file

The following Text Translation Tool compatible text files are used by the station standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 12: Text Translation Tool compatible text files

File	Functionality
SAI_STA2.TXT	Text file for the database creation of the standard function

6.4.3.4 Help text file

The path to the station help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 13: Station help text file

File	Functionality
SAI_STA2.HLP	Standard function installation help file

6.4.3.5 Configuration files

The following configuration files are used by the station standard function. The path is / SA_LIB/BASE/BBONE/INST.

Table 14: Configuration files used by the station standard function

File	Functionality
SAI_STA2.DAT	Contains the configuration data for station when it is created, modified or deleted with the configuration tools.
SAI_STA2.POT	Contains the process object definitions for the Process Object Tool.
SAI_DAU.TXT	Contains the list of data acquisition units.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for the installation tool.

6.4.3.6 Other text files

The following text file is used by the station. The path is /SA_LIB/BASE/BBONE/USE.

Table 15: Another text file used by the station

File	Functionality
BGU_AUTH.TXT	Contains the SCIL code for the station authority check. The code is executed by the dialog for the station.

6.4.3.7 Process objects

The following process objects are created depending on the configuration of the station.

Table 16: Process objects created by manually operated station local/remote switch

Index	Obj. type	Process object	Remarks	Group identifier
10	DB/BI	Station local/remote switch ind.	-	FPASTAILRS
15	AI	Blockings	Internal	FPASTAXBLK
16	AI	Station selected on monitor	Internal	FPASTAXMEV
110	EVREC	Station local/remote switch ind.	RP-570 only	FPASTAELRS
110	AI	Station local/remote switch ind.	ANSI only	FPASTAELRS

Table 17: Process objects created by remotely controllable station local/remote

Index	Obj. type	Process object	Remarks	Group identifier
10	DB/BI	Station local/remote switch ind.	-	FPASTAILRS
11	BO	L/R-sw. to Local -command or Station Remote command	-	FPASTACLRS
12	BO	L/R-sw. to Station -command	-	FPASTACLRS
13	BO	L/R-sw. to Remote -command	-	FPASTACLRS
14	BO	L/R-sw. to Out of use -command	-	FPASTACLRS
15	AI	Blockings	Internal Optional	FPASTAXBLK
16	AI	Station selected on monitor	Internal Optional	FPASTAXMEV
110	EVREC	Station local/remote switch ind.	RP-570 only Optional	FPASTAELRS
110	AI	Station local/remote switch ind.	ANSI only	FPASTAELRS
111-114	TERM	Command termination indication for output signals	IEC 60870-5-101/104 and -103 only Optional	FPASTATOBC

6.4.3.8 Scale objects

At the first installation, the station creates the scale 1_1 (linear 1:1 scale). For SPACOM/SRIO (ANSI) station type, the installation will also create the scale FPU_1_1000 (linear 1:1000 scale).

6.4.3.9 Command procedures

At the first installation, the station creates the command procedure BGU_UPDATE, which updates process objects when they are update-deblocked. This feature is similar to the

SYS600 Blocking Display function. It is also used by the Update process data function in the control dialog.

For SPACOM/SRIO (ANSI) station type, the installation creates the command procedure BGU_AI2DB, which updates the station level local/remote switch indication double binary from an analog input.

6.4.3.10 Event channels

For SPACOM/SRIO (ANSI) station types, the installation creates the event channel BGU_AI2DB, which is set to activate the command procedure BGU_AI2DB.

6.5 Bay

6.5.1 Standard function installation

This section describes the installation of the bay standard function in Power Process Library. The standard function for the bay is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Process/Bay). This standard function is configured by using configuration tools, such as the Object Navigator.

6.5.1.1 Symbol installation

The Power Process symbol for the bay standard function is installed by using the Display Builder functions:

1. Object Browser
2. Palette

For more information, see the SYS600 Process Display Design manual.

Table 18: Bay Switch Indication

File name	Symbol
Bay BI.sd	

6.5.2 Standard function configuration

6.5.2.1 Configuration with tools

The bay standard function has the following attributes to be configured with the Object Navigator:

Table 19: Configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
P_OBJECT_LN	Logical name of database process objects	-
Table continues on next page		

Attribute	Meaning	Default
STATION_TYPE	The type of control device & protocol	LON
CMD_PARAMETER	Parameter for control method for IEC 60870-5-101/104 and DNP 3.0 protocol	N/A
OUTPUT_STATUS	Process objects for output status with DNP 3.0 protocol	N/A
AUTHORIZATION_GROUP	The name of the authorization group	MV_CONTROL
BAY_LR_IN_USE	Physical L/R-switch exists/not available	Bay L/R-switch exists
BAY_LR_POLARITY	Polarity of the bay L/R switch	L1
IND_DOUBLE_BINARY	Indication type double binary/ single binary	Double binary
LR_REMOTELY_CONTROLLABLE	Bay L/R-switch remotely controllable/ manually operated	Manually operated
EVENT_RECORDING	Process objects for event recording with RP-570	No process objects
OPC_ITEM_PREFIX	Specifies the OPC Item Prefix used as IEC 61850 instance information	IEC61850 Subnetwork.IED1
OPC_LN_INSTANCES	Specifies the OPC Logical Node names used as IEC 61850 instance information	VECTOR("LLN0")
OPERATOR_PLACE_HANDLING	Specifies the baywise operator place (SCS/NCC) authority check	None
MULTILEVEL_CONTROL	Allows the multilevel operator place control	No multilevel control

For more detailed description on the configurable attributes, see Section 6.15.

6.5.2.2 Configuration of process objects

Depending on the configuration of the bay standard function, the tools create a certain set of process objects in the database. Those process objects that will be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

The tools create process objects based on the following list:

Table 20: Process objects created for the bay

Index	Explanation	Purpose
10	Bay local/remote switch indication	The bay (disabled)/local/remote switch is used to determine whether the control to the bay objects is completely inhibited, or it is allowed locally or remotely.
11	Bay local/remote switch to Disabled command, or bay local/remote command	The bay local/remote switch to Disabled command object is an output object which is designed to set the bay control to disabled (no control locally or remotely). If IND_DOUBLE_BINARY is not checked and STATION_TYPE is IEC 61850-8, this output object is used for setting bay L/R switch either to local or remote.
12	Bay local/remote switch to Local command	The bay local/remote switch to Local command object is an output object, which is designed to set the bay control to the local (panel).

Table continues on next page

Index	Explanation	Purpose
13	Bay local/remote switch to Remote command	The bay local/remote switch to Remote command object is an output object, which is designed to set the bay control to substation or network control system (remote).
14	Bay local/remote switch to Reserved command	The bay local/remote switch to Reserved command object is an output object, which is designed to set the bay control to reserved. The reserved position can be used for special purposes.
15	External interlocking ON/OFF (HW) indication (optional)	The external interlocking in use/out of use (HW) ind. is meant as an ON/OFF type of indication for the hardware switch of the interlocking function.
16	External interlocking ON/OFF (SW) - indication (optional)	The external interlocking in use/out of use (SW) ind. is meant as an ON/OFF type of indication for the software switch of the interlocking function.
17	External interlocking ON/OFF -command (optional)	The external interlocking in use/out of use command is an output object, which can be used to remotely enable/disable interlocking function of the field device.
18	Internal interlocking ON/OFF -indication (optional)	The internal interlocking in use/out of use indication is meant as an ON/OFF type of indication of the internal interlocking functions. By disabling internal interlocking, there will not be any interlocking checking in the control dialogs during operation.
19	Internal interlocking ON/OFF -command (optional)	The internal interlocking in use/out of use command is an object, which is defined to internally control the state of the internal interlocking in use/out of use indication.
20	Bay blockings (optional)	The bay blockings indication is an internal object, which is meant for generating events and printouts of the bay blockings.
21	Bay selected on monitor (optional)	The bay selected on monitor indication is an internal object and it is used to inform other users that the bay dialog has been opened by another user(s).
22	Bay operator place switch (optional)	The bay operator place switch defines the authorized control location (SCS/NCC/Both) for devices belonging to the bay. If the IU attribute of the process object is set to 0 (not in use), Bay operator place is not checked in control dialogs.
23	Bay operator place switch to not in use cmd	Used for setting Bay operator place switch to not in use or switching the operator place between Station/NCC (IEC 61850)
24	Bay operator place switch to station cmd	Used for setting Bay operator place switch to Station (SCS).
25	Bay operator place switch to remote cmd	Used for setting Bay operator place switch to Remote (NCC).
26	Bay operator place switch to station/ remote cmd	Used for setting Bay operator place switch to Station and Remote.
30	Multilevel control	Used for enabling multilevel control (both Station and NCC authorized to control) for a bay.
110	Bay local/remote switch indication	Created only for ANSI and RTU type stations. For ANSI it is used for converting the indication from analog value to double binary value. For RTU it is used as an event recording object.
Table continues on next page		

Index	Explanation	Purpose
111-114, 117	Indication for command termination (optional)	Indicates with IEC 60870-5-101/104 and -103 whether the issued command was successful or not. Output status object with DNP 3.0 type of stations.
115	External interlocking ON/OFF (HW) (optional)	Event recording object (for accurate time stamp) with RTU 2xx/ RP-570 type of stations.
116	External interlocking ON/OFF (SW) (optional)	Event recording object (for accurate time stamp) with RTU 2xx/ RP-570 type of stations.

6.5.2.3 Example of a bay configuration

Table 21: Example configuration of the bay standard function

Attribute	Value
STATION_NAME	Eastwick
BAY_NAME	Outgoing HA2
P_OBJECT_LN	ESTHA02BAY
STATION_TYPE	LON
CMD_PARAMETER	N/A
OUTPUT_STATUS	N/A
AUTHORIZATION_GROUP	MV_CONTROL
BAY_LR_IN_USE	Yes
BAY_LR_POLARITY	L1
IND_DOUBLE_BINARY	Yes
LR_REMOTELY_CONTROLLABLE	No
EVENT_RECORDING	N/A

6.5.3 Application engineering information

6.5.3.1 Structure of the bay standard function

This section describes the structure of the bay standard function. All subdrawing files, form pictures, help and other text files, as well as database objects are included. The bay is a part of the standard functions of the Power Process Library and has a directory.

6.5.3.2 Files

The table below lists all bay standard function related files and their functionality.

Table 22: Bay standard function related files

File	Functionality	Path
Bay BI.sd Bay.sd	Power Process symbol for the bay standard function	/PROG/GRAPHICSENGINE/PALETTE/04-SA_Indication
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

6.5.3.3 Language text file

The following Text Translation Tool compatible text files are used by the bay standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 23: Text Translation Tool compatible text files

File	Functionality
SAI_BAY2.TXT	Text file for the database creation of the standard function

6.5.3.4 Help text file

The path to the bay help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 24: Bay help text file

File	Functionality
SAI_BAY2.HLP	Standard function installation help file

6.5.3.5 Configuration files

The following configuration files are used by the bay standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 25: Configuration files used by the bay standard function

File	Functionality
SAI_BAY2.DAT	Contains the configuration data for the bay when it is created, modified or deleted by the configuration tools.
SAI_BAY2.POT	Contains the process object definitions for the Object Navigator.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for the installation tool.

6.5.3.6 Other text files

The following text file is used by the bay. The path is /SA_LIB/BASE/BBONE/USE.

Table 26: Another text file used by the bay

File	Functionality
BGU_AUTH.TXT	Contains the SCIL code for the station authority check. The code is executed by the dialog of the bay.

6.5.3.7 Command procedures

At the first installation, the bay creates the command procedure BGU_UPDATE, which updates process objects when they are update-deblocked. This feature is similar to the SYS600 Blocking Display function. It is also used by the **Update process data** dialog.

Table 27: Command procedures used by bay

File	Functionality
BGU_CONTROL	Performs the actual control operations.

6.5.3.8 Process objects

The following process objects are created depending on the configuration of the bay.

Table 28: Process objects created by manually operated bay local/remote switch

Index	Obj. type	Process object	Remarks	Group identifier
10	DB/BI	Bay local/remote switch ind.	-	FPABAYILRS
15	BI	Interl. in use/out of use (SW)	Hardware sw. (MFU) Optional	FPABAYIINH
16	BI	Interl. in use/out of use (HW)	Software sw. (MFU) Optional	FPABAYIINH
17	BO	Interl. in use/out of use -cmd	Command to MFU Optional	FPABAYCINH
18	BI	Interl. in use/out of use	Internal Optional	FPABAYIINH
19	BO	Interl. in use/out of use -cmd	Internal Optional	FPABAYCINH
20	AI	Blockings	Internal Optional	FPABAYXBLK
21	AI	Bay selected on monitor	Internal Optional	FPABAYXMEV
110	EVREC	Bay local/remote switch ind.	RP-570 only Optional	FPABAYELRS
110	AI	Bay local/remote switch ind.	ANSI only	FPABAYELRS
115	EVREC	Interl. in use/out of use (SW)	RP-570 only Optional	FPABAYEINH
116	EVREC	Interl. in use/out of use (HW)	RP-570 only Optional	FPABAYEINH
117,	TERM	Command termination indication for output signals	IEC 60870-5-101/104 and -103 only Optional	FPASTATINH

Table 29: Process objects created by remotely controllable bay local/remote switch

Index	Obj. type	Process object	Remarks	Group identifier
10	DB/BI	Bay local/remote switch ind.	-	FPABAYILRS
11	BO	L/R-sw. to Local -command or Local Remote command	Local Remote command if station type is IEC 61850-8	FPABAYCLRS
12	BO	L/R-sw. to Local -command	-	FPABAYCLRS
13	BO	L/R-sw. to Remote -command	-	FPABAYCLRS
14	BO	L/R-sw. to Reserved -command	Only if ind. type DB	FPABAYCLRS
15	BI	Interl. in use/out of use (HW)	Hardware sw. (MFU) Optional	FPABAYIINH
16	BI	Interl. in use/out of use (SW)	Software sw. (MFU) Optional	FPABAYIINH
17	BO	Interl. in use/out of use -cmd	Command to MFU Optional	FPABAYCINH
18	BI	Interl. in use/out of use -ind.	Internal Optional	FPABAYIINH
19	BO	Interl. in use/out of use -cmd	Internal Optional	FPABAYCINH

Table continues on next page

Index	Obj. type	Process object	Remarks	Group identifier
20	AI	Blockings	Internal Optional	FPABAYXBLK
21	AI	Bay selected on monitor	Internal Optional	FPABAYXMEV
22	DB	Bay operator place switch	Optional Process objects for Bay operator place switch are created also for remotely controllable bay	FPABAYIOPS
23	BO	Bay operator place switch to not in use cmd or Bay Operator place-switch command	Optional	FPABAYCOPS
24	BO	Bay operator place switch to station cmd	Optional	FPABAYCOPS
25	BO	Bay operator place switch to remote cmd	Optional	FPABAYCOPS
26	BO	Bay operator place switch to station/remote cmd	Optional	FPABAYCOPS
30	BI	Multilevel control	Optional	FPABAYIMLT
110	EVREC	Bay local/remote switch ind.	RP-570 only Optional	FPABAYELRS
110	AI	Bay local/remote switch ind.	ANSI only	FPABAYELRS
111-114	TERM/OS	Command termination indication for output signals Output status object	IEC 60870-5-101/104 and -103 only Optional Output status object when station type is DNP 3.0, optional	FPASTATOBC
115	EVREC	Interl. in use/out of use (HW)	RP-570 only	FPABAYEINH
116	EVREC	Interl. in use/out of use (SW)-ind.	Only RP-570 only	FPABAYEINH
117,	TERM	Command termination indication for output signals	IEC 60870-5-101/104 and -103 only	FPASTATINH

6.5.3.9 Scale objects

At the first installation, the bay creates the scale 1_1 (linear 1:1 scale). For SPACOM/SRIO (ANSI) station type. The installation also creates the scale FPU_1_1000 (linear 1:1000 scale).

6.5.3.10 Event channels

For SPACOM/SRIO (ANSI) station type, the installation creates the event channel BGU_AI2DB, which is set to activate the command procedure BGU_AI2DB.

6.6 Switching device

6.6.1 Standard function installation

This section describes the installation of the switching device standard function from the Power Process Library. The standard function for the switching device is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Process/Switching Device). This standard function is configured by using configuration tools, such as the Object Navigator.

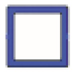
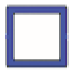
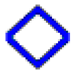
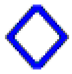
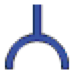
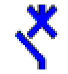
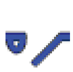
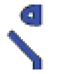
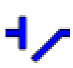
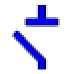
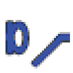
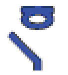
6.6.1.1 Symbol installation

The Power Process symbol for the switching device standard function is installed by using the following Display Builder functions:

1. Object Browser
2. Palette

For more information, see the SYS600 Process Display Design manual.

Table 30: Power Process symbol for switching device standard function

File name		Representation	Symbol	
Horizontal	Vertical		Horizontal	Vertical
Breaker H.sd	Breaker.sd	ANSI		
Disconnector H.sd	Disconnector.sd	ANSI		
Truck H.sd	Truck.sd	Common		
Breaker H.sd	Breaker.sd	IEC		
Contactor H.sd	Contactor.sd	IEC		
Disconnector H.sd	Disconnector.sd	IEC		
Load Breaker H.sd	Load Breaker.sd	IEC		
Earth H.sd	Earth.sd	Common		

6.6.2 Standard function configuration

6.6.2.1 Configuration with tools

The switching device standard function has the following attributes to be configured with the Object Navigator:

Table 31: Configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of database process objects	-
Table continues on next page		

Attribute	Meaning	Default
SWITCHING_DEVICE_TYPE	Type of switching device	Circuit breaker
SWITCHING_DEVICE_PURPOSE	Usage of the selected switching device type	Circuit breaker
SWITCH_SECTION	Switch section of the truck or 3-state switch	Truck: first section 3-state switch: disconnector
STATION_TYPE	Type of control device & protocol	LON
INDICATION_TYPE	Type of position indication process object	Double binary
MOTORIZED	Selection of motorized or manually operated	Motorized
CMD_PARAMETER	Parameter for control method for IEC 60870-5-101/104 and DNP 3.0 protocol	N/A
CONTROL_TYPE	Type of control process object(s)	Secured control with 4 binary outputs
OUTPUT_STATUS	Process objects for output status with DNP 3.0 protocol	N/A
CONTROL_PULSE_LENGTH	Length of control pulse	0
CONTROL_BITS	Bits to be used to send a control command	Empty vector
AUTHORIZATION_GROUP	Name of the authorization group	MV_CONTROL
BAY_LR_POLARITY	Polarity of bay L/R switch	L1
TAGOUT	Release the Tagout function for this switching device	Not used
AUXILIARY_PLUG	Selection if the switching device is mounted in the truck	Stand-alone switch
EVENT_RECORDING	Process objects for event recording with RP-570	No process objects
INTERLOCKING_BYPASS	Specifies whether the interlocking bypass function is used or not	FALSE
SYNCHROCHECK_BYPASS	Specifies whether the synchrocheck bypass function is used or not	FALSE
OPC_ITEM_PREFIX	Specifies the OPC Item Prefix used as IEC 61850 Instance information	IEC61850 Subnetwork.IED1.LD1
OPC_LN_INSTANCES	Specifies the OPC Logical Node names used as IEC 61850 Instance information	VECTOR("CSWI1","CILO1")
QUALIFIERS	Qualifier is the value written to the QL attribute of process object when writing the control command. Qualifiers are given as a vector of integers.	Minimum length is 0 (default) and maximum length 5
ADD_CAUSE_IX	Index for process object for AddCause value.	55
Table continues on next page		

Attribute	Meaning	Default
OPERATOR_PLACE_HANDLING	Specifies the operator place (local/ Station/NCC) handling of the device	None
CONTROL_SUPERVISION_TIMEOUT	Specifies the timeout for the indication change after the execute command	0
CONTROL_SUPERVISION_ALARM	Specifies the alarm class for the control supervision alarm	1

For more detailed description on the configurable attributes, see Section 6.15.

6.6.2.2 Configuration of process objects

Depending on the configuration of the switching device standard function, the tools create a certain set of process objects in the database. Those process objects that will be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

The tools create process objects based on the following list:

Table 32: Process objects created for switching device

Index	Explanation	Purpose
10	Position indication	Used for position indication of the switching device state open/closed/ intermediate/faulty.
11	Open select or open execute	Depending on the defined control type it is used for sending Open select or Open execute to the control unit.
12	Close select or close execute	Depending on the defined control type it is used for sending Close select or Close execute to the control unit.
13	Execute selected or select open/close and execute/cancel or execute open	Depending on the defined control type it is used for sending Execute or Select open/close and Execute/Cancel to the control unit.
14	Cancel command or execute close command	Used for sending Cancel command or Execute close command to the control unit.
15	External control blocking (optional)	Receives a control blocking signal from the control unit and prevents the control actions in the single line diagram.
16	Open interlocked (optional)	Receives a control blocking signal for Open command from the control unit and prevents the Open command in the single line diagram.
17	Close interlocked (optional)	Receives a control blocking signal for Close command from the control unit and prevents the Close command in the single line diagram.
18	Interlocking cause (optional)	Receives an interlocking cause, that is, an integer value that represents a reason for an unsuccessful control action.
19	Selected on monitor (optional)	An internal tag that shows if the switching device is selected on different picture(s) or monitor(s).
20	Command event (optional)	An internal or external tag that shows if the switching device is right under the command sequence.
21	Auxiliary plug state (optional)	An internal or external tag to indicate if the switching device is fully racked out from the cubicle, and therefore it should not be shown in the single line diagram.
25	Cancel command	Used for sending a Cancel command to the control unit.
Table continues on next page		

Index	Explanation	Purpose
41	Open blocked	Receives a control blocking signal from the control unit and prevents open control actions in control dialog.
42	Close blocked	Receives a control blocking signal from the control unit and prevents close control actions in control dialog.
49	Tagout indication	Used for Tagout priority value storage and Tagout text storage.
50	Tagout history	Used for tagout activity event generation.
55 (configurable)	Add cause of command	Used for showing the failure of select/execute operation.
70	Local/Remote-switch	Used for Device level Local/Remote-switch indication
71	Operator place-switch	Used for device level operator place (Station/NCC) indication
72	Operator place-switch command	Used for device level operator place (Station/NCC) command
(108) 110	Position indication other type than double indication	Receives position indication as a single indication (110), analog input (110) or two single indications (110-open, 108-close).
111-114	Indication for command termination or indication for output status (optional)	Indicates with IEC 60870-5-101/104 and -103 whether the issued command was successful or not. With DNP 3.0 indicates the status of output object.
115	External control blocking (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
116	Open interlocked (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
117	Close interlocked (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
120	Command event (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
121	Auxiliary plug state (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
(208) 210	Position indication other type than double indication	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.

6.6.2.3 Example of a switching device configuration

The configuration of a switching device as a circuit breaker can be given as an example:

Table 33: Configuration of the switching device

Attribute	Value
STATION_NAME	Eastwick
BAY_NAME	Outgoing HA2
DEVICE_NAME	Q0
P_OBJECT_LN	ESTHA02Q0
SWITCHING_DEVICE_TYPE	Circuit breaker
SWITCHING_DEVICE_PURPOSE	Main busbar circuit breaker
SWITCH_SECTION	N/A
Table continues on next page	

Attribute	Value
STATION_TYPE	LON
INDICATION_TYPE	Double binary
MOTORIZED	Yes
CMD_PARAMETER	N/A
CONTROL_TYPE	Secured control with 4 binary outputs
OUTPUT_STATUS	N/A
CONTROL_PULSE_LENGTH	N/A
CONTROL_BITS	N/A
AUTHORIZATION_GROUP	MV_CONTROL
BAY_LR_POLARITY	L1
TAGOUT	No
AUXILIARY_PLUG	No
EVENT_RECORDING	N/A

6.6.3 Application engineering information

6.6.3.1 Structure of switching device standard function

This section describes the structure of the switching device standard function. All subdrawing files, form pictures, help and other text files, as well as database objects are included. The switching device is a part of the standard functions of the Power Process Library and has a directory.

6.6.3.2 Files

The table below lists all switching device standard function related files and their functionality.

Table 34: Switching device standard function related files

File	Functionality	Path
Breaker.sd	SA ANSI symbol for breaker	/PROG/GRAPHICSENGINE/PALETTE/03 - SA_ANSI
Disconnector.sd	SA ANSI symbol for disconnector	/PROG/GRAPHICSENGINE/PALETTE/03 - SA_ANSI
Truck.sd	SA Common symbol for truck	/PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Breaker.sd	SA IEC symbol for breaker	/PROG/GRAPHICSENGINE/PALETTE/02 - SA_IEC
Contactor.sd	SA IEC symbol for contactor	/PROG/GRAPHICSENGINE/PALETTE/02 - SA_IEC
Disconnector.sd	SA IEC symbol for disconnector	/PROG/GRAPHICSENGINE/PALETTE/02 - SA_IEC
Load Breaker.sd	SA IEC symbol for load breaker	/PROG/GRAPHICSENGINE/PALETTE/02 - SA_IEC
Earth.sd	SA Common symbol for earth	/PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

6.6.3.3 Language text file

The following Text Translation Tool compatible text files are used by the switching device standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 35: Text Translation Tool compatible text files

File	Functionality
SAI_SSW.TXT	Text file for the database creation of the standard function

6.6.3.4 Help text file

The path to the switching device help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 36: Switching device help text file

File	Functionality
SAI_SSW.HLP	Standard function installation help file

6.6.3.5 Configuration files

The following configuration files are used by the switching device standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 37: Configuration files used by the switching device standard function

File	Functionality
SAI_SSW.DAT	Contains the configuration data for the switching device when it is created, modified or deleted by the configuration tools.
SAI_SSW.POT	Contains the process object definitions for the Object Navigator.
INDEX5_B1.TXT	Selection list of the Power Process Library standard functions for the installation tool.

6.6.3.6 Other text files

The following text file is used by the switching device. The path is

Table 38: Another text file used by the switching device

File	Functionality
BGU_AUTH.TXT	Contains the SCIL code for the station authority check. The code is executed by the dialog of the switching device.

6.6.3.7 Command procedures

At the first installation, the bay creates the command procedure BGU_UPDATE, which updates process objects when they are update-deblocked. This feature is similar to the SYS600 Blocking Display function. It is also used when simulation mode is disabled in the control dialog.

For the indication type of single indication (BI), the installation will create the command procedure BGU_BI2DB, which updates the position indication double binary from the binary input.

For the indication type of an analog input (AI), the installation will create the command procedure BGU_AI2DB, which updates the position indication double binary from the analog input.

For the indication type of two single indication (2xBI), the installation creates the command procedures BGU_2BI2DB and BGU_SETDB, which are used to update the position indication double binary from two binary inputs.

Table 39: Command procedures used by the switching device

BGU_CONTROL Performs the actual control operations	
Function	**Functionality**
OBJECT_INFO	Gets configuration of the standard function.
GET_OI_TEXT	Gets object identification text.
SELECT_ON_MONITOR	Sets standard function to selected on monitor state.
OPEN_SELECT	Performs open select command on standard function.Uses text file /sa_lib/base/bbone/use/BGU_SW_SEL.TXT
UNSELECT_ON_MONITOR	Removes selected on monitor state from standard function.
CLOSE_SELECT	Performs close select command on standard function.Uses text file /sa_lib/base/bbone/use/BGU_SW_SEL.TXT
OPEN_EXECUTE	Performs open execute command on standard function.Uses text file /sa_lib/base/bbone/use/BGU_SW_EXE.TXT
CLOSE_EXECUTE	Performs close execute command on standard function.Uses text file /sa_lib/base/bbone/use/BGU_SW_EXE.TXT
CANCEL	Performs cancel command on standard function. Uses text file.
SEARCH_OBJECTS	Searches for objects that belong to standard function.
CHECK_AUTHORITY	Checks if the user is authorized to control the station.
CHECK_AUTHORITY_WORKPLACE	Checks if the used workstation is authorized to make control operations.
INTERRUPT_AR_SEQUENCE	Interrupts a running auto-reclosure sequence.
SET_SIMULATION	Sets simulation state for objects that belong to standard function.
GET_SIMULATION	Gets simulation state of objects that belong to standard function.
GET_ALARMS	Gets alarms of standard function.
ACKNOWLEDGE_ALARM	Acknowledges single alarm of standard function.
GET_OFFSETS	Gets index offsets for indexes that are used for indicating command event, selection on monitor, control blocking, alarm, external blocking, open blocking and close blocking.
BGU_BLOCK Performs the blocking operations	
Function	Description
GET_BLOCKING_STATE	Gets blocking state for objects that belong to the standard function.
SET_BLOCKING_STATE	Sets blocking state for the objects that belong to the standard function. If standard function is a bay, applies to all objects that belong to the bay. If standard function is a station, applies to all objects that belong to the station.
GET_ENABLED_BLOCKINGS	Disables blocking possibility in some standard functions. For example, control blocking cannot be used with a measurement and alarm indicator standard functions.

6.6.3.8 Process objects

Depending on the configuration the following process objects are common to all switching devices.

Table 40: The process objects to be created with the tools.

Index	Obj. type	Process object	Remarks	Group identifier
70	BI	Local/Remote-switch	Optional	FPQxxxILRS
71	BI	Operator place-switch	Optional	FPQxxxIOPS
72	BO	Operator place-switch command	Optional	FPQxxxCOPS

The following process objects are created depending on the configuration of the switching device.

Table 41: Manually operated circuit breaker, disconnector or earth switch. The process objects to be created with the tools.

Index	Obj. type	Process object	Remarks	Group identifier
10	DB	Position indication		FPQxxxIPOS
19	AI	Object selected on monitor	Optional	FPQxxxXMEV
21	BI	Auxiliary plug indication	Optional	FPQxxxIAUX
49	DI	Tagout indication	Tagout released	FPXTAGXIND
50	DI	Tagout history	Tagout released	FPXTAGXEVT
108	BI	Close position indication	Indication type 2xBI	FPQxxxEPOS
110	BI or AI	Position indication or open position indication	Ind. type AI, BI or 2xBI	FPQxxxEPOS
110	EVREC DB	Event recording for position indication	Indication type DB RP-570 only Optional	FPQxxxEPOS
121	EVREC BI	Event recording for auxiliary plug indication	RP-570 only Optional	FPQxxxEAUX
208	EVREC BI	Event recording for close position indication	Indication type 2xBI RP-570 only Optional	FPQxxxEPOS
210	EVREC BI or AI	Event recording for position indication or open position indication	Ind. type AI, BI or 2xBI RP-570 only Optional	FPQxxxEPOS

The three characters (xxx) in the group identifier are replaced by other characters based on the selection made with configurable attributes SWITCHING_DEVICE_TYPE and SWITCHING_DEVICE_PURPOSE.

Table 42: Motorized circuit breaker, disconnector or earth switch. The process objects to be created with the tools.

Index	Obj. type	Process object	Remarks	Group identifier
10	DB	Position indication		FPQxxxIPOS
11	BO	Open command or Open execute	Sec. control with 4xBO, 2xBO or 2xDO	FPQxxxCOBC
12	BO	Close command or Close execute	Sec. control with 4xBO, 2xBO or 2xDO	FPQxxxCOBC
13	BO	Execute command or Open/Close select and Execute/Cancel command or Open execute command	Sec. control with 4xBO, or 5xBO	FPQxxxCOBC
14	BO	Cancel command or close execute command	Sec. control with 4xBO or 5xBO	FPQxxxCOBC
15	BI/AI	External control blocking	AI when the selected station type is IEC 61850-8	FPQxxxIBLK
16	BI	External interlock for open cmd	Optional	FPQxxxIINH
17	BI	External interlock for close cmd	Optional	FPQxxxIINH
18	AI	External interlocking cause	Optional	FPQxxxIINH
19	AI	Object selected on monitor	Optional	FPQxxxXMEV
20	BI	Object command event	Optional	FPQxxxICEV
21	BI	Auxiliary plug indication	Optional	FPQxxxIAUX
25	BO	Cancel command	5xBO	FPQxxxCOBC
41	BI	Externally open blocked	Created with IEC 61850-8 and DNP 3.0 protocols	FPQxxxIBLK
42	BI	Externally close blocked	Created with IEC 61850-8 and DNP 3.0 protocols	FPQxxxIBLK
49	DI	Tagout indication	Tagout released	FPXTAGXIND
50	DI	Tagout history	Tagout released	FPXTAGXEVT
55	AI	Add cause of command	Optional	FPQxxxTAEC
108	BI	Close position indication	Indication type 2xBI	FPQxxxEPOS

Table continues on next page

Index	Obj. type	Process object	Remarks	Group identifier
110	BI or AI	Position indication or open position indication	Ind. type AI, BI or 2xBI	FPQxxxEPOS
110	EVREC DB	Event recording for position indication	Indication type DB RP-570 only Optional	FPQxxxEPOS
111-114	TERM/OS	Command termination indication for output signals	IEC 60870-5-101/104 and -103 only Optional	FPQxxxTOBC
115	EVREC BI	Event recording for external control blocking	Only RP-570 Optional	FPQxxxEBLK
116	EVREC BI	Event recording for external interlock for open cmd	Only RP-570 Optional	FPQxxxEINH
117	EVREC BI	Event recording for external interlock for close cmd	Only RP-570 Optional	FPQxxxEINH
120	EVREC BI	Event recording for object command event	RP-570 only Optional	FPQxxxECEV
121	EVREC BI	Event recording for auxiliary plug indication	RP-570 only Optional	FPQxxxEAUX
208	EVREC BI	Event recording for close position indication	Indication type 2xBI RP-570 only Optional	FPQxxxEPOS
210	EVREC BI or AI	Event recording for position indication or open position indication	Ind. type AI, BI or 2xBI RP-570 only Optional	FPQxxxEPOS
252	AI	Calculated interlocking cause	Optional	FPQxxxIINH

The three characters (xxx) in the group identifier are replaced by other characters based on the selection made with configurable attributes SWITCHING_DEVICE_TYPE and SWITCHING_DEVICE_PURPOSE.

Table 43: Manually operated truck. The process objects to be created with the tools.

Index	Obj. type	Process object	Remarks	Group identifier
30	DB	Position indication		FPQT01IPOS
39	AI	Object selected on monitor	Optional	FPQT01XMEV
49	DI	Tagout indication	Tagout released	FPXTAGXIND
50	DI	Tagout history	Tagout released	FPXTAGXEVT
128	BI	Close position indication	Indication type 2xBI	FPQT01EPOS

Table continues on next page

Index	Obj. type	Process object	Remarks	Group identifier
130	BI or AI	Position indication or open position indication	Ind. type AI, BI or 2xBI	FPQT01EPOS
130	EVREC DB	Event recording for position indication	Indication type DB RP-570 only Optional	FPQT01EPOS
228	EVREC BI	Event recording for close position indication	Indication type 2xBI RP-570 only Optional	FPQT01EPOS
230	EVREC BI or AI	Event recording for position indication or open position indication	Ind. type AI, BI or 2xBI RP-570 only Optional	FPQT01EPOS

Table 44: Motorized truck. The process objects to be created with the tools.

Index	Obj. type	Process object	Remarks	Group identifier
30	DB	Position indication		FPQT01IPOS
31	BO	Open command or Open execute	Sec. control with 4xBO, 2xBO or 2xDO	FPQT01COBC
32	BO	Close command or Close execute	Sec. control with 4xBO, 2xBO or 2xDO	FPQT01COBC
33	BO	Execute command or Open/Close select and Execute/Cancel command or Open execute command	Sec. control with	FPQT01COBC
34	BO	Cancel command or Close execute command	Sec. control with 4xBO or 5xBO	FPQT01COBC
35	BI/AI	External control blocking	AI when the selected station type is IEC 61850-8	FPQT01IBLK
36	BI	External interlock for open cmd	Optional	FPQT01IINH
37	BI	External interlock for close cmd	Optional	FPQT01IINH
38	AI	External interlocking cause	Optional	FPQT01IINH
39	AI	Object selected on monitor	Optional	FPQT01XMEV
40	BI	Object command event	Optional	FPQT01ICEV
45	BO	Cancel command	5xBO	FPQT01COBC
49	DI	Tagout indication	Tagout released	FPXTAGXIND
50	DI	Tagout history	Tagout released	FPXTAGXEVT
61	BI	Externally open blocked	Created with IEC 61850-8 and DNP 3.0 protocols	FPQT01IBLK

Table continues on next page

Index	Obj. type	Process object	Remarks	Group identifier
62	BI	Externally close blocked	Created with IEC 61850-8 and DNP 3.0 protocols	FPQT01IBLK
128	BI	Close position indication	Indication type 2xBI	FPQT01EPOS
130	BI or AI	Position indication or open position indication	Ind. type AI, BI or 2xBI	FPQT01EPOS
130	EVREC DB	Event recording for position indication	Indication type DB RP-570 only Optional	FPQT01EPOS
131-134	TERM	Command termination indication for output signals	IEC 60870-5-101/104 and -103 only Optional	FPQxxxTOBC
135	EVREC BI	Event recording for external control blocking	Only RP-570 Optional	FPQT01EBLK
136	EVREC BI	Event recording for external interlock for open cmd	Only RP-570 Optional	FPQT01EINH
137	EVREC BI	Event recording for external interlock for close cmd	Only RP-570 Optional	FPQT01EINH
140	EVREC BI	Event recording for object command event	RP-570 only Optional	FPQT01ECEV
228	EVREC BI	Event recording for close position indication	Indication type 2xBI RP-570 only Optional	FPQT01EPOS
230	EVREC BI or AI	Event recording for position indication or open position indication	Ind. type AI, BI or 2xBI RP-570 only Optional	FPQT01EPOS
251	AI	Calculated interlocking cause	Optional	FPQT01IINH

Table 45: Manually operated three-state switch. The process objects to be created with the tools.

Index	Obj. type	Process object	Remarks	Group identifier
10	DB	Position indication	Disconnector section	FPQ301IPOS
11	DB	Position indication	Earth switch section	FPQ301IPOS
25	AI	Object selected on monitor	Optional	FPQ301XMEV
49	DI	Tagout indication	Tagout released	FPXTAGXIND
50	DI	Tagout history	Tagout released	FPXTAGXEVT
108	BI	Close position indication	Indication type 2xBI Disconnector section	FPQ301EPOS
Table continues on next page				

Index	Obj. type	Process object	Remarks	Group identifier
109	BI	Earth position indication	Indication type 2xBI Earth switch section	FPQ301EPOS
110	BI or AI	Position indication or open position indication	Ind. type AI, BI or 2xBI Disconnector section	FPQ301EPOS
111	BI or AI	Position indication or free position indication	Ind. type AI, BI or 2xBI Earth switch section	FPQ301EPOS
110	EVREC DB	Event recording for position indication	Indication type DB RP-570 only Optional	FPQ301EPOS
111	EVREC DB	Event recording for position indication	Indication type DB RP-570 only Optional	FPQ301EPOS
208	EVREC BI	Event recording for close position indication	Indication type 2xBI RP-570 only Optional	FPQ301EPOS
209	EVREC BI	Event recording for earth position indication	Indication type 2xBI RP-570 only Optional	FPQ301EPOS
210	EVREC BI or AI	Event recording for position indication or open position indication	Ind. type AI, BI or 2xBI RP-570 only Optional	FPQ301EPOS
211	EVREC BI or AI	Event recording for position indication or free position indication	Ind. type AI, BI or 2xBI RP-570 only Optional	FPQ301EPOS

Table 46: Motorized three-state switch. The process objects to be created with the tools.

Index	Obj. type	Process object	Remarks	Group identifier
10	DB	Position indication	Disconnector section	FPQ301IPOS
11	DB	Position indication	Earth switch section	FPQ301IPOS
12	BO	Open command or Open execute	Sec. control with 4xBO, 2xBO or 2xDO	FPQ301COBC
13	BO	Close command or Close execute	Sec. control with 4xBO, 2xBO or 2xDO	FPQ301COBC
14	BO	Free command or Free execute	Sec. control with 4xBO, 2xBO or 2xDO	FPQ301COBC
15	BO	Earth command or Earth execute	Sec. control with 4xBO, 2xBO or 2xDO	FPQ301COBC
16	BO	Execute command or open/close select and execute/ cancel cmd	Sec. control with 4xBO, BO or AO Direct control with BO	FPQ301COBC
Table continues on next page				

Index	Obj. type	Process object	Remarks	Group identifier
17	BO	Cancel command or free/earth select and execute/cancel cmd	Sec. control with 4xBO, BO or AO Direct control with BO	FPQ301COBC
18	BI	External control blocking	Optional	FPQ301IBLK
19	BI	External interlock for open cmd	Optional	FPQ301IINH
20	BI	External interlock for close cmd	Optional	FPQ301IINH
21	BI	External interlock for free cmd	Optional	FPQ301IINH
22	BI	External interlock for earth cmd	Optional	FPQ301IINH
23	AI	External inter-locking cause	Optional	FPQ301IINH
24	AI	External interlocking cause	Optional	FPQ301IINH
25	AI	Object selected on monitor	Optional	FPQ301XMEV
26	BI	Object command event	Optional	FPQ301ICEV
41	BI	Externally open blocked	Created with IEC 61850-8 and DNP 3.0 protocols, disconnector section	FPQ301IBLK
42	BI	Externally close blocked	Created with IEC 61850-8 and DNP 3.0 protocols disconnector section	FPQ301IBLK
49	DI	Tagout indication	Tagout released	FPXTAGXIND
50	DI	Tagout history	Tagout released	FPXTAGXEVT
61	BI	Externally open blocked	Created with IEC 61850-8 and DNP 3.0 protocols, earth switch section	FPQ301IBLK
62	BI	Externally close blocked	Created with IEC 61850-8 and DNP 3.0 protocols, earth switch section	FPQ301IBLK
108	BI	Close position indication	Indication type 2xBI	FPQ301EPOS
109	BI	Earth position indication	Indication type 2xBI	FPQ301EPOS
110	BI or AI	Position indication or open position indication	Ind. type AI, BI or 2xBI	FPQ301EPOS
111	BI or AI	Position indication or free position indication	Ind. type AI, BI or 2xBI	FPQ301EPOS

Table continues on next page

Index	Obj. type	Process object	Remarks	Group identifier
110	EVREC DB	Event recording for position indication	Indication type DB RP-570 only Optional	FPQ301EPOS
111	EVREC DB	Event recording for position indication	Indication type DB RP-570 only Optional	FPQ301EPOS
112-117	TERM	Command termination indication for output signals	IEC 60870-5-101/104 and -103 only Optional	FPQxxxTOBC
118	EVREC BI	Event recording for external control blocking	RP-570 only Optional	FPQ301EBLK
119	EVREC BI	Event recording for external interlock for open cmd	RP-570 only Optional	FPQ301EINH
120	EVREC BI	Event recording for external interlock for close cmd	RP-570 only Optional	FPQ301EINH
121	EVREC BI	Event recording for external interlock for free cmd	RP-570 only Optional	FPQ301EINH
122	EVREC BI	Event recording for external interlock for earth cmd	RP-570 only Optional	FPQ301EINH
126	EVREC BI	Event recording for object command event	RP-570 only Optional	FPQ301ECEV
208	EVREC BI	Event recording for close position indication	Indication type 2xBI RP-570 only Optional	FPQ301EPOS
209	EVREC BI	Event recording for earth position indication	Indication type 2xBI RP-570 only Optional	FPQ301EPOS
210	EVREC BI or AI	Event recording for position indication or open position indication	Ind. type AI, BI or 2xBI RP-570 only Optional	FPQ301EPOS
211	EVREC BI or AI	Event recording for position indication or free position indication	Ind. type AI, BI or 2xBI RP-570 only Optional	FPQ301EPOS
251	AI	Calculated interlocking cause	Optional	FPQ301IINH
252	AI	Calculated interlocking cause	Optional	FPQ301IINH

6.6.3.9 Scale objects

At the first installation, the switching device creates the scale 1_1 (linear 1:1 scale). For indication type of analog input, the installation will also create the scale BGU_1_1000 (linear 1:1000 scale).

6.6.3.10 Operation counters

In a switch control dialog, the CO, CV, CE and CL attributes of the DB process object are used for operation counting. Regardless of whether the counting is used or not, the attributes are always readable.

6.7 Tagout

Tagout is a procedure for disabling equipment to protect maintenance and service personnel from either an unexpected release of energy, or an accidental start-up while performing field activities.

The Tagout function provides locks (control block)/inhibits (update blocks) on control/data points.

A tagout can be assigned to switching devices. The main focus is on breaker tagout facility. The facility allows for selection from a set of user defined tagout classes, which specify lock, remote lock, inhibit, remote inhibit, priority and other class attributes.

Lock, remote lock specifies whether a control can pass through locally or remotely. Inhibit, remote inhibit specifies whether status will be reported locally or remotely.

The system allows operators to lock control of devices by means of a secure, multi-level tag feature. This feature allows operators to apply an unlimited number of tags to each point, each tag being stored with a date/time stamp and an operator entered description. Every tag and un-tag operation can be logged and it generates an event.

The system permits no means of bypassing the control inhibit caused by a tag. This applies to any and every application interacting with the system.

A tag symbol indicating the control inhibit conditions can be displayed next to the device on all displays where the device is presented. The following information can be defined for each tag type:

- tag symbol to be displayed
- tag priority
- tag reason
- type of controls that will be inhibited by the tag, i.e. open and close, or information only (no control inhibit)
- other optional tag information

The Tagout function can be enabled with the Power Process Library Standard Function Configuration tool.

6.7.1 Standard function installation

For the Tagout function to be taken into use for selected switching devices, it has to be configured. For that purpose, the attribute TAGOUT within the Standard Function Installation and Configuration Tool for switching devices can be used. This attribute is disabled by default. It will be enabled after the Tagout function has been initialized. The initialization takes place with the first Tagout Class Editor usage.

The tagout attribute will be used for the Control dialog to show/hide the **Tagout** tab.

6.7.2 Tagout class properties

A set of attributes forms a Tagout Class. Tagout classes can be adjusted with the Tagout Class Editor.

Table 47: Tagout class attributes

Attribute name	Description and properties
Mandatory attributes	
Date/Time	Date and time of tagout activation and modification.
Priority	None, Very Low, Low, Medium, High, Very High. The list content can be modified with the Tagout Class Editor.
Reason	Men at work, equipment in service, abnormal condition, other. The list content can be modified with the Tagout Class Editor.
Owner	The user/owner name will be taken over from the current active MicroSCADA user.
Comments	Free comment text
Action attributes	
Action methods	Lock, block control local (Station), remote (NCC) Inhibit, block update local (Station), remote (NCC) Other standard blocking attributes: alarm blocking event blocking printout blocking reprocessing blocking
Tag Repr	Tagout symbol reference (hidden to user)
Tag text	For Symbol Type Tagout Text
Up to 6 optional attributes	
Optional attribute 1..5	Predefined name for 5 optional attributes of text type
Optional Attribute 6	Predefined name for optional attribute 6 of selector type

The attribute names and their properties can be modified with the Tagout Class Editor.

Each attribute has the following basic properties:

1. Name (internal name, not editable),
2. Title, editable with Tagout Class Editor,
3. Used or not used within this class,
4. Visible in tagout views as there are, **Control** Dialog Tagout tab and Tagout List,
5. Enabled, attribute value is visible but the user cannot change the default value,
6. Logging release,
7. History/event handling release.

6.7.2.1 Tagout storage

Class data
Only one tagout class can be active at a time. Out of the amount of pre-configured classes, one class can be activated.

The active tagout class can only be changed when there is no tagout assigned to any switching device.

All class properties are stored in a SYS600 SCIL Database (SDB) located in the application \apl_ directory.

The tagout class database is located in sc/'application name'/apl_.

The database file name is apl_tagout.sdb.

Active tagout

Active tagout(s) on a control/data point will be stored in the tagout class database with their logical name (LN) and index (IX) as section name.

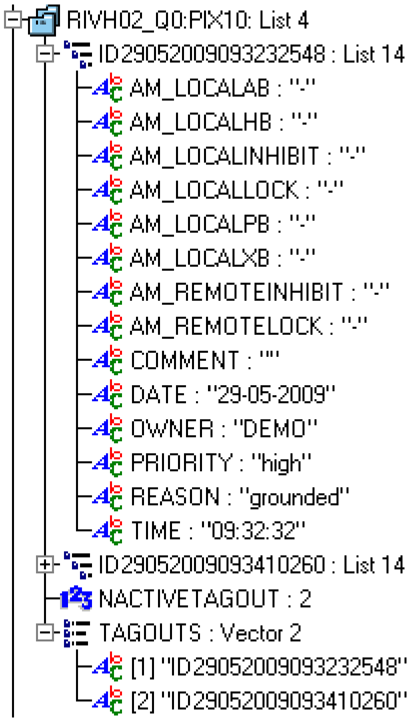


Figure 47: Active Tagout storage structure in Tagout Class Editor

Multiple tagouts, as well as multiple priorities on one object are supported. Depending on the used tagout presentation type, the tagout indicators for either the highest priority, or all priorities will be shown on the display.

Tagout event and logging

All Tagout changes generate a new event in the event list. A printout of the tagout can also be generated if the option is released in the process object database.

Event generation for certain attribute and tagout changes can be released with the Tagout Class Editor.

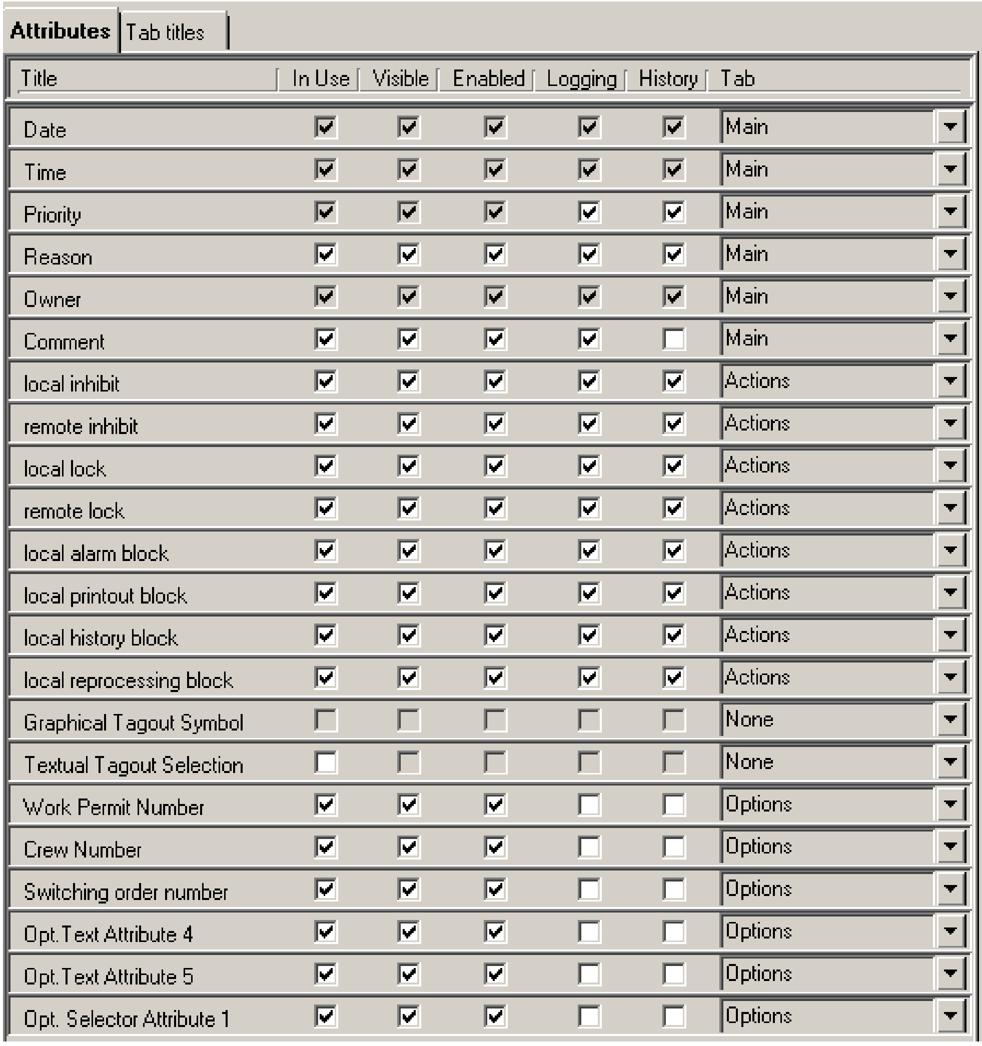


Figure 48: Tagout class attribute event handling settings

The following information will form the event entry:

- Date & time of change
- Object Identifier (OI)
- User name, in Object Text (OX)
- Attribute information, in Object Text (OX)
- Kind of tagout activity, add/remove or edit tagout, in Event State Text (EH)

The Object Text (OX) attribute for the different activities has the following format:

For edit and remove attribute actions:

Date + Time + Object Identifier + Prefix OX + Owner Name + Separator + Attr. Title + Separator + Attr. Value

29-05-09 09:32:32.898	Rivers	Wilbur	Q 0	Tagout: DEMO,Priority,high	Attr.value changed
29-05-09 10:45:08.851	Rivers	Wilbur	Q 0	Tagout: DEMO,Priority,high	Attribute removed

For tagout add and removal actions:

29-05-09 10:45:09.866	Rivers	Wilbur	Q O	Tagout: ,Owner DEMO	Removed
29-05-09 09:34:10.804	Rivers	Wilbur	Q O	Tagout: ,Owner DEMO	Insert

Date + Time + Object Identifier + Prefix OX + Separator + Owner Attr. Title + Owner Name

Table 48: Event descriptions

Event Item	Description	Definition in Class Editor	Default
Date Time	Date and time of tagout activity	Tagout Date attribute value Tagout Time attribute value	
Object identifier	From position indication object		
Prefix OX	Tagout name descriptor	Root node\Settings\History \Tagout prefix OX	Tagout:
Owner name	Current logged on MicroSCADA user		
Owner attribute title	Title for the attribute Owner	Active class\attribute\owner\title	Owner
Separator	Separator between each event item in the object	Root node\Settings\History \Separator	,
Attribute Title	Title for the attribute causing the event	Active class\attribute title	
Attribute Value	New value assigned to the attribute causing the event	Active class\attribute value	

The Event state text uses the event handling object SAGR_TAGOUT_HIST. The state text can be modified with the Tagout Class Editor by editing the root/settings/history section.

	Tagout Event Handling State Text (ST)
Attribute edit	Attr.value changed
Attribute removal	Attribute removed
Tagout add	Insert
Tagout removal	Removed

Figure 49: Event state text in event handling object

All activities related to tagout will be logged in an ASCII text file.

The tagout log file is located in sc/'application name'/apl_.

The file name is apl_tagout'year'.log.

One log file will be created for each year.

The following data will be logged:

- Date and time
- Object Identifier (OI)
- Tagout owner name
- Kind of Tagout activity, add/remove or edit Tagout
- Changed attribute name and value

Tagout Class Editor can be used for releasing tagout activity logging for all attributes, see Figure 49.

The log file can be viewed with Log Viewer or with any text editor. Log Viewer is included in the **Tagout** List, but can only used for viewing and not editing. Select a year from the drop down list to view the content of that year's log file.

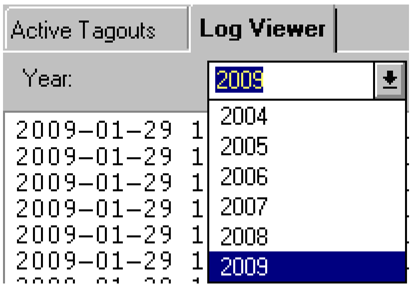


Figure 50: Log file selection in Log Viewer

The log file entry for the different activities has the following format:

For edit and remove attribute actions:

Date + Time + Separator + Object Identifier + Separator + Owner Name + Separator + Tagout activity + Separator + Attr. Title + Separator + Attr. Value

29-05-2009 17:33:44	Rivers Wilbur Q0	DEMO	Attr. value changed	Reason	Out of service
29-05-2009 17:29:14	Rivers Wilbur Q0	DEMO	Attribute removed	local lock	Set

For tagout adding and removal:

Date + Time + Separator + Object Identifier + Separator + Owner Name + Separator + Tagout activity

29-05-2009 17:29:23	Rivers Wilbur QO	DEMO	Removed
29-05-2009 17:29:14	Rivers Wilbur QO	DEMO	New Tagout added

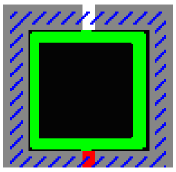
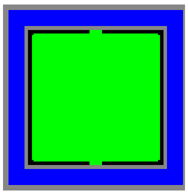
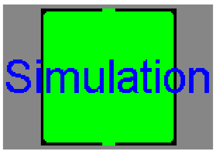
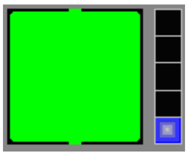
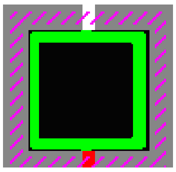
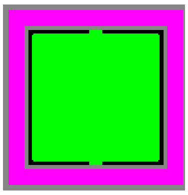
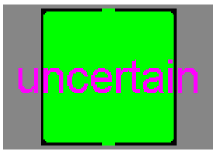
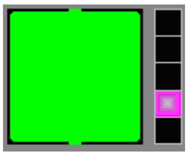
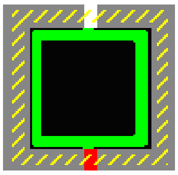
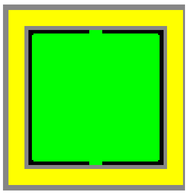
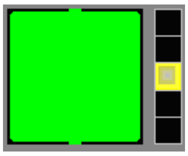
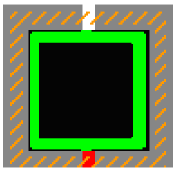
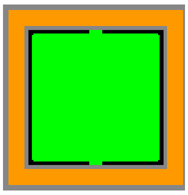
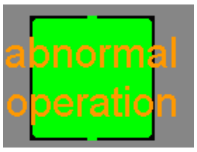
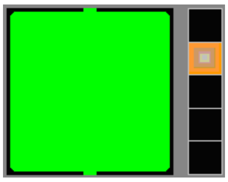
Table 49: Log text description

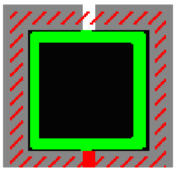
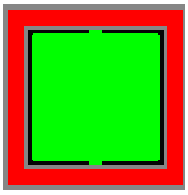
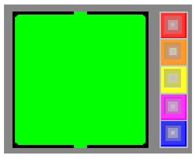
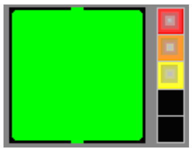
Log item	Description	Definition in Class Editor
Date Time	Date and time of tagout activity	Tagout Date attribute value
Object Identifier	From position indication object	
Owner Name	Current logged on MicroSCADA user	
Tagout Activity	| | Tagout Action Text | Attribute edit: Attr.value changed Attribute removal: Attribute removed Tagout add: New Tagout added Tagout removal: removed	Root node\Settings\Logging \Tagout Action Text
Separator	Separator between each log item	Root node\Settings\Logging \Separator
Attributes Title	Attribute title for the attribute causing the entry log	Active class\attribute title
Attributes value	New value assigned to the attribute causing the log entry	Active class\attribute value

6.7.3 Tagout presentation symbols

The following table shows some possible symbols. In this example, the ANSI type of breaker function is used.

Table 50: Tagout presentation symbols

Presentation				Priority value (Bit weight)
Option 1	Option 2	Option 3	Option 4	
Priority order			5 priority levels	
Pattern frame	Solid frame	Text	Indicator	
				2 (1)
				4 (2)
				8 (3)
				16 (4)
Table continues on next page				

Presentation				Priority value
Option 1	Option 2	Option 3	Option 4	(Bit weight)
Priority order			5 priority levels	
Pattern frame	Solid frame	Text	Indicator	
				32 (5)
				62 (1,2,3,4,5)
				56

Only one representation type per tagout instance can be used. It is not possible for the operator to change the representation type for a tagout instance.

One special type of symbol is part of the delivery. This symbol acts as a container for up to 8 subdrawings, which can be modified on a project specific base. See Figure 51 below.

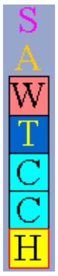


Figure 51: Tagout multi symbol presentation

The subsymbols can be found in the \sc\prog\graphicsEngine\Palette\01 - SA_Common \TagoutxSymbol1(2..8).sd directory. Before making modification, it is recommended to copy the subdrawings to the local \sc\application name\aplmod4\ Palette\01 - SA_Common directory. The sub-symbols will be visible according to the priority attribute value assigned to them. The priority value can be defined by using the Tagout Class Editor.

6.7.3.1 Tagout symbol dynamics

The quantifier is the value stored in the tagout indication process object value (OV) for the selected priority.

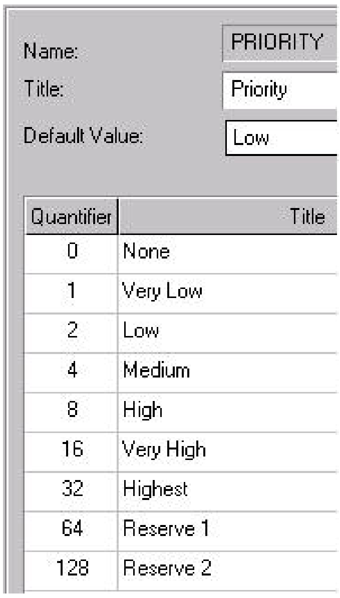


Figure 52: Priority quantifier

The priority plays the most important role when it comes to the tagout symbol shown in the displays.

For the simple type of symbols as, for example, Tagout with solid frame, the color selection is based on the currently assigned priority. If multiple tagouts are assigned, the highest priority will be used for the color presentation.

By using the Tagout indicator symbol, the priority (up to 5 priority levels) for all active tagouts for one switching device can be made visible.

6.7.3.2 Color setting tool

The tagout symbol colors representing the different priority states can be modified with the Color Setting Tool, see Figure 53 below.

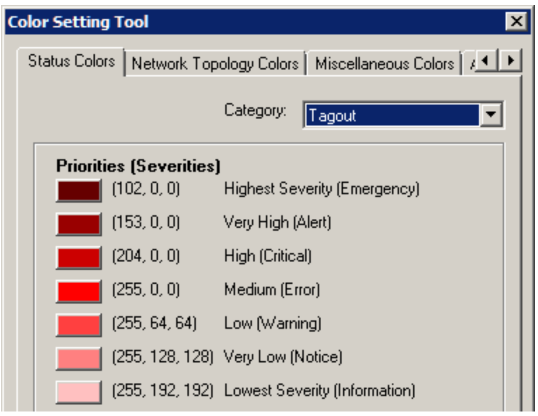


Figure 53: Tagout settings on Color Settings Tool

The following table shows the relation between priority quantity and Power Process Library color indexes.

Table 51: Color index reference to priority value

Data variable	Power Process Library	Class attribute	
Value	**Color index**	**Priority (default)**	**Quantity**
>=0 and <1	116	None (Lowest)	0
>=1 and <2	115	Very low	1
>=2 and <4	114	Low	2
>=4 and <8	113	Medium	4
>=8 and <16	112	High	8
>=16 and <32	111	Very high	16
>=32 and <64	110	Highest	32
>=64 and <128	117	Free (Reserve 1)	64
>=128 and <256	118	Free (Reserve 2)	128
-	119	Reserved 3	-

6.7.3.3 Tagout symbol installation with Display Builder

The tagout symbols are installed with Display Builder tool using the Object Browser.

There are three options for installing a tagout symbol:

1. As a group of symbols, switch device symbol together with the tagout symbol, with automatic creation and mapping of all used data variable. See Figure 54 below.

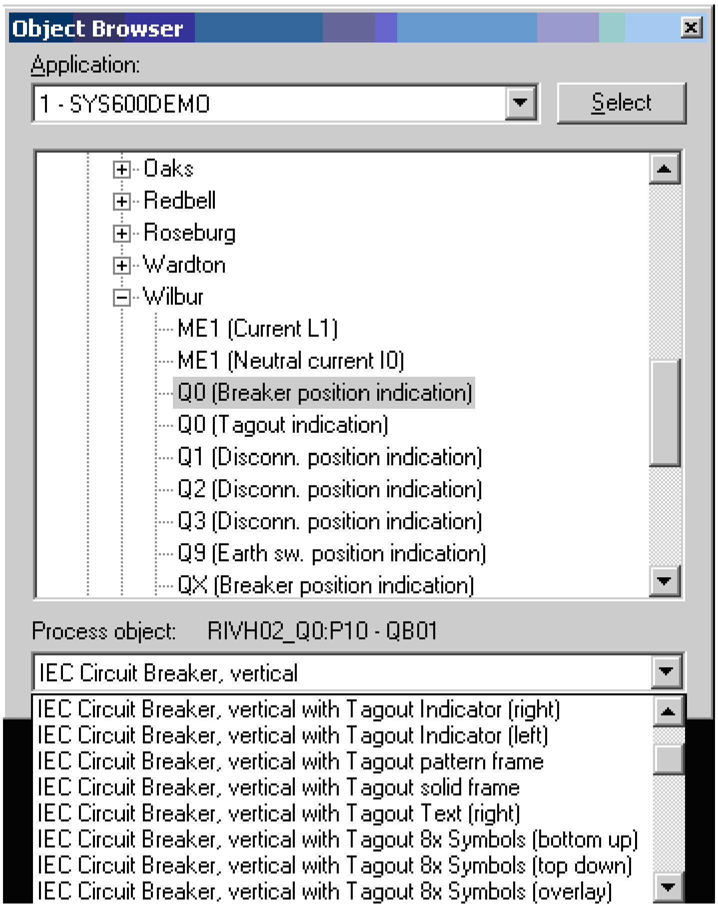


Figure 54: Tagout symbol group installation

2. Only the tagout symbol, with automatic creation and mapping of all used data variable. See Figure 55 below.

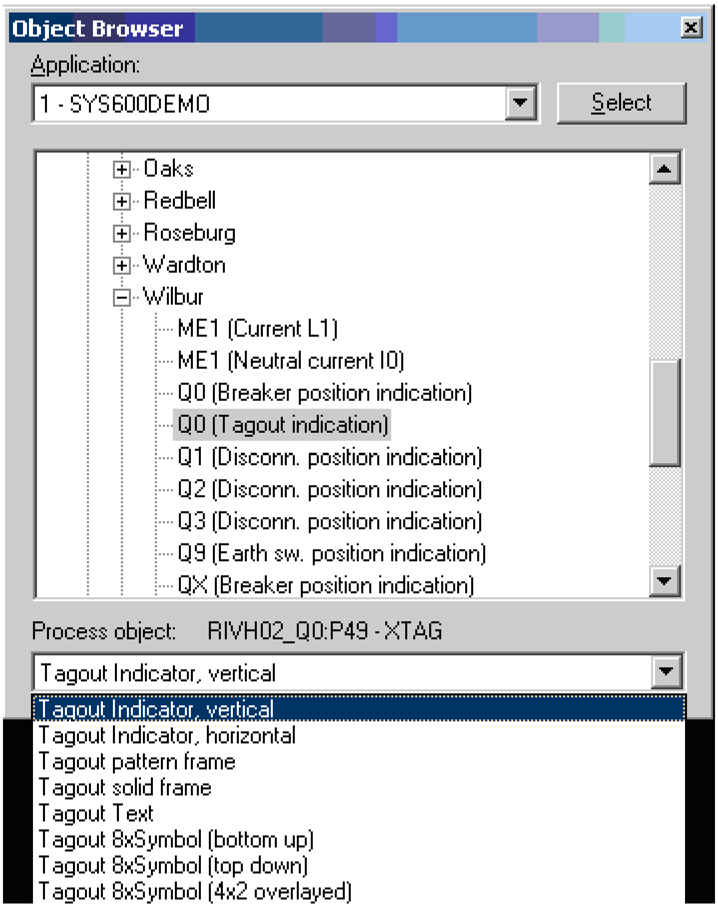


Figure 55: Tagout symbol single installation

3. From the 01 - SA_Common palette without automatic creation and mapping of all used data variable. See Figure 56 below.

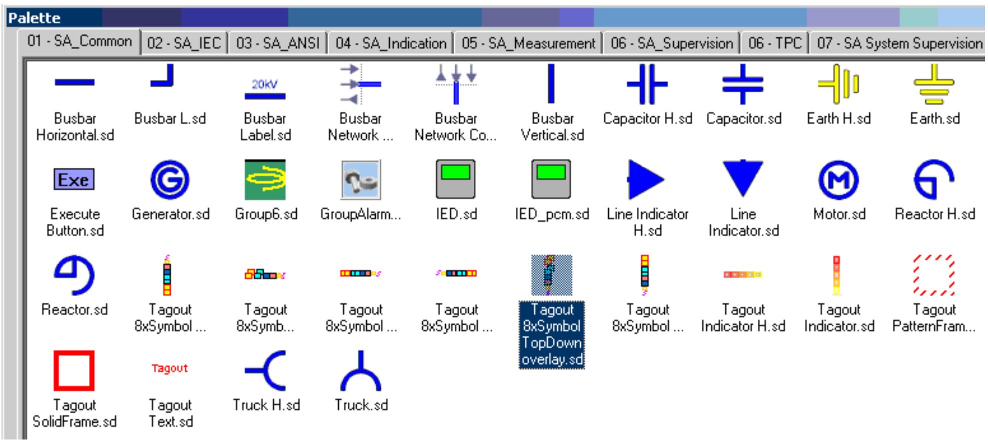


Figure 56: Tagout symbol selection from the palette

6.7.4 Tagout Control dialog

In the **Switch Control** dialog, the **Blocking** tab will be replaced with a **Tagout** tab if the active tagout class contains action method attributes. The **Tagout** tab is only visible for objects where the tagout function has been enabled with the Standard Function Configuration Tool.

The properties and view for the **Tagout** tab can be adjusted with the Tagout Class Editor.

The **Tagout** tab is used for presenting all active tagouts for the selected switching device, see Figure 57 below.

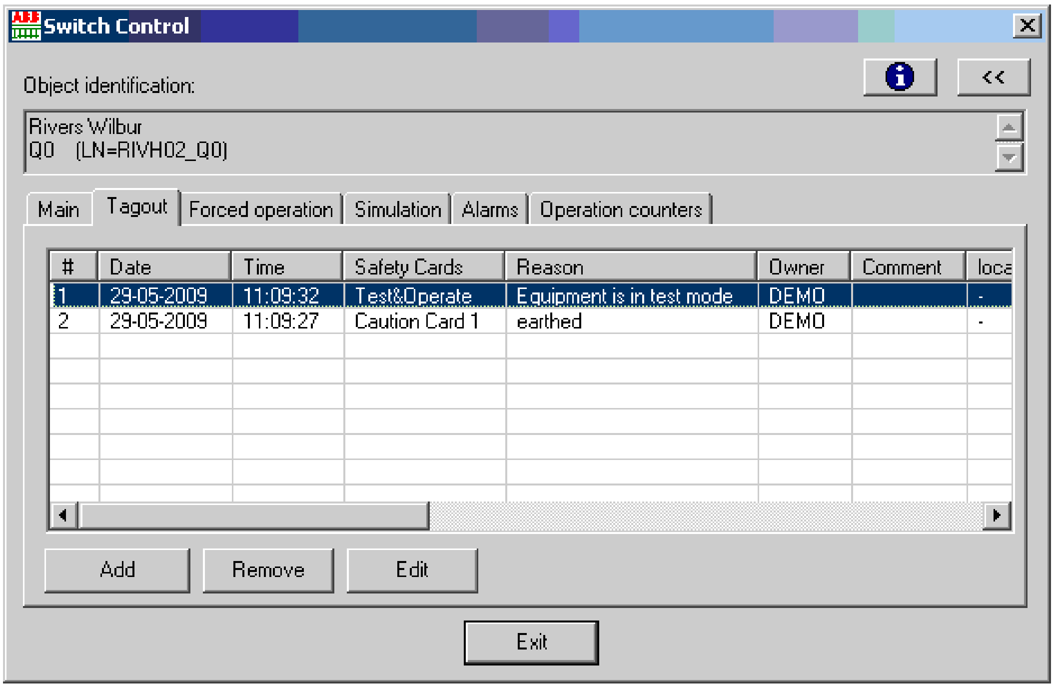


Figure 57: Switch Control dialog with active tagouts

The tab is also the only place where the user can add, edit and remove tagouts by using the **Add**, **Remove** and **Edit** buttons. Active tagouts can only be edited and removed by the owner of the tagout. The owner name is fetched from the currently logged on MicroSCADA user.

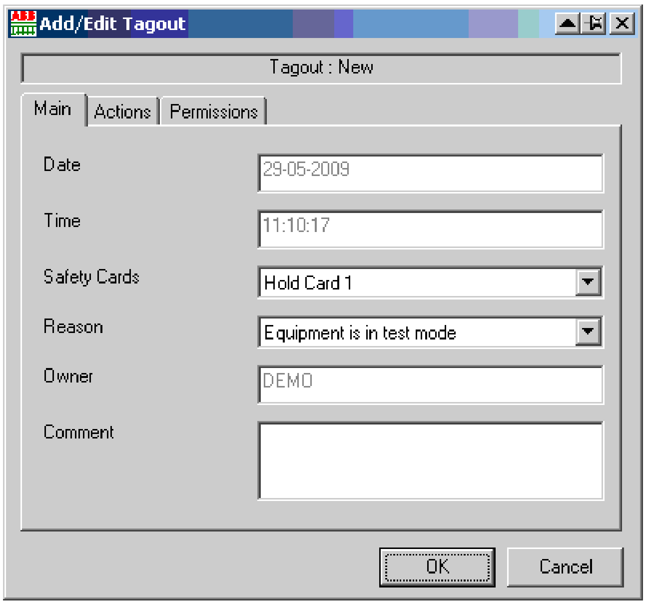


Figure 58: Add/Edit Tagout dialog

The dialog for adding and editing tagouts can contain several tabs, see Figure 58 above. The title and position of these tabs can be adjusted with the Tagout Class Editor.

In case of an error in communication between **Switch Control** dialog and the tagout class database, the user will be informed with and error notification dialog, see Figure 59 below.

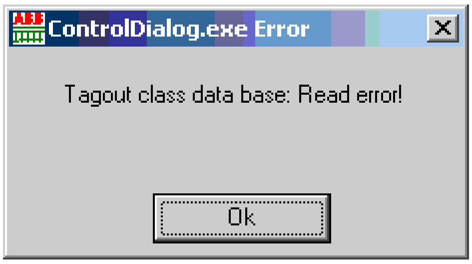


Figure 59: Switch Control dialog error notification

The error message text can be modified with the Tagout Class Editor.

6.7.5 Tagout action methods

The standard blocking attributes need a special treatment when used within the Tagout function. To highlight this, the term Action_Methods is be used for these attributes.

Table 52: Tagout Action Methods

Method name	Purpose	SYS600 attribute notation
Local lock	Block control from SA system	UB update blocking commands
Lock remote	Block control form remote system	Depends on remote connection. For COM500i, commands received from an NCC will be blocked and a negative action confirmation will be sent back to the NCC.
Inhibit local	Update block from process	UB update blocking for indication. Reset will force a data update to the process level device
Inhibit remote	Update block upwards to NCC	Depends on the remote connection. For COM500i, sending an update to NCC will be blocked.
Standard blocking attributes		
Inhibit Alarm	Blocks all alarm activation from alarming objects within the current group	AB, alarm blocking
Inhibit Event	Blocks all alarm events for objects with enabled history function within the current group	HB, history blocking
Inhibit Printout	Blocks all printouts for objects with enabled printout function within the current group	PB, printout blocking
Inhibit Processing	Blocks all further processing for objects with enabled reprocessing function within the current group	XB, activation blocking

Blocking attributes, which have been activated or deactivated with the Tagout function, are not enabled in other standard blocking attribute views, such as the **Blocking** list and the **Blocking** tab inside the **Switch Control** dialog. The blocking state will be visible but it cannot be changed.

The set of process objects included in the object search, depends on the selection made in the Tagout Class Editor, see Section 6.7.6.8.

Every change in any of the action method attributes will launch the user defined command procedure SAGR_TAGOUT_AMUD.

The following parameters will be provided:

- Logical Name
- Main indexes (Index 10 for Indication, Index xx for commands, depending on command type)
- Action Method attribute name, e.g. LOCALLOCK, REMOTEINHIBIT
- New value for changed attribute

This user defined command procedure can be used, for project specific code, to perform actions related to process level devices, either to activate some locking inside an IED or to inform a possible external gateway, for example, COM6xx or COM581, about tagout changes related to their function.

6.7.6 Tagout Class Editor

Tagout Class Editor is a tool for modifying the properties of each tagout class and their attributes.

6.7.6.1 Basic tagout class attribute settings

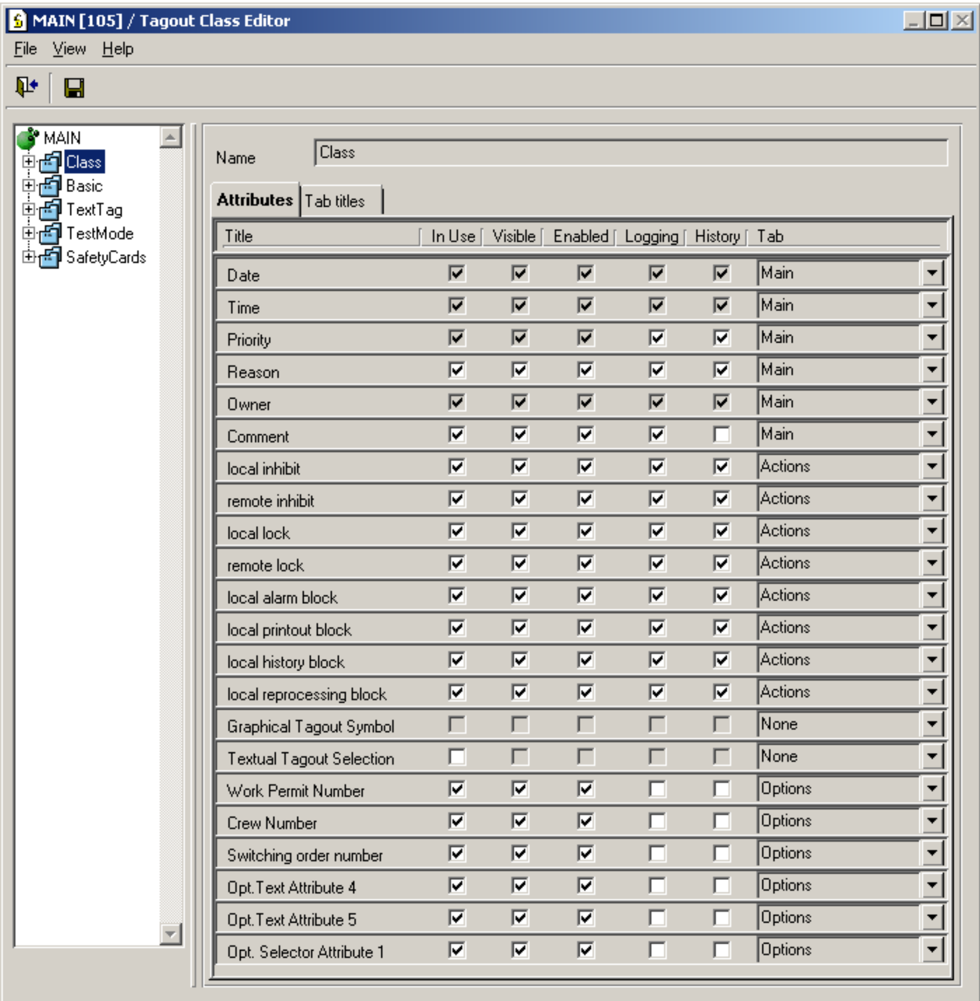


Figure 60: Tagout Class editor, Class root page, Attributes tab

The following basic settings for tagout class attributes can be done in the class root page (see Figure 60 above):

- Taking an attribute in use or out of use.
  Attributes set to Not in use will be ignored in all Tagout function components
- Setting an attribute as **Visible** in **List** and **Tab** view
- Enabling changing attribute value in **add/edit** dialog
- Releasing attribute changes for logging and event generation (history) purposes
- Assigning an attribute to any existing tab of the **add/edit** dialog

The following figures show the different behavior for attributes depending on the settings described above.

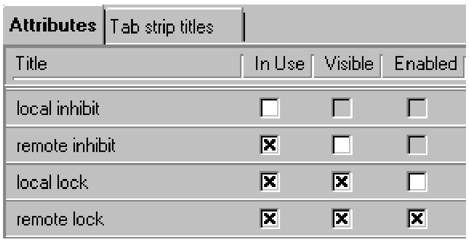


Figure 61: Attribute behavior settings

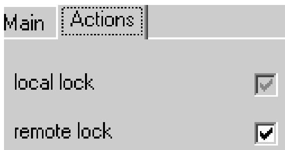


Figure 62: Attribute presentation in Switch Control dialog

The first attribute **local inhibit** is out of use and will not be used at all.

The second attribute **remote inhibit** is in use but not visible anywhere. The specified default value will be used when a tagout is added.

The third attribute **local lock** is in use and visible. The specified default value will be used when a tagout is added.

The fourth attribute **remote lock** is in use, visible and enabled. The user can change the specified default value before a new tagout is added.

6.7.6.2 Tab settings

With the **Tab titles** the user can:

- add/remove/edit the tab titles for the **Add/Edit Tagout** dialog,
- define the tab position within the **Add/Edit Tagout** dialog.

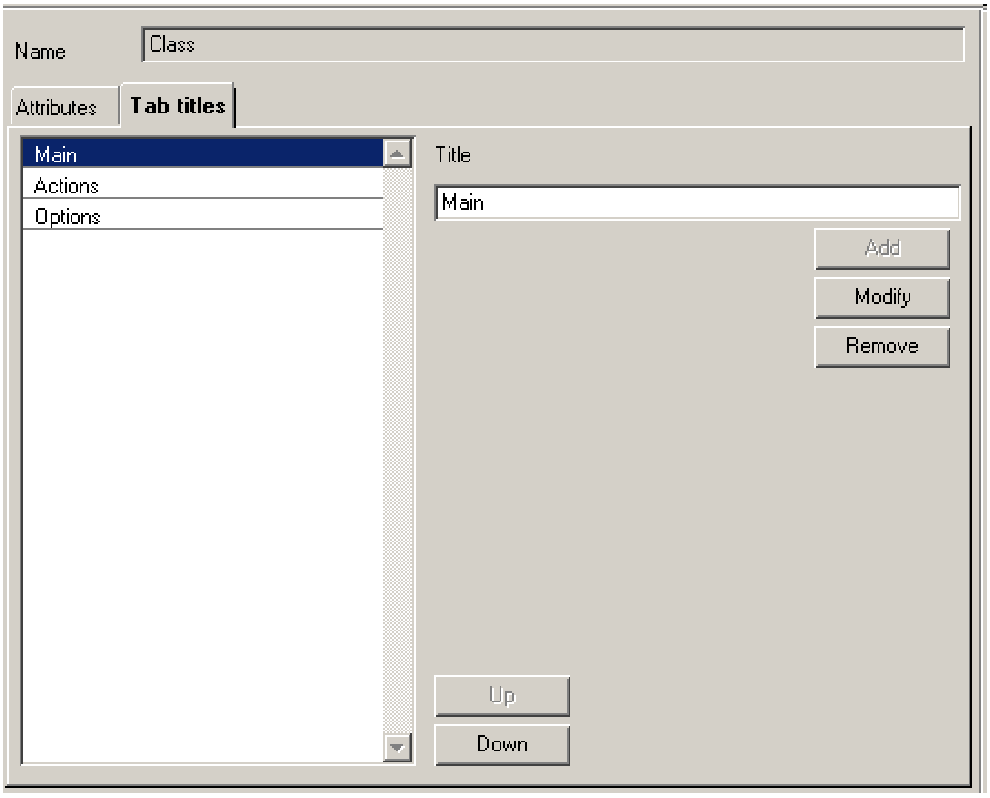


Figure 63: Handling Add/Edit Tagout dialog tabs

Tabs can only be removed when there is no class attribute assigned to them.

6.7.6.3 Attribute position in views

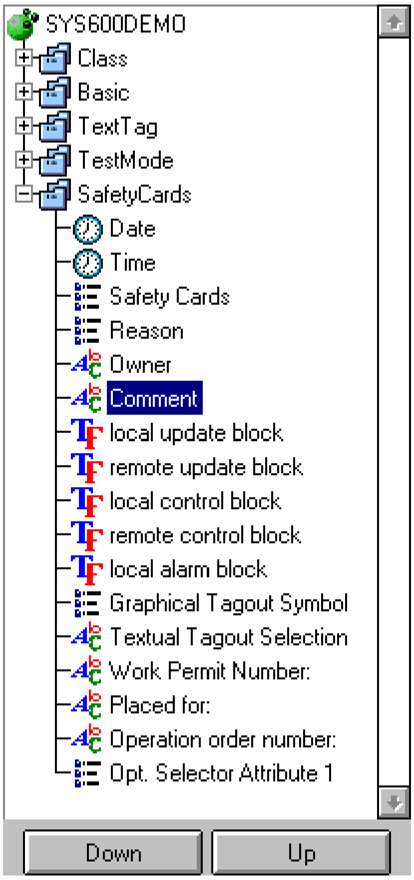


Figure 64: Changing attribute position

The absolute attribute position within the Switch Control dialog Tagout List can be changed with the **Down** and **Up** buttons.

The relative attribute position in the assigned tab of the **Add/Edit Tagout** dialog can also be changed with the **Down** and **Up** buttons.

6.7.6.4 Attribute type Selector

For attributes of the type Selector, the following properties can be modified:

- The attribute title shown in the **Switch Control** dialog's **Tagout** List view and in the **Add/ Edit Tagout** dialog.
- The default value used for the **add/edit** dialog when a new tagout is added

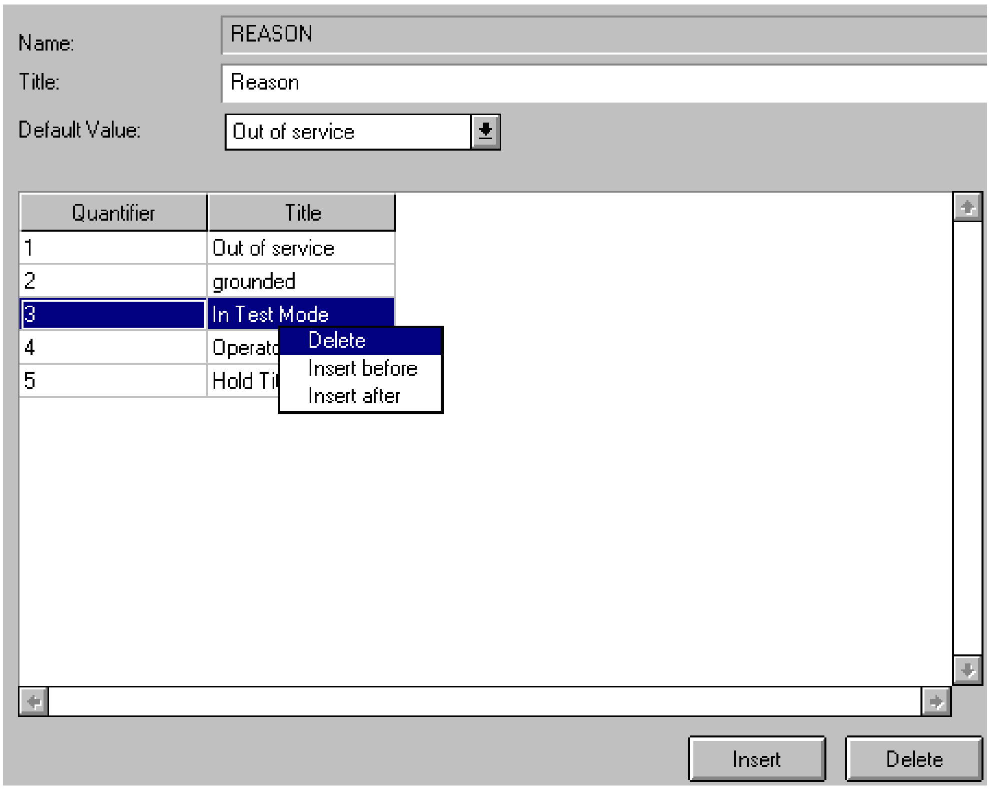


Figure 65: Editing the Selector attribute properties

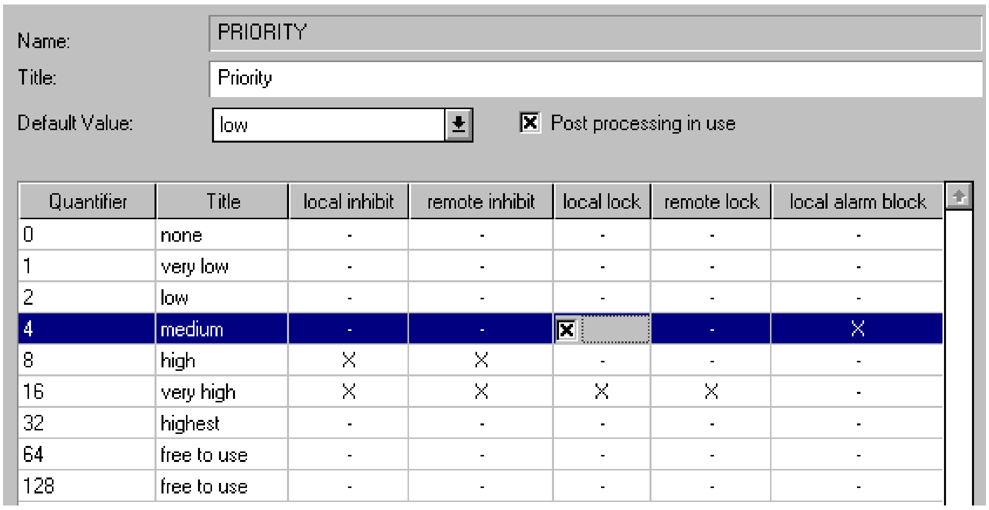
The quantifier is the value that is written into the process object database for the selected title. In the current version, the quantifier will be used only for the priority attribute.

It is possible to add, remove or edit the attribute selector titles and quantifier. However, at least one selector title must remain.

With the **Insert** button, a new row can be added above the selected row. With the context menu that opens when a row is right-clicked, a new row can be added above or below the selected row.

A special treatment for post processing activities is needed for the priority attribute. Post processing activities are attribute changes which depend on priority value selection.

The post processing selection, shown in Figure 66 below, is only available for classes containing action methods. Only action methods that are a part of the selected class and in use can be used in post processing.



Quantifier	Title	local inhibit	remote inhibit	local lock	remote lock	local alarm block
0	none	-	-	-	-	-
1	very low	-	-	-	-	-
2	low	-	-	-	-	-
4	medium	-	-	X	-	X
8	high	X	X	-	-	-
16	very high	X	X	X	X	-
32	highest	-	-	-	-	-
64	free to use	-	-	-	-	-
128	free to use	-	-	-	-	-

Figure 66: Post processing priority editing

The action method attribute values are set with the **Add/Edit Tagout** dialog after the priority value selection has been made. Below, Figure 67 shows **Add/Edit Tagout** dialog for the rows handled above in Figure 66.

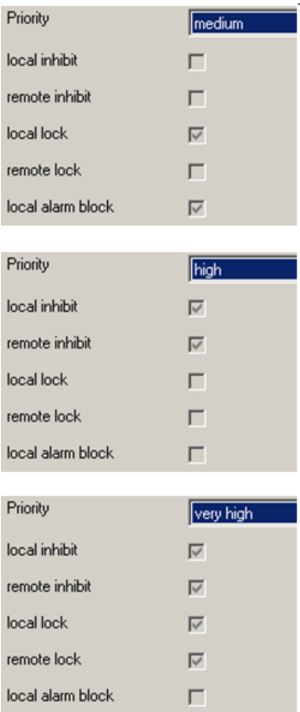


Figure 67: Priority post processing in Add/Edit Tagout dialog

6.7.6.5 Attribute type Text

For attributes of type Text, the following properties can be modified:

- the attribute title shown in the **Switch Control** Dialog's **Tagout** List view and in the **Add/ Edit Tagout** dialog,
- the default value used for the **Add/Edit Tagout** dialog when a new tagout is added,
- the minimum and maximum length.

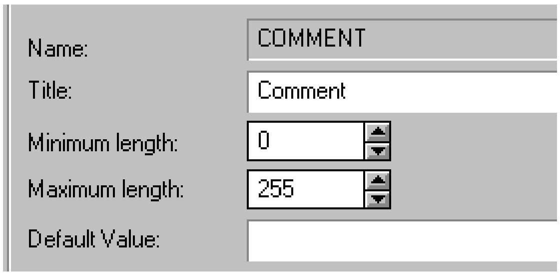


Figure 68: Editing attribute properties for type Text

6.7.6.6 Attribute type Boolean

For attributes of type Boolean, the following properties can be modified:

- The attribute title shown in the **Switch Control** Dialog's **Tagout** List view and in the **Add/ Edit Tagout** dialog
- The default value used for the **Add/Edit Tagout** dialog when a new tagout is added

Figure 69: Editing attribute properties for type Boolean

6.7.6.7 Overall settings

Class activation
The activation of one class out of the list of existing classes can be done in the **Main** page of the **Activate Class** tab, see Figure 70 below.

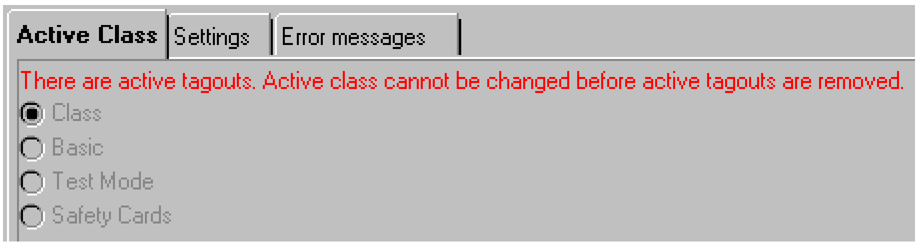


Figure 70: Class activation

The active class can only be changed if there are no active tagouts at that moment. Once a tagout has been activated, the properties from the active tagout class will be made read only. This prevents any mismatch in properties between active tagouts and active tagout class.

6.7.6.8 Action Method settings

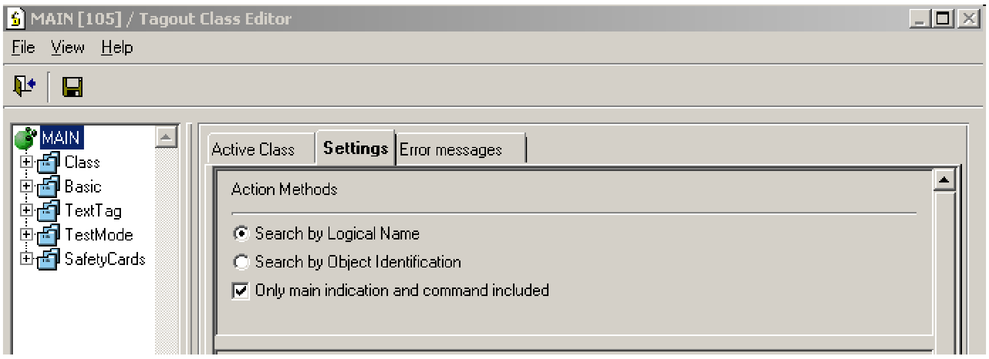


Figure 71: Action Method settings

The following settings are available on the **Settings** tab (see also Section 6.7.6.8):

- Only main indication and command objects included
  Index 10 (and 11 for three-state switch) for the status indication
  Main object command indexes for switch device control (Index range depends on command type)

- Filter on Object Identification (OI)
  or
- Filter on Logical Name (LN).

Blockings will be set for all process objects based on the defined filter condition.

Excluded from object search command are all internal objects, such as:

- Selected on monitor
- Tagout indication object (Index 49)
- Tagout event & logging object (Index 50)

6.7.6.9 History handling settings

For the event (History) handling, the following settings are available on the **Settings** tab (see Figure 72):

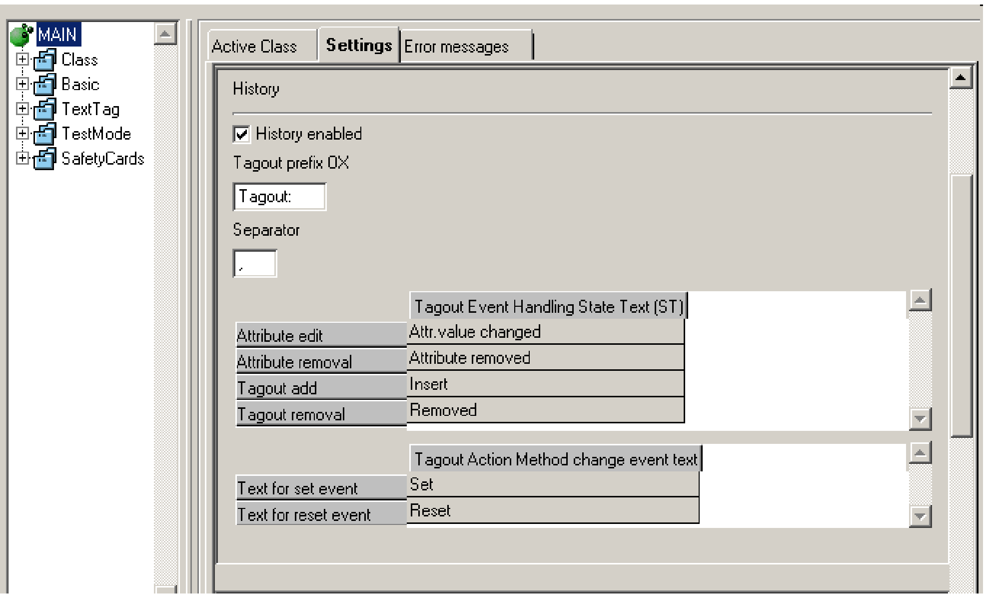


Figure 72: History Handling settings

By using the **History enabled** check box, the event generation can be taken out of use for all tagout activities. This setting will overrule the history handling settings made for class attributes.

6.7.6.10 Activity logging settings

For the Logging function the following settings are available on the **Settings** tab (see Figure 73):

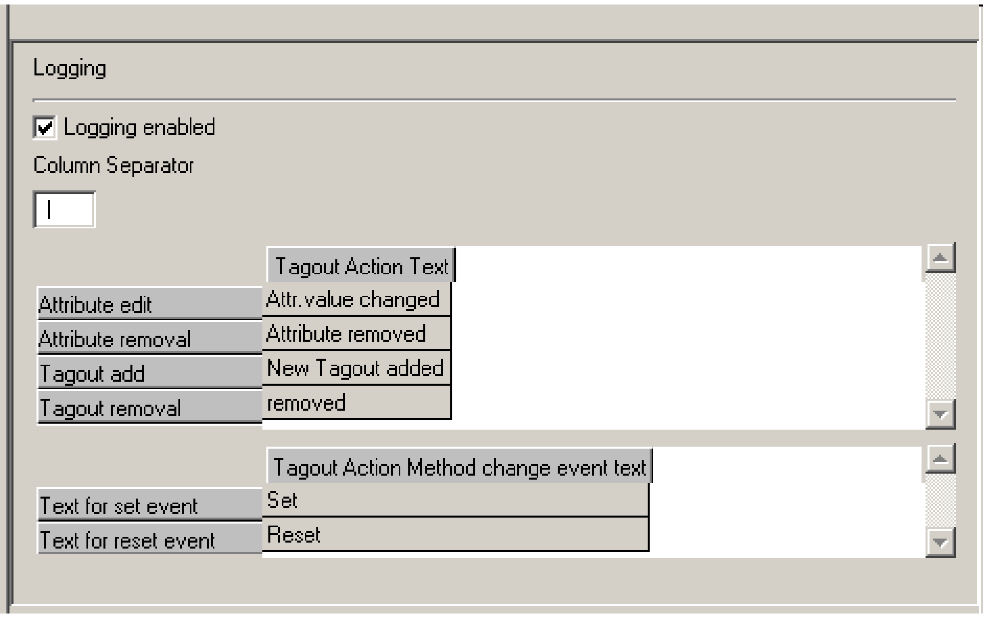


Figure 73: Activity logging settings

Using the **Logging enabled** check box the logging function can be taken out of use for all tagout activities. This setting will overrule the logging handling settings made for class attributes.

6.7.7 Viewing Tagouts

Tagout Lists opens from the **View** menu of the Tagout Class Editor. The Active Tagouts tab of Tagout List presents all active tagouts, see Figure 74 below. Tagout List can be added as an item into the existing menu structure. Also, launching via a toolbar button or icon from any display will be supported. The Tagout List can only be used to view active tagouts.

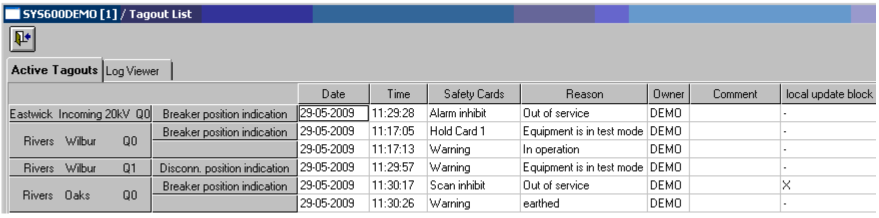


Figure 74: Tagout List

The Tagout Log Viewer is integrated in the **Tagout List** as a separate **Log Viewer** tab.

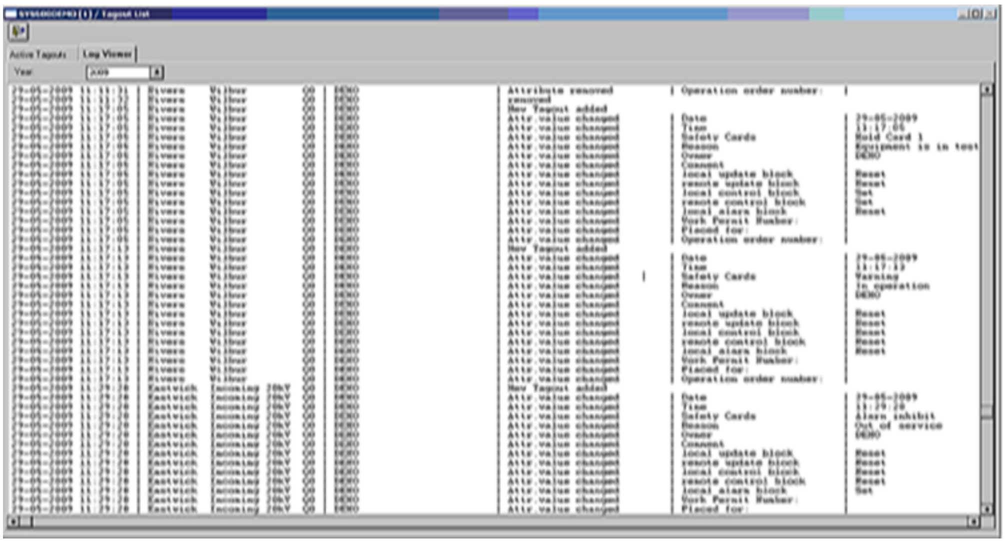


Figure 75: Tagout Log Viewer in Tagout List

The **Tagout List** can be set to be opened from the menu as a Visual SCIL dialog. The following item properties should be used for the set up in the **Specify VSCIL Tool** dialog (see also Figure 76 below):

- File: C:\sc\Stool\AplBuild\TOEDIT.VSO
- Object name: TOList
- Tool Appearance: Tool size: Size 1 (960x720)

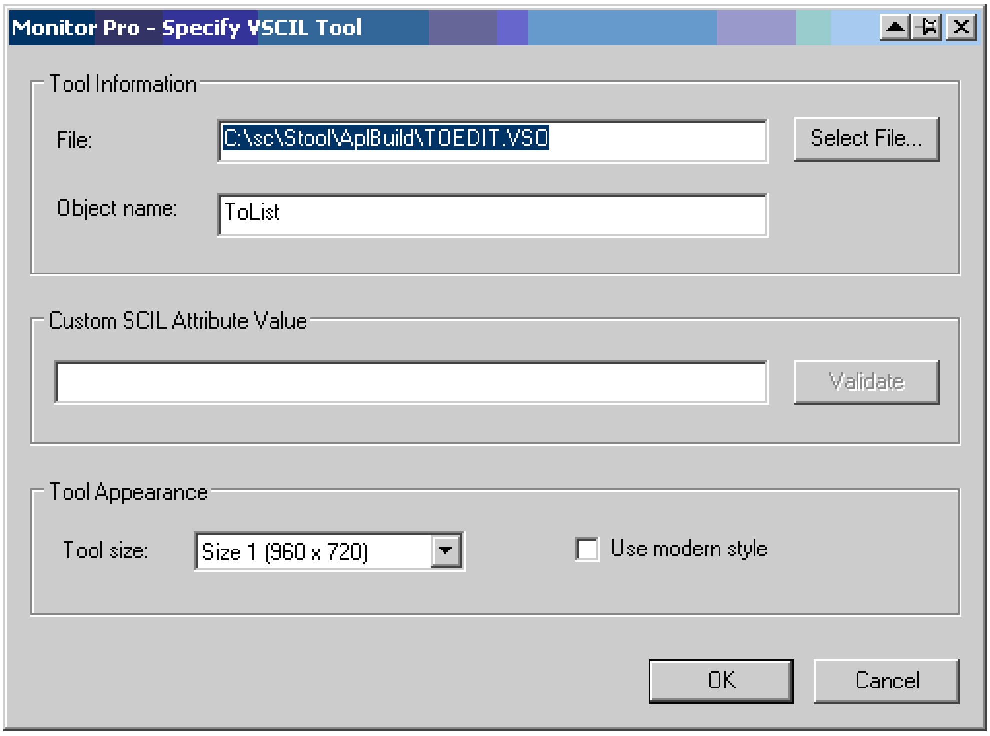


Figure 76: Settings for Tagout List to open as Open Visual SCIL Tool

To make the **Tagout List** open as a Visual SCIL dialog from a display, the following settings should be used in the **Tool Launcher Settings** dialog (see Figure 77 below):

- The **Visual SCIL dialog** radio button should be selected
- **VSO file**: C:\sc\Stool\AplBuild\TOEDIT.VSO
- **Visual SCIL object name**: TOList

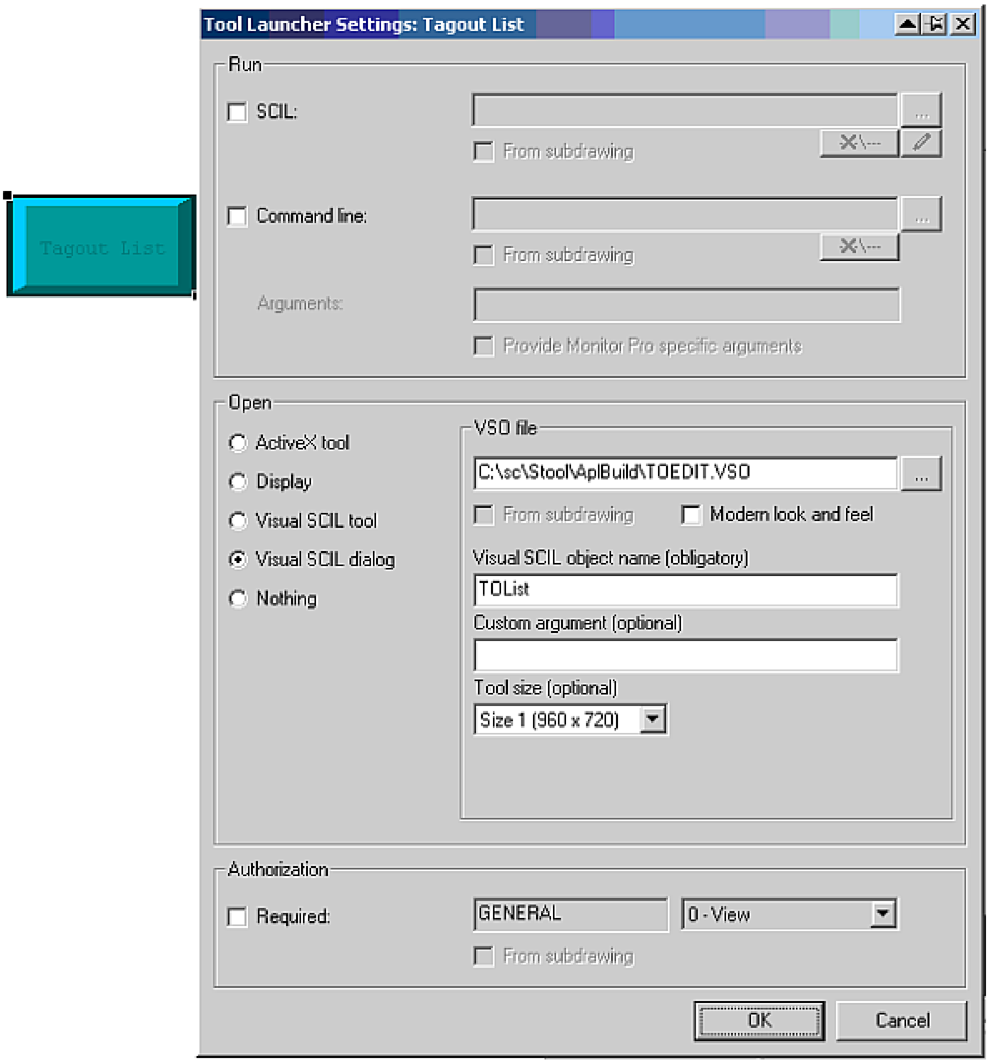


Figure 77: Launcher Settings for Tagout List

The Blocking Display can also be used to present all active blockings including the blockings activated by the Tagout function, see Figure 78 below. However, it is not possible to modify the blocking attributes set by the Tagout function in the Blocking Display.

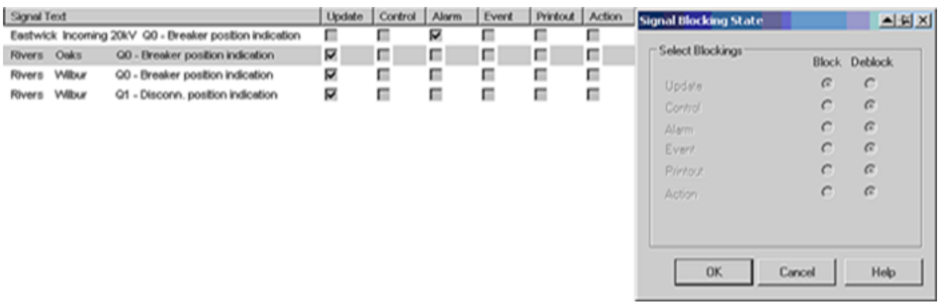


Figure 78: Blocking Display with active blockings

The Event Display can be used to list all historical tagout activities. The Tagout filter function in the **Filter Settings** dialog can be used for listing only tagout related events in Event Display.

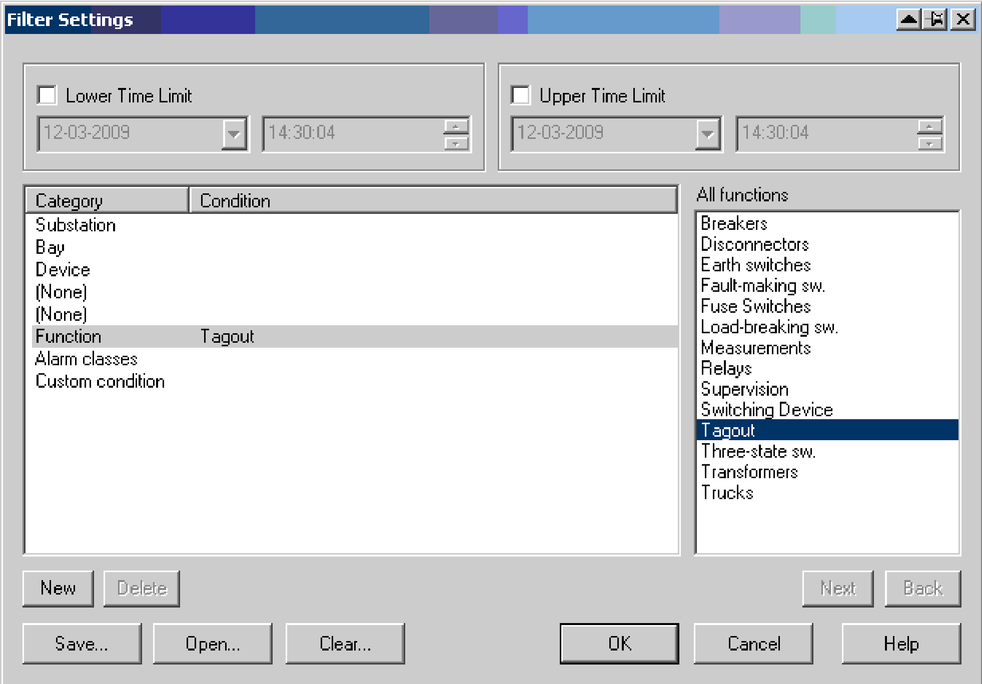


Figure 79: Event Display Filter Settings dialog

6.7.8 Localization

Tagout Class Editor dialog items and notification messages can be translated with the Text Translation Tool.

Tagout Class attributes are part of the Tagout Class and need to be translated with the **Tagout Class Editor**. The tagout Class attributes include, for example, attribute titles, which are used for the **Switch Control** Dialog **Tagout** tab and **Tagout List**.

Some basic text, for example the button caption text for the **Switch Control** Dialog **Tagout** Tab, needs to be translated with third party software. This text is located in the Control Dialog resource file ControlDialog2_EN.dll. See Table 181.

6.7.9 Application engineering information

6.7.9.1 Files

The table below lists all the Tagout function related files and their functionality.

Table 53: Tagout function files

File	Functionality	Path
BGU_TAGOUT.SDB	Class database template	/SA_LIB/DEFAULTS/TAGOUT
ToEdit.vso	Class editor	/STOOL/APLBUILD
Tagout 8xSymbol.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout 8xSymbol BottomUp.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common

Table continues on next page

File	Functionality	Path
Tagout 8xSymbol LeftRight.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout 8xSymbol RightLeft.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout 8xSymbol TopDown overlay.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout 8xSymbol TopDown.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout Indicator H.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout Indicator.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout PatternFrame.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout SolidFrame.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Tagout Text.sd		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
tagoutxSymbol1(..8).sd *		/ PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common

6.7.9.2 Command procedures

At the first Tagout Class Editor usage, the following command procedures will be created:

Table 54: Tagout function related command procedures

Name	Functionality
SAGR_TAGOUT	SCIL interface to tagout Class Database
SAGR_TAGOUT_AM	Action Method (blocking) handling
SAGR_TAGOUT_AM_EX	Interface with COM500i
SAGR_TAGOUT_AM_UD	For user defined action on Action Method activation and deactivation
SAGR_TAGOUT_HIST	Logging handling
SAGR_TAGOUT_LOG	Event and History handling

6.7.9.3 Process objects

During the installation with the Standard Function Installation tool, two process objects within the switching device logical name group will be created:

Table 55: Process object attributes related to Tagout function

Index	Obj.	Process object	Remarks	Group Identifier
49	DI	Tagout Indication	Internal	FPXTAGXIND
50	ID	Tagout History activation	Internal	FPXTAGXEVT

6.7.9.4 Event handling object

Table 56: Event handling object for Tagout event text

Name	Functionality
SAGR_TAGOUT_HIST	Event state text for tagout activities

6.8 Tap changer

6.8.1 Standard function installation

This section describes the installation of the Tap changer standard function from the Power Process Library. The standard function for the Tap changer is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Process/Tap Changer). This standard function is configured by using configuration tools, such as the Object Navigator.

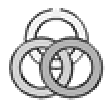
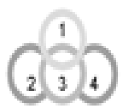
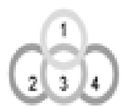
6.8.1.1 Symbol installation

The Power Process symbol for the tap changer standard function is installed by using the following Display Builder functions:

1. Object Browser
2. Palette

For further information, see the SYS600 Process Display Design manual.

Table 57: Power Process symbol for tap changer standard function

File name		Representation	Symbol	
Horizontal	Vertical		Horizontal	Vertical
02 -SA_ANSI Transformer2wH.sd	Transformer2w.sd	ANSI		
02 -SA_IEC\ Transformer2wH.sd	Transformer2w.sd	IEC		
02 -SA_ANSI\ Transformer3wH.sd	Transformer3w.sd	ANSI		
02 -SA_IEC\ Transformer3wH.sd	Transformer3w.sd	IEC		
02 -SA_ANSI\ Transformer4wH.sd	Transformer4w.sd	ANSI		
02 -SA_IEC\ Transformer4wH.sd	Transformer4w.sd	IEC		

6.8.2 Standard function configuration

6.8.2.1 Configuration with tools

The Tap changer standard function has the following attributes to be configured with the Object Navigator:

Table 58: Tap changer attributes configured with the tools

Attribute	Meaning	
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
MIN_POSITION	Minimum position of tap changer	1
MAX_POSITION	Minimum position of tap changer	19
P_OBJECT_LN	Logical name of database process object	-
STATION_TYPE	The type of control device & protocol	ANSI
CMD_PARAMETER	A selector for selecting of control method. This attribute has meaning only if IEC 60870-5-101/104 or DNP 3.0 protocol is selected, otherwise this attribute is disabled.	N/A
OUTPUT_STATUS	Process objects for output status with DNP 3.0 protocol.	N/A
AUTO_MANUAL_IND_TYPE	The type of the auto/manual signal	01=manual, 10=auto (DB)
AUTHORIZATION_GROUP	The name of the authorization group	MV_CONTROL
TRANSFORMER_TYPE	The number of windings	2-winding
OPC_ITEM_PREFIX	Specifies the OPC Item Prefix to as IEC 61850 instance information.	IEC61850 Subnetwork.IED1.LD1
OPC_LN_INSTANCES	Specifies the OPC Logical Node names used as IEC 61850 Instance information.	VECTOR
OPERATOR_PLACE_HANDLING	Specifies the operator place (local/Station/NCC) handling of the device	None

For more detailed description on the configurable attributes, see Section 6.15.

6.8.2.2 Configuration of process objects

Depending on the configuration of the Tap changer standard function, the tools create a certain set of process objects into the database. Process objects that will be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

When process objects are created, default scales are assigned to the analog input signals meant to be connected to the process (indices 24,25 and 26).

Table 59: Process objects created for tap changer

Index	Explanation	Purposes
10	Tap position indication	Used for indication of the tap changer position.
11	Tap ch. auto/ manual ind.	Used for indication of the transformer auto manual state.
12	Tap ch. single/ parallel ind.	Used for indication of the transformer single / parallel state.
13	Tap ch. master/slave ind.	Used for indication of the transformer Master/Slave state.
14	Tap ch. manual cmd	Used for sending "manual" command to the control unit.
15	Tap ch. auto cmd	Used for sending "auto" command to the control unit.
16	Tap ch. raise cmd	Used for sending "raise" command to the control unit.
17	Tap ch. lower cmd	Used for sending "lower" command to the control unit.
18	Tap ch. single/ parallel cmd	Used for sending "single" or "parallel" command to the control unit.
19	Tap ch. master/slave command	Used for sending a "master" or "slave" command to the control unit.
20	Transformer ext. blocking	Used for indication of the transformer blocking signal.
24	Voltage	Used for indication of the voltage.
25	Reference voltage ind.	Used for indication of the reference voltage.
26	Reference voltage command	Used for sending the reference voltage to the control unit
28	Set tap position	Used for setting tap directy to a certain position
29	Tap changer operation counter	In internal use, used for counting the number of tap changer step changes.
30	Selected on monitor	In internal use, tags the selection of the object made in different pictures or monitors.
70	Local/ Remote-switch	Used for Device level Local/Remote-switch indication
71	Operator place-switch	Used for device level operator place (Station/NCC) indication
72	Operator place-switch	Used for device level operator place (Station/NCC) command
249	Transformer routing prim. info	For network topology coloring purposes.
Table continues on next page		

250	Transformer predefined color	For network topology coloring purposes.
253	Transformer fict. pos. ind.	For network topology coloring purposes.
255	Transformer predefined color	For network topology coloring purposes.

6.8.2.3 Example of a tap changer configuration

Table 60: Example configuration of a tap changer

Attributes	Value
STATION_NAME	Eastwick
BAY_NAME	Transformer 1
DEVICE_NAME	TR1
MIN_POSIT	1
MAX_POSIT	19
P_OBJECT_LN	ESTH03_T1
STATION_TYPE	SPA
AUTHORIZATION_GROUP	MV_CONTROL
TRANSFORMER_TYPE	2-winding

6.8.3 Application engineering information

6.8.3.1 Structure of the tap changer standard function

This section describes the structure of the Tap changer standard function. All subdrawing files, form pictures, help and other text files, as well as database objects are included. The Tap changer is a part of the standard functions of Power Process Library and has the directory /SA_LIB/BASE/BBONE and the standard Power Process Library subdirectories INST, LANG0 and USE.

6.8.3.2 Files

The table below lists all Tap changer standard function related files and their functionality.

Table 61: Tap changer standard function related files

File	Functionality	Path
Transformer2w.sd	SA ANSI symbol for two winding transformer	/PROG/GRAPHICSENGINE/PALETTE/03 - SA_ANSI
Transformer3w.sd	SA ANSI symbol for three winding transformer	/PROG/GRAPHICSENGINE/PALETTE/03 - SA_ANSI
Transformer2w.sd	SA IEC symbol for two winding transformer	/PROG/GRAPHICSENGINE/PALETTE/03 - SA_IEC
Transformer3w.sd	SA IEC symbol for three winding transformer	/PROG/GRAPHICSENGINE/PALETTE/02 - SA_IEC
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

6.8.3.3 Language text files

The following Text Translation Tool compatible text files are used by the Tap changer standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 62: Text Translation Tool compatible text files used by the tap changer standard function

File	Functionality
SAI_TR2R.TXT	Text file for the database creation of the standard function
SAI_TR2S.TXT	Text file for the database creation of the standard function
SAI_HVT2.TXT	Text file for the database creation of the standard function network topology coloring
SAI_HVT3.TXT	Text file for the database creation of the standard function network topology coloring

6.8.3.4 Help text file

The path to the Tap changer help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 63: Tap changer help text file

File	Functionality
SAI_TR2.HLP	Standard function installation help file

6.8.3.5 Configuration files

The following configuration files are used by the Tap changer standard function. The path is / SA_LIB/BASE/BBONE/INST.

Table 64: Configuration files used by the tap changer standard function

File	Functionality
SAI_TRI_2.DAT	Contains the configuration data for tap changer when it is created modified or deleted by configuration tools.
SAI_TRI_2.POT	Contains the process object definitions for the Object Navigator.
SAI_HVT23.POT	Contains the process object network topology coloring definitions for the Object Navigator.
SAI_TRAIDX.TXT	Contains the process object network topology coloring definitions for the Object Navigator.
SAI_HVT2.TPL	Contains the process object network topology coloring template definitions for the Object Navigator.
SAI_HVT3.TPL	Contains the process object network topology coloring template definitions for the Object Navigator.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for installation tool.

6.8.3.6 Other text file

The following text file is used by the Tap changer. The path is /SA_LIB/BASE/BBONE/USE.

Table 65: Text file used by the tap changer

File	Functionality
BGU_AUTH.TXT	Contains the SCIL code for station authority check. The code is executed by the dialog of the tap changer.

6.8.3.7 Command procedures

At the first installation, the Tap changer creates the command procedure BGU_UPDATE, which updates process objects when they are update-deblocked. This feature is similar to the SYS600 Blocking Display function.

For ANSI type of station, the command procedure BGU_AI2DB will be created while installing. The command procedure BGU_AI2DB updates the position indication double binary from the analog input. The command procedure BGU_TCCO is created for updating the operation counter object.

Table 66: Command procedures used by the tap changer

BGU_TRAFO Performs the control operations	
Function	**Description**
AUTOMATIC_SELECT	Performs select operation for automatic command on transformer standard function.
AUTOMATIC_EXECUTE	Performs execute operation for automatic command on transformer standard function. Uses text file /SA_LIB/BASE/BBONE/USE/BGU_SW_EXE.TXT
AUTOMATIC_CANCEL	Performs cancel operation for automatic command on transformer standard function.
LOWER_SELECT	Performs select operation for lower command on transformer standard function.
LOWER_EXECUTE	Performs execute operation for lower command on transformer standard function.
LOWER_CANCEL	Performs cancel operation for lower command on transformer standard function.
MANUAL_SELECT	Performs select operation for manual command on transformer standard function.
MANUAL_EXECUTE	Performs execute operation for manual command on transformer standard function.
MANUAL_CANCEL	Performs cancel operation for manual command on transformer standard function.
PARALLEL_SELECT	Performs select operation for parallel command on transformer standard function.
PARALLEL_EXECUTE	Performs execute operation for parallel command on transformer standard function.
PARALLEL_CANCEL	Performs cancel operation for parallel command on transformer standard function.
RAISE_SELECT	Performs select operation for raise command on transformer standard function.
RAISE_EXECUTE	Performs execute operation for raise command on transformer standard function.
RAISE_CANCEL	Performs cancel operation for raise command on transformer standard function.
SINGLE_SELECT	Performs select operation for single command on transformer standard function.
SINGLE_EXECUTE	Performs execute operation for single command on transformer standard function.
SINGLE_CANCEL	Performs cancel operation for single command on transformer standard function.
SELECT_ON_MONITOR	Sets transformer standard function to selected on monitor state.
UNSELECT_ON_MONITOR	Removes selected on monitor state from transformer standard function.
Table continues on next page	

BGU_TRAFO Performs the control operations	
Function	**Description**
OBJECT_CONF	Gets object configuration for transformer standard function.
GET_STATES	Gets process object states of transformer standard function.
GET_MESSAGES	Gets object messages of transformer standard function.
BGU_BLOCK Performs the blocking operations	
Function	Description
GET_BLOCKING_STATE	Gets blocking state for objects that belong to standard function.
SET_BLOCKING_STATE	Sets blocking state for objects that belong to standard function. If standard function is bay, applies to all objects that belong to bay. If standard function is station, applies to all objects that belong to a station.
GET_ENABLED_BLOCKINGS	Disables blocking possibility in some standard functions. For example, control blocking cannot be used with a measurement and alarm indicator standard functions.

6.8.3.8 Process objects

The following process objects will be created depending on the configuration of the Tap changer.

Table 67: Process objects for Generic tap changer

Index	Object type	Meaning of Process object	Remarks	Group identifier
10	AI	Tap changer position	-	FPTAVRIPOS
11	BI/DB	Tap changer manual/auto -ind.	-	FPTAVRIMAN
12	BI/DB	Tap changer single/parallel -ind.	DB with IEC 61850-8	FPTAVRISGL
13	BI/AO	Tap changer master/slave -ind. or Raise/Lower cmd.	AO with IEC 61850-8	FPTAVRIMST or FPTAVRCPOS
14	BO	Tap changer Manual cmd or	-	FPTAVRCMAN
15	BO	Tap changer Auto cmd or Parallel select cmd	-	FPTAVRCMAN or FPTAVRCSGL
16	BO	Tap changer Raise cmd or tap changer Raise/Lower cmd or	-	FPTAVRCPOS or FPTAVRCSGL
17	BO	Tap changer Lower cmd or Parallel execute cmd	-	FPTAVRCPOS or FPTAVRCSGL
18	BO	Single execute cmd	Only IEC 61850-8	FPTAVRCSGL
19	BO	Single/Parallel cancel cmd	Only IEC 61850-8	FPTAVRCSGL
20	BI/AI	Tap changer external blocking	AI with IEC 61850-8	FPTAVRIBLK
24	AI	Voltage	-	FPTAVRMVOL
25	AI	Reference voltage	-	
Table continues on next page				

Index	Object type	Meaning of Process object	Remarks	Group identifier
26	AO	Reference voltage command		
29	AI	Tap changer operation counter	-	FPTAVRICNR
30	AI	Tap changer selected on monitor	-	FPTAVRXMEV
31	BI	Master indication	Only if operation mode is Master/ Slave	FPTAVRIMST
32	BI	Follower indication	Only if operation mode is Master/ Slave	FPTAVRIMST
33	BO	Master command	Only if operation mode is Master/ Slave	FPTAVRIMAN
34	BO	Follower command	Only if operation mode is Master/ Slave	FPTAVRIMAN
70	BI	Local/Remote-switch	Optional	FPTAVRILRS
71	BI	Operator place-switch	Optional	FPTAVRIOPS
72	BO	Operator place-switch command	Optional	FPTAVRCOPS

6.8.3.9 Scale objects

At the first installation, the Tap changer creates the scale 1_1 (linear 1:1 scale). For ANSI station type, the installation creates also the scale BGU_1_1000 (linear 1:1000 scale).

6.8.3.10 Operation counters

In the **Tap changer control** dialog, the AZ, OV, IU and HI attributes of the dedicated analog input point are used for operation counting. The AZ, OV and HI attributes are not readable if the IU attribute value is 0.

6.9 Measurement

6.9.1 Standard function installation

This section describes the installation of measurement standard function from the Power Process Library. The standard function for the measurement is found in the directory /SA_LIB/ BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Process/Measurement). This standard function is configured by using the configuration tools, such as the Object Navigator:

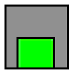
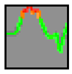
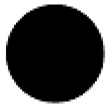
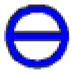
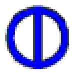
6.9.1.1 Symbol installation

The Power Process symbol for the measurement standard function is installed by using the Display Builder functions:

1. Object Browser
2. Palette

For further information, see the SYS600 Process Display Design manual.

Table 68: Power Process symbols for measurement standard function

File name		Representation	Symbol	
Horizontal	Vertical		Horizontal	Vertical
Bar Graph.sd				
Line Graph 20 Sample.sd				
Value.sd			20.62kV	
Circle.sd				
Current Transducer ANSI H.sd	Current Transducer ANSI.sd	ANSI		
Current Transducer IEC H.sd	Current Transducer IEC.sd	IEC		
Voltage Transducer ANSI H.sd	Voltage Transducer ANSI.sd	ANSI		
Voltage Transducer IEC H.sd	Voltage Transducer IEC.sd	IEC		

6.9.2 Standard function configuration

6.9.2.1 Configuration with tools

The measurement standard function has the following attributes to be configured with the Object Navigator:

Table 69: Configurable measurement picture function attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of database process objects	-
STATION_TYPE	The type of control device & protocol	LON
TYPE_MEAS_'number'	The type of the measurement	None
FOLDER_'number_TITLE	Abbreviation of the measurement	-
MEAS_'number_DECIMALS	Decimals used for presenting the value	0
Table continues on next page		

Attribute	Meaning	Default
AUTHORIZATION_GROUP	The name of the authorization group	MV_CONTROL
EVENT_RECORDING	Process objects for event recording with RP-570	No process objects
OPC_ITEM_PREFIX	Specifies the OPC Item Prefix used as IEC 61850 Instance information	"IEC61850 Subnetwork.IED1
OPC_LN_INSTANCE_1	Specifies the OPC Logical Node names used as IEC 61850 Instance information for the measurement type 1	VECTOR("MMXU1")
OPC_LN_INSTANCE_2	Specifies the OPC Logical Node names used as IEC 61850 Instance information for the measurement type 2	VECTOR("MMXU1")
OPC_LN_INSTANCE_3	Specifies the OPC Logical Node names used as IEC 61850 Instance information for the measurement type 3	VECTOR("MMXU1")
OPC_LN_INSTANCE_4	Specifies the OPC Logical Node names used as IEC 61850 Instance information for the measurement type 4	VECTOR("MMXU1")

For more detailed description on the configurable attribute, see Section 6.15.

6.9.2.2 Configuration of process objects

Depending on the measurement standard function configuration, the tools create a certain set of process objects in the database. Process objects that will to be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

Table 70: Process objects created for measurement

Index	Explanation	Purpose
10	Current L1	Current measurement on phase L1
11	Current L2	Current measurement on phase L2
12	Current L3	Current measurement on phase L3
13	Neutral current I0	Neutral current measurement
14	Directional neutral current Ij	Directional neutral current measurement
15	Maximum demand I15min	Current measurement of 15 minutes maximum demand
16	Voltage U12	Phase to phase voltage measurement on phases L1-L2
17	Voltage U23	Phase to phase voltage measurement on phases L2-L3
18	Voltage U31	Phase to phase voltage measurement on phases L3-L1
19	Residual voltage U0	Residual voltage measurement
20	Active power P	Active power measurement
21	Reactive power Q	Reactive power measurement
22	Apparent power S	Apparent power measurement

Table continues on next page

Index	Explanation	Purpose
23	Power factor Cosj	Power factor measurement
24	Frequency f	Frequency measurement
25	Harmonic distortion	Harmonic distortion measurement
26	Temperature	Temperature measurement
27-32	User-defined measurement	Any analog input (AI) type of user-defined measurement
33-37	User-defined pulse counter	Any pulse counter (PC) type of user-defined measurement
38	Active energy (kWh)	Active energy counter for kWh
39	Active energy (MWh)	Active energy counter for MWh
40	Active energy (GWh)	Active energy counter for GWh
41	Reversed active energy (kWh)	Reversed active energy counter for kWh
42	Reversed active energy (MWh)	Reversed active energy counter for MWh
43	Reversed active energy (GWh)	Reversed active energy counter for GWh
44	Reactive energy (kvarh)	Reactive energy counter for kvarh
45	Reactive energy (Mvarh)	Reactive energy counter for Mvarh
46	Reactive energy (Gvarh)	Reactive energy counter for Gvarh
47	Reversed reactive energy (kvarh)	Reversed reactive energy counter for kvarh
48	Reversed reactive energy (Mvarh)	Reversed reactive energy counter for Mvarh
49	Reversed reactive energy (Gvarh)	Reversed reactive energy counter for Gvarh
50	Voltage U1	Voltage for phase U1
51	Voltage U2	Voltage for phase U2
52	Voltage U3	Voltage for phase U3
53	Voltage U0	Voltage for phase U0
54	Residual Current I0	Residual Current I0
73	Active power L1	Active power for L1
74	Active power L2	Active power for L2

Table continues on next page

Index	Explanation	Purpose
75	Active power L3	Active power for L3
79	Reactive power L1	Reactive power for L1
80	Reactive power L2	Reactive power for L2
81	Reactive power L3	Reactive power for L3
91	Power factor L1	Power factor for L1
92	Power factor L2	Power factor for L2
93	Power factor L3	Power factor for L3
131	Current	Current (not allocated to a phase)
132	Voltage	Voltage (not allocated to a phase)
133	Active Power	Active Power (not allocated to a phase)
134	Reactive Power	Reactive Power (not allocated to a phase)
135	Apparent Power	Apparent Power (not allocated to a phase)
136	Power factor	Power factor (not allocated to a phase)
137	Impedance	Impedance (not allocated to a phase)
138	Frequency	Frequency (not allocated to a phase)
139	Positive, Negative and Zero Sequence Current	Positive, Negative and Zero Sequence Current
140	Positive, Negative and Zero Sequence Current	The absolute measured values of Positive, Negative and Zero Sequence Current
141	Positive, Negative and Zero Sequence Current	The absolute measured values of Positive, Negative and Zero Sequence Current
142	Positive, Negative and Zero Sequence Voltage	The absolute measured values of Positive, Negative and Zero Sequence Voltage
143	Positive, Negative and Zero Sequence Voltage	The absolute measured values of Positive, Negative and Zero Sequence Voltage
144	Positive, Negative and Zero Sequence Voltage	The absolute measured values of Positive, Negative and Zero Sequence Voltage

With RTU 2xx/RP-570 type of stations Event recording objects (for accurate time stamp) are created with convention Index + 100. These are optional.

6.9.2.3 Example of measurement configuration

The configuration of the measurement with a horizontal field can be given as an example:

Table 71: Measurement configuration

Attribute	Value
STATION_NAME	Eastwick
BAY_NAME	Outgoing HA2
DEVICE_NAME	CT
P_OBJECT_LN	ESTHA02CT
STATION_TYPE	LON
TYPE_MEAS_1	Current L1
FOLDER_1_TITLE	IL1
MEAS_1_DECIMALS	0
TYPE_MEAS_2	Active power P
FOLDER_2_TITLE	P
MEAS_2_DECIMALS	1
TYPE_MEAS_3	Reactive power Q
FOLDER_3_TITLE	Q
MEAS_3_DECIMALS	1
TYPE_MEAS_4	None
FOLDER_4_TITLE	N/A
MEAS_4_DECIMALS	N/A
AUTHORIZATION_GROUP	MV_CONTROL
EVENT_RECORDING	N/A

6.9.3 Application engineering information

6.9.3.1 Structure of measurement standard function

This section describes the structure of the measurement standard function. All subdrawing files, form pictures, help and other text files, as well as database objects are included. The measurement is a part of the standard functions of Power Process Library and has a directory /SA_LIB/BASE/BBONE, and the standard Power Process Library subdirectories INST, LANG0 and USE.

6.9.3.2 Files

The table below lists all measurement standard function related files and their functionality.

Table 72: Measurement standard function files

File	Functionality	Path
Bar Graph.sd	Measurement bar graph symbol	/PROG/GRAPHICSENGINE/PALETTE/05 - SA_Measurement
Line Graph.sd	Measurement line graph symbol	/PROG/GRAPHICSENGINE/PALETTE/05 - SA_Measurement
Value.sd	Measurement value symbol	/PROG/GRAPHICSENGINE/PALETTE/05 - SA_Measurement
Circle.sd	Measurement circle symbol	/PROG/GRAPHICSENGINE/PALETTE/05 - SA_Measurement
Current Transducer ANSI.sd	SA ANSI symbol for current transducer	/PROG/GRAPHICSENGINE/PALETTE/05 - SA_Measurement
Current Transducer IEC.sd	SA IEC symbol for current transducer	/PROG/GRAPHICSENGINE/PALETTE/05 - SA_Measurement
Voltage Transducer ANSI.sd	SA ANSI symbol for voltage transducer	/PROG/GRAPHICSENGINE/PALETTE/05 - SA_Measurement
Voltage Transducer IEC.sd	SA IEC symbol for voltage transducer	/PROG/GRAPHICSENGINE/PALETTE/05 - SA_Measurement
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

6.9.3.3 Language text file

The following Text Translation Tool compatible text file is used by the measurement standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 73: Language text file

File	Functionality
SAI_ME.TXT	Text file for the database creation of the standard function

6.9.3.4 Help text file

The path to the measurement help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 74: Help text file

File	Functionality
SAI_ME1.HLP	Standard function installation help file

6.9.3.5 Configuration files

The following configuration files are used by the measurement standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 75: Configuration files used by the measurement standard function

File	Functionality
SAI_ME1.DAT	Contains the configuration data for measurement when it is created modified or deleted by configuration tools.
SAI_ME1.POT	Contains the process object definitions for the Object Navigator.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for installation tool.

6.9.3.6 Other text files

The following text file is used by the measurement. The path is /SA_LIB/BASE/BBONE/USE.

Table 76: Text file used by the measurement

File	Functionality
BGU_AUTH.TXT	Contains the SCIL code for station authority check. The code is executed by the dialog of the measurement.

6.9.3.7 Command procedures

At the first installation, the measurement creates the command procedure BGU_UPDATE, which updates process objects when they are update-deblocked. This feature is similar to the SYS600 Blocking Display function. It is also used when simulation mode is disabled in control dialog.

Table 77: Command procedures used by measurement

BGU_CONTROL Performs the control operations	
Function	**Description**
OBJECT_INFO	Gets configuration of the standard function.
GET_OI_TEXT	Gets object identification text.
CHECK_AUTHORITY_WORKPLACE	Checks if the used workstation is authorized to make control operations.
GET_ALARMS	Gets alarms of standard function.
GET_OFFSETS	Gets index offsets for indexes that are used for indicating command event, selection on monitor, control blocking, alarm, external blocking, open blocking and close blocking.
BGU_BLOCK Performs the blocking operations	
Function	Description
GET_BLOCKING_STATE	Gets blocking state for objects that belong to the standard function.
SET_BLOCKING_STATE	Sets blocking state for the objects that belong to the standard function. If standard function is a bay, applies to all objects that belong to the bay. If standard function is a station, applies to all objects that belong to the station.
GET_ENABLED_BLOCKINGS	Disables blocking possibility in some standard functions. For example, control blocking cannot be used with a measurement and alarm indicator standard functions.

6.9.3.8 Process objects

The following process objects are created depending on the configuration of the measurement.

Table 78: Measurement process objects

Index	Obj. type	Process object	Remarks	Group identifier
10	AI	Current L1	-	FPPMEAMCUR
11	AI	Current L2	-	FPPMEAMCUR
12	AI	Current L3	-	FPPMEAMCUR
13	AI	Neutral current I0	-	FPPMEAMNCU
14	AI	Directional neutral current Ij	-	FPPMEAMDNC

Table continues on next page

Index	Obj. type	Process object	Remarks	Group identifier
15	AI	Maximum demand I15min	-	FPPMEAMMDC
16	AI	Voltage U12	-	FPPMEAMVOL
17	AI	Voltage U23	-	FPPMEAMVOL
18	AI	Voltage U31	-	FPPMEAMVOL
19	AI	Residual voltage U0	-	FPPMEAMRVL
20	AI	Active power P	-	FPPMEAMAPW
21	AI	Reactive power Q	-	FPPMEAMRPW
22	AI	Apparent power S	-	FPPMEAMAPP
23	AI	Power factor Cosj	-	FPPMEAMCOS
24	AI	Frequency f	-	FPPMEAMFRQ
25	AI	Harmonic distortion	-	FPPMEAMHDS
26	AI	Temperature	-	FPPMEAMTMP
27-32	AI	User defined measurement	-	FPPMEAMXXX
33-37	PC	User defined pulse counter	-	FPPMEAMXXX
38	PC	Active energy E (kWh)	-	FPPMEAMAEN
39	PC	Active energy E (MWh)	-	FPPMEAMAEN
40	PC	Active energy E (GWh)	-	FPPMEAMAEN
41	PC	Reverse active energy E (kWh)	-	FPPMEAMAEN
42	PC	Reverse active energy E (MWh)	-	FPPMEAMAEN
43	PC	Reverse active energy E (GWh)	-	FPPMEAMAEN
44	PC	Reactive energy E (kvarh)	-	FPPMEAMREN
45	PC	Reactive energy E (Mvarh)	-	FPPMEAMREN
46	PC	Reactive energy E (Gvarh)	-	FPPMEAMREN
47	PC	Reverse reactive energy E (kvarh)	-	FPPMEAMREN
48	PC	Reverse reactive energy E (Mvarh)	-	FPPMEAMREN
49	PC	Reverse reactive energy E (Gvarh)	-	FPPMEAMREN
50	AI	Voltage U1	-	FPPMEAMVOL
51	AI	Voltage U2	-	FPPMEAMVOL
52	AI	Voltage U3	-	FPPMEAMVOL
53	AI	Voltage U0	-	FPPMEAMVOL
54	AI	Residual Current I0	-	FPPMEAMRCR
73	AI	Active power L1	-	FPPMEAMPAP

Table continues on next page

Index	Obj. type	Process object	Remarks	Group identifier
74	AI	Active power L2	-	FPPMEAMPAP
75	AI	Active power L3	-	FPPMEAMPAP
79	AI	Reactive power L1	-	FPPMEAMPRP
80	AI	Reactive power L2		FPPMEAMPRP
81	AI	Reactive power L3	-	FPPMEAMPRP
91	AI	Power factor L1	-	FPPMEAMCOS
92	AI	Power factor L2	-	FPPMEAMCOS
93	AI	Power factor L3	-	FPPMEAMCOS
131	AI	Current	-	FPPMEAMCUR
132	AI	Voltage	-	FPPMEAMVOL
133	AI	Active Power	-	FPPMEAMAPW
134	AI	Reactive Power	-	FPPMEAMRPW
135	AI	Apparent Power	-	FPPMEAMAPP
136	AI	Power factor	-	FPPMEAMCOS
137	AI	Impedance	-	FPPMEAMIMP
138	AI	Frequency	-	FPPMEAMFRQ
139	AI	Positive, Negative and Zero Sequence Current	-	FPPMEAMCSQ
140	AI	Positive, Negative and Zero Sequence Current	-	FPPMEAMCSQ
141	AI	Positive, Negative and Zero Sequence Current	-	FPPMEAMCSQ
142	AI	Positive, Negative and Zero Sequence Voltage	-	FPPMEAMVSQ
143	AI	Positive, Negative and Zero Sequence Voltage	-	FPPMEAMVSQ
144	AI	Positive, Negative and Zero Sequence Voltage	-	FPPMEAMVSQ

For RP-570, event recordind objects are created with convention shown in the table below. These are Optional.

Table 79: Process objects for Event recording

Index	Obj. type	Process object	Remarks	Group identifier
Index + 100 (Index refers to Index in previous table)	EVREC	Same as Index -100	Only RP-570, optional	xxxxxxExxx, where x is same as in Index - 100

6.9.3.9 Scale objects

At the first installation, the measurement creates the scale 1_1 (linear 1:1 scale). For ANSI station type, the scale BGU_1_1000 (linear 1:1000 scale) is also created at the installation.

6.10 Alarm indicator

6.10.1 Standard function installation

This section describes the installation of alarm indicator standard function from the Power Process Library. The standard function for the alarm indicator is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Process/Alarm Indicator). This standard function is configured by using configuration tools, such as the Object Navigator.

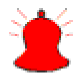
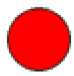
6.10.1.1 Symbol installation

The Power Process symbol for alarm indicator standard function is installed by using the Display Builder functions:

1. Object Browser
2. Palette

For further information, see the SYS600 Process Display Design manual.

Table 80: Power Process symbol for the alarm indicator standard function

File Name	Symbol
Alarm A.sd	
Alarm Bell.sd	
Alarm Led.sd	
Alarm Star.sd	

6.10.1.2 Configuration with tools

The alarm indicator standard function has the following attributes to be configured with the Object Navigator:

Table 81: Alarm indication attributes to be configured with the tools

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of database process object	-
STATION_TYPE	The type of control device & protocol.	LON
TYPE_OF_SIGNAL'x'	The type of the input signal.	-
SIGNAL'x'_TEXT	Identification for signal shown in event lists, alarm list and so on	Not available
Table continues on next page		

Attribute	Meaning	Default
ALARMING_STATE_OF_SIGNAL’x’	The state of the signal when an alarm (warning) is activated.	Not available
LINK_SIGNAL’x’_LN	The logical name of the process object to which the signal is linked.	Not available
LINK_SIGNAL’x’_IX	The index of the process object to which the signal is linked.	Not available
AUTHORIZATION_GROUP	The name of the authorization group.	MV_CONTROL
EVENT_RECORDING	Selection whether the event recording process objects are created when the RP-570 protocol is used.	Not available

For more detailed description on the configurable attributes, see Section 6.15.

6.10.1.3 Configuration of process objects

Depending on the configuration of the alarm indicator standard function, the tools will create a certain set of process objects in the database. Process objects that will be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

If the value of the attribute TYPE_OF_SIGNAL'nr' is linked, the process object is not created by the tools, but the name and the index of the signal is given to the attributes LINK_SIGNAL'nr'_TO_LN and LINK_SIGNAL'nr'_TO_IX.

Table 82: Process objects created for the alarm indicator

Index	Explanation	Purpose
10...17	Alarm indication	Used for indication of the alarming state of the signal

6.10.1.4 Example of alarm indicator configuration

Table 83: Alarm indicator configuration

Attribute	Value
STATION_NAME	Eastwick
BAY_NAME	Outgoing HA5
DEVICE_NAME	AI
P_OBJECT_LN	EST_HA5AI
STATION_TYPE	LON
TYPE_OF_SIGNAL1	Binary Input (BI)
SIGNAL1_TEXT	SF6 low pressure
ALARMING_STATE_OF_SIGNAL1	(1)
AUTHORIZATION_GROUP	MV_CONTROL

6.10.2 Application engineering information

6.10.2.1 Structure of alarm indicator standard function

This section describes the structure of the alarm indicator standard function. All subdrawing files, form pictures, help and other text files, as well as database objects are included. The alarm indicator is a part of the standard functions of Power Process Library and has a directory /SA_LIB/BASE/BBONE, and the standard Power Process Library subdirectories INST, LANG0 and USE.

6.10.2.2 Files

The table below lists all alarm indicator standard function related files and their functionality.

Table 84: Alarm indicator standard function related files

File	Functionality	Path
Alarm A.sd	Power Process symbol for alarming A	/PROG/GRAPHICSENGINE/PALETTE/04 - SA_Indication
Alarm Bell.sd	Power Process symbol for alarming bell	/PROG/GRAPHICSENGINE/PALETTE/04 - SA_Indication
Alarm Led.sd	Power Process symbol for alarming led	/PROG/GRAPHICSENGINE/PALETTE/04 - SA_Indication
Alarm Star.sd	Power Process symbol for alarming star	/PROG/GRAPHICSENGINE/PALETTE/04 - SA_Indication
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

6.10.2.3 Language text file

The following Text Translation Tool compatible text files are used by the alarm indicator standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 85: Text Translation Tool compatible text file used by the alarm indicator

File	Functionality
SAI_AI.TXT	Text file for the database creation of the standard function

6.10.2.4 Help text file

The path to the alarm indicator help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 86: Alarm indicator help text file

File	Functionality
SAI_AI.HLP	Standard function installation help file

6.10.2.5 Configuration files

The following configuration files are used by the alarm indicator standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 87: Configuration files used by the alarm indicator

File	Functionality
SAI_AI.DAT	Contains the configuration data for alarm indicator when it is created modified or deleted by configuration tools.
SAI_AI.POT	Contains the process object definitions for the Object Navigator.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for installation tool.

6.11 Auto Reclose

6.11.1 Standard function installation

This section describes the installation of Auto Reclose standard function from the Power Process Library. The standard function for the Auto Reclose is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Process/Auto Reclose). This standard function is configured by using configuration tools, such as the Object Navigator.

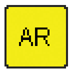
6.11.1.1 Symbol installation

The Power Process symbol for Auto Reclose standard function is installed by using the Display Builder functions:

1. Object Browser
2. Palette

For further information, see the SYS600 Process Display Design manual.

Table 88: Power Process symbol for Auto Reclose standard function

File Name	Symbol
Auto-Reclosing BI.sd	
Auto-Reclosing.sd	

6.11.1.2 Configuration with tools

The Auto Reclose standard function has the following attributes to be configured with the Object Navigator:

Table 89: Auto Reclose standard function configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of database process objects	-
STATION_TYPE	The type of control device & protocol	LON
Table continues on next page		

Attribute	Meaning	Default
OUTPUT_STATUS	Process objects for output status with DNP 3.0 protocol	N/A
AR_WITH_MULTIPLE_CYCLES	Select object type to indicate auto-reclosure in progress	No
SHOW_AR_IN_USE	Show tag during auto-reclosure	Show tag
SHOW_AR_NOT_IN_USE	Show tag also when no auto-reclosing is in progress	Do not show tag
EVENT_RECORDING	Process objects for event recording with RP-570	No process objects
LINK_AR_TAG_TO_LN	Create link between existing auto-reclosing objects from other Object Browser functions	-
LINK_AR_TAG_TO_MODULE	Select type designation of the auto-reclosing module	No linking
OPC_ITEM_PREFIX	Specifies the OPC Item Prefix used as IEC 61850 Instance information	IEC61850 Subnetwork.IED1.LD1
OPC_LN_INSTANCES	Specifies the OPC Logical Node names used as IEC 61850 Instance information	VECTOR("RREC1")

For more detailed description on the configurable attributes, see Section 6.15.

6.11.1.3 Configuration of process objects

Depending on the configuration of the Auto Reclose standard function, the tools create a certain set of process objects in the database. Process objects that will be linked to an actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

Table 90: Process objects created for Auto Reclose

Index	Explanation	Purpose
10	Auto-reclosing in use/out of use indication (optional)	Indicates if the auto-reclose is in use (software switch of the auto-reclosing unit).
11	Auto-reclosing in use/out of use indication (optional)	Indicates if the auto-reclose is in use (hardware switch of the auto-reclosing unit).
12	Auto-reclosing in progress/not in progress ind. (optional)	Indicates if there is an auto-reclosing sequence running.
13	Auto-reclosing interrupt command (optional)	Interrupts the auto-reclosing sequence.
14	Auto-reclosing in use/out of use command (optional)	Enables/disables the auto-reclosure.
110	Auto-reclosing in use/out of use indication (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
111	Auto-reclosing in use/out of use indication (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.

Table continues on next page

Index	Explanation	Purpose
112	Auto-reclosing in progress/not in progress ind. (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
113	Auto-reclosing interrupt command (optional)	Command termination object with IEC 60870-5-101/104 and -103 type of stations. Output status object with DNP 3.0 type of stations.
114	Auto-reclosing in use/out of use command (optional)	Command termination object with IEC 60870-5-101/104 and -103 type of stations. Output status object with DNP 3.0 type of stations.

6.11.2 Application engineering information

6.11.2.1 Structure of Auto Reclose standard function

This section describes the structure of the Auto Reclose standard function. All sub-drawing files, form pictures, help and other text files, as well as database objects are included. The Auto Reclose is a part of the standard functions of Power Process Library and has a directory /SA_LIB/BASE/BBONE, and the standard Power Process Library subdirectories INST, LANG0 and USE.

6.11.2.2 Files

The table below lists all Auto Reclose standard function related files and their functionality.

Table 91: Auto Reclose standard function related files

File	Functionality	Path
Auto-Reclosing.sd	Power Process symbol for Auto Reclose	/PROG/GRAPHICSENGINE/PALETTE/04 - SA_Indication
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

6.11.2.3 Language text file

The following Text Translation Tool compatible text files are used by the Auto Reclose standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 92: Text Translation Tool compatible text file used by the Auto Reclose standard function

File	Functionality
SAI_ARC2.TXT	Text file for the database creation of the standard function

6.11.2.4 Help text file

The path to the Auto Reclose help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 93: Auto Reclose help text file

File	Functionality
SAI_ARC2.HLP	Standard function installation help file

6.11.2.5 Configuration files

The following configuration files are used by the Auto Reclose standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 94: Configuration files used by the Auto Reclose stardard function

File	Functionality
SAI_ARC2.DAT	Contains the configuration data for the Auto Reclose when it is created, modified or deleted by configuration tools.
SAI_ARC2.POT	Contains the process object definitions for the Object Navigator.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for installation tool.

6.11.2.6 Process objects

The following process objects are created for the Auto Reclose standard function.

Table 95: Process objects created for the Auto Reclose standard function

Index	Obj. type	Process object	Remarks	Group ident.
10	BI	Auto-reclosing in use	Software switch Optional	FPFARCISCE
11	BI	Auto-reclosing in use	Hardware switch Optional	FPFARCISCE
12	BI or AI	Auto-reclosing in progress	Optional	FPFARCIARC
13	BO	Auto-reclosing interruption	Optional	FPFARCCINR
14	BO	AR in use/ out of use -cmd	Optional	FPFARCCSCE
110	EVREC	Event recording for auto-reclosing in use	Only RP-570 Optional	FPFARCESCE
111	EVREC	Event recording for auto-reclosing in use	Only RP-570 Optional	FPFARCESCE
112	EVREC	Event recording for auto-reclosing in progress	Only RP-570 Optional	FPFARCEARC
113	TERM / OS	Command termination of auto-reclosing interruption Output status object	Only IEC 60870-5-101/104 and -103; Optional Output status object when station type is DNP 3.0, Optional	FPFARCTINR
114	TERM / OS	Command termination of auto-reclosing in use/ out of use – cmd Output status object	Only IEC 60870-5-101/104 and -103; Optional Output status object when station type is DNP 3.0, Optional	FPFARCTSCE

6.11.2.7 Scale objects

At the first installation the Auto Reclose creates the scale 1_1 (linear 1:1 scale). For multiple shots with ANSI station type, the installation creates also the scale BGU_1_1000 (linear 1:1000 scale).

6.12 Trip Signal

6.12.1 Standard function installation

This section describes the installation of Trip Signal standard function from the Power Process Library. The standard function for the Trip Signal is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Process/Trip Signal). This standard function is configured by using the configuration tools, such as the Object Navigator.

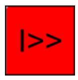
6.12.1.1 Symbol installation

The Power Process symbol for Trip Signal standard function is installed by using the Display Builder functions:

1. Object Browser
2. Palette

For further information, see the SYS600 Process Display Design manual.

Table 96: Power Process symbol for the Trip Signal standard function

File Name	Symbol
Tripping.sd	

6.12.1.2 Configuration with tools

The Trip Signal standard function has the following attributes to be configured with the Object Navigator:

Table 97: Configurable Attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of database process objects	-
STATION_TYPE	The type of control device & protocol	LON
OUTPUT_STATUS	Process objects for output status with DNP 3.0 protocol	N/A
Table continues on next page		

Attribute	Meaning	Default
EVENT_RECORDING	Process objects for event recording with RP-570	No process objects
LINK_TRIP_TAG_TO_LN_x	The logical name of the database process objects to which this trip tag is linked.	-
LINK_TRIP_TAG_TO_	The module to which this trip tag is linked.	No linking

For more detailed description on the configurable attributes, see Section 6.15.

6.12.1.3 Configuration of process objects

Depending on the configuration of the Trip Signal standard function, the tools create a certain set of process objects in the database. Process objects that will be linked to actual process should be edited to have a station number, an address and a proper switching state. All other attributes should have suitable values for normal operation by default.

Table 98: Process objects created for Trip Signal

Index	Explanation	Purpose
10	General protection trip indication (optional)	The general protection trip indication can be used as ON/OFF type indication of any protection trip within the bay.
11	General trip relay latched indication (optional)	The general trip relay latched indication can be used as ON/OFF type indication of any latched protection trip relay within the bay.
12-	General trip relay reset command (optional)	The general trip relay reset command object is an output object which can be used to reset latched output relays, trip indicators and memorised parameters.
110	General protection trip indication (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
111	General trip relay latched indication (optional)	Event recording object (for accurate time stamp) with RTU 2xx/RP-570 type of stations.
112-	Indication for command termination (optional)	Indicates with IEC 60870-5-101/104 and -103 whether the issued command was successful or not.

6.12.2 Application engineering information

6.12.2.1 Structure of the Trip Signal standard function

This section describes the structure of the Trip Signal standard function. All subdrawing files, form pictures, help and other text files, as well as database objects are included. The Trip Signal is a part of the standard functions of Power Process Library and has the directory /SA_LIB/BASE/BBONE, and the standard Power Process Library subdirectories INST, LANG0 and USE.

6.12.2.2 Files

The table below lists all Trip Signal standard function related files and their functionality.

Table 99: Trip Signal stardard function related files

File	Functionality	Path
Tripping.sd	Power Process symbol for Trip Signal	PROG/GRAPHICSENGINE/PALETTE/04 - SA_Indication
FORM5SAGR1.PIC	Generic format picture for printing	SA_LIB/BASE/BBONE/USE

6.12.2.3 Language text file

The following Text Translation Tool compatible text file is used by the Trip Signal standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 100: Text Translation Tool compatible text file used by the Trip Signal standard function

File	Functionality
SAI_TRIP2.TXT	Text file for the database creation of the standard function

6.12.2.4 Help text file

The path to the Trip Signal help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 101: Help text file used by the Trip Signal

File	Functionality
SAI_TRIP2.HLP	Standard function installation help file

6.12.2.5 Configuration files

The following configuration files are used by the Trip Signal standard function. The path is / SA_LIB/BASE/BBONE/INST.

Table 102: Configuration files used by the Trip Signal standard function

File	Functionality
SAI_TRIP2.DAT	Contains the configuration data for Trip Signal when it is created modified or deleted by configuration tools.
SAI_TRIP2.POT	Contains the process object definitions for the Object Navigator.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for installation tool.

6.12.2.6 Process objects

The following process objects are created for the Trip Signal standard function.

Table 103: Process objects created for the Trip Signal

Index	Obj. type	Process object	Remarks	Group ident.
10	BI	Protection trip	Optional	FPFTRPITRP
11	BI	Trip relay latched	Optional	FPFTRPITRL
12-	BO	Latched trip relay reset	Optional	FPFTRPCTRL
Table continues on next page				

Index	Obj. type	Process object	Remarks	Group ident.
110	EVREC	Protection trip	Only RP-570 Optional	FPFTRPETRP
111	EVREC	Trip relay latched	Only RP-570 Optional	FPFTRPETRL
112-	TERM	Command termination of latched trip relay reset	Only IEC 60870-5-101/104 and -103; Optional	FPFTRPTTRL

6.12.2.7 Scale objects

At the first installation the Trip Signal creates the scale 1_1 (linear 1:1 scale).

6.13 Generator

6.13.1 Standard function installation

This section describes the installation of generator standard function from the Power Process Library. You can find the standard function for generator in the /SA_LIB/BASE/BBONE/INST directory and install it by using the Power Process Library installation procedures (SA_LIB/Process/Generator). To configure this standard function, you can use the Object Navigator.

6.13.1.1 Symbol installation

The Power Process symbol for the Generator standard function is installed by using the following Display Builder functions:

1. Object Browser
2. Palette

For further information, see the SYS600 Process Display Design manual.

Table 104: Power Process symbol for the generator standard function

File Name	Symbol
Generator.sd	

6.13.1.2 Configuration with tools

The following generator standard function attributes can be configured with the Object Navigator:

Table 105: Generator standard function configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of database process objects	-
STATION_TYPE	The type of control device & protocol	IEC61850

For more detailed description on the configurable attributes,see Section 6.15.

6.13.1.3 Configuration of process object

The generator's configuration creates one process object in the database.

Table 106: Process object created for the generator

Index	Explanation	Purpose
10	Generator indication	Used for feeding/not feeding information for topology coloring
11	Generator state	Used for Stopped/Starting/Running/Stopping information
255	Generator color	Used for the generator coloring object by DMS600

6.13.2 Application engineering information

6.13.2.1 Structure of the generator standard function

This section describes the structure of the generator standard function. All the subdrawing files, form pictures, help and other text files, as well as database objects are included in it. The generator is a part of the standard functions of Power Process Library and has the directory /SA_LIB/BASE/BBONE and the standard Power Process Library subdirectories INST, LANG0 and USE.

6.13.2.2 Files

The table below lists all the generator standard function related files and their functionality.

Table 107: Generator standard function related files

File	Functionality	Path
Generator.sd	Power Process symbol for the generator	PROG/GRAPHICSENGINE/PALETTE/ 01 - SA_Common

6.13.2.3 Language text file

The following Text Translation Tool compatible text file is used by the generator standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 108: Text Translation Tool compatible text file used by the generator standard function

File	Functionality
SAI_GEN.TXT	Text file for the database creation of the standard function

6.13.2.4 Help text file

The path to the generator help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 109: Help text file used by the generator

File	Functionality
SAI_GEN.DAT	Contains the configuration data and process object definitions for the generator when it is created, modified or deleted by configuration tools.

6.13.2.5 Configuration files

The following configuration files are used by the generator standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 110: Configuration files used by the generator standard function

File	Functionality
SAI_GEN.DAT	Contains the configuration data and process object definitions for the generator when it is created, modified or deleted by configuration tools.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for the Installation tool.

6.13.2.6 Command procedures

The Generator uses the command procedure (and Event channel) BGU_SET_GENERATOR_ON_OFF, which re-routes the generator state (index 11) to generator indication (index 10). Generator indication is used by topology coloring for feeding/not feeding information.

6.13.2.7 Process object

The following process object is created for the generator standard function.

Table 111: Process object created for the generator

Index	Obj. type	Process object	Remarks	Group ident
10	AI	Generator indication	-	FPEGENIGIF
11	DI	Generator state	-	FPEGENIGST
255	AI	Generator color	-	BCHTCIXPCD

6.13.2.8 Scale object

At the first installation the generator creates the scale 1_1 (linear 1:1 scale).

6.14 Line Indicator

6.14.1 Standard function installation

This section describes the installation of line indicator standard function from Power Process Library. The standard function for the line indicator can be found in the /SA_LIB/BASE/BBONE/INST directory and it can be installed by using the Power Process Library installation procedures (SA_LIB/Process/Line Indicator). To configure this standard function, the Object Navigator can be used.

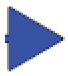
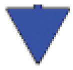
6.14.2 Symbol installation

The Power Process symbol can be installed for the line indicator standard function by using the Display Builder functions:

1. Object Browser
2. Palette

For further information, see the SYS600 Process Display Design manual.

Table 112: Power Process symbol for the line indicator standard function

File Name		Symbol	
Horizontal	Vertical	Horizontal	Vertical
Line Indicator H.sd	Line Indicator.sd		

6.14.3 Configuration with tools

The line indicator standard function has the following attributes to be configured with the Object Navigator.

Table 113: Line indicator standard function configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
P_OBJECT_LN	Logical name of the database process objects	-
LINE_INDICATOR_COLORED_BY	Specifies which product or function is used for the coloring	DMS600

For more detailed description on the configurable attributes, see Section 6.15.

6.14.4 Configuration of process objects

Configuration of the line indicator creates a certain set of process objects in the database.

Table 114: Process objects created for the line indicator

Index	Explanation	Purpose
10	DMS600 color number	Used for the line indicator coloring object by DMS600
11	Power flow direction	Used for storing the power flow direction, summary of local and remote power flow)
12	Power flow direction (local)	Used for storing the local power flow direction
13	Power flow direction (remote)	Used for storing the remote power flow direction
253	Virtual switch for Topol. Col.	Virtual switch for network topology coloring
254	Ext. ground ind. For Topol. Col.	External ground indicator for network topology coloring
255	Infeed color for Topol. Col.	Infeed color for network topology coloring

6.14.5 Application engineering information

6.14.5.1 Structure of the line indicator standard function

This section describes the structure of the line indicator standard function. All the subdrawing files, form pictures, help and other text files, as well as database objects are included. The line indicator is a part of the standard functions of Power Process Library and has the directory /SA_LIB/BASE/BBONE, and the standard Power Process Library subdirectories INST, LANG0 and USE.

6.14.5.2 Files

The table below lists all the line indicator standard function related files and their functionality.

Table 115: Line indicator standard function related files

File	Functionality	Path
Line Indicator.sd	Vertical Power Process symbol for Line Indicator	PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common
Line Indicator H.sd	Horizontal Power Process symbol for Line Indicator	PROG/GRAPHICSENGINE/PALETTE/
FORM5SAGR1.PIC	Generic format picture for printing	SA_LIB/BASE/BBONE/USE

6.14.5.3 Language text file

The following Text Translation Tool compatible text file is used by the line indicator standard function. The path is /SA_LIB/BASE/BBONE/LANG0.

Table 116: Text Translation Tool compatible text file used by the line indicator standard function

File	Functionality
SAI_LIND.TXT	Text file for the database creation of the standard function

6.14.5.4 Help text file

The path to the line indicator help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 117: Help text file used by the line indicator

File	Functionality
SAI_LIND.HLP	Standard function installation help file

6.14.5.5 Configuration files

The following configuration files are used by the line indicator standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 118: Configuration files used by the line indicator

File	Functionality
SAI_LIND.DAT	Contains the configuration data and process object definitions for the line indicator when it is created, modified or deleted by configuration tools.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for the Installation tool.

6.14.5.6 Command procedures

At first installation, the Line indicator creates the command procedure SAI_MTSTS, which updates the status of the line indicator color.

For the indication type of an analog input (AI), the installation creates the command procedure SAI_MT2TC, which updates the fictitious power indicator, grounding indicator and active voltage color.

6.14.5.7 Process objects

The following process objects are created for the line indicator standard function.

Table 119: Process objects created for the line indicator

Index	Obj. type	Process object	Remarks	Group ident.
10	AI	DMS600 color number	-	BCHMTCXFCD
11	DI	Power flow direction	-	BCHMTCIPFD
12	DI	Power flow direction (local)	-	BCHMTCIPFL
13	DI	Power flow direction (remote)	-	BCHMTCIPFR
253	BI	Virtual switch for Topol. Col.	-	BCHTCIXFPI
254	AI	Ext. ground ind. for Topol. Col.	-	BCHTCIXCCD
255	AI	Infeed color for Topol. Col	-	BCHTCIXPCD

6.14.5.8 Scale object

At first installation, the line indicator creates the scale 1_1 (linear 1:1 scale).

6.14.5.9 Configuring power flow direction calculation

Background

Template logic for power flow calculation is implemented in command procedure BGU_POWER_FLOW_CALC:C. The following convention is used in logic:

- Active power being supplied to the Busbar (Active power flow in) has a + sign
- Active power being consumed from the Busbar (Active power flow out) has a – sign
- Inductive Reactive Power is assumed to be consumed (all loads are more or less inductive), therefore inductive reactive power has a + sign
- Capacitive Reactive Power is assumed to be supplied (by a specific equipment to compensate inductive loads), therefore capacitive reactive power has a – sign

Additionally voltage inconsistency and VT group alarm can be taken into account in the logic. The handling for these needs to be added to logic project by project.

Based on the convention above the following states are calculated:

- 1 = Undetermined
- 2 = De-Energized (no voltage)
- 3 = Energized, but no power flow
- 4 = Active power flow in
- 5 = Active power flow out
- 6 = Capacitive reactive power (only if no active power flow)
- 7 = Inductive reactive power

The calculated power flow state is first set to local power flow indication (index 12). If process object for remote is in use, it is checked before the summary of local and remote power flow is set to index 11. Index 11 is the one that should be used in the displays.

For backward compatibility reasons the calculated power flow state is converted to index 10 where

- Value > 0 => Incoming power flow
- Value < 0 => Outgoing power flow
- Value = 0 => Passive

Configuration

The bay where power flow calculation logic is connected should have the following process objects:

- Active power (mandatory)
- Reactive power (optional)
- All phase to phase voltage measurements (mandatory)
- Line disconnector (mandatory)
- Earth switch (mandatory)
- Line indicator (mandatory)

Event channels (AN) is connected to process objects as shown in the picture below

LN	IX	[UN]	[OA]/IN	[OB]/	OI	OX	AN
ESTH02_LI	11				Eastwick Outgoing 110kV	Power flow direction	BGU_SET_POWER_FLOW_DIRECTION
ESTH02_LI	12				Eastwick Outgoing 110kV	Power flow direction (local)	BGU_POWER_FLOW_SUMMARY
ESTH02_LI	13				Eastwick Outgoing 110kV	Power flow direction (remote)	BGU_POWER_FLOW_SUMMARY
ESTH02_MEC	20				Eastwick Outgoing 110kV	Active power P	BGU_POWER_FLOW_CALC
ESTH02_MEC	21				Eastwick Outgoing 110kV	Reactive power Q	BGU_POWER_FLOW_CALC
ESTH02_MEV	16				Eastwick Outgoing 110kV	Voltage U12	BGU_POWER_FLOW_CALC
ESTH02_MEV	17				Eastwick Outgoing 110kV	Voltage U23	BGU_POWER_FLOW_CALC
ESTH02_MEV	18				Eastwick Outgoing 110kV	Voltage U31	BGU_POWER_FLOW_CALC
ESTH02_Q2	10				Eastwick Outgoing 110kV Q2	Disconn. position indication	BGU_POWER_FLOW_CALC
ESTH02_Q9	10				Eastwick Outgoing 110kV Q9	Earth sw. position indication	BGU_POWER_FLOW_CALC

Figure 80: Connecting power flow calculation to process objects

If you need to adapt command procedures, copy them to some other name and use those in the project. Otherwise the product update will override your changes

If no extensive logic is needed, another way to show power flow direction is just to re-route the information from e.g. active power measurement to index 11. First check that event channel and command procedure BGU_SET_POWER_FLOW_DIRECTION is attached to it and then:

- If active power > 0 -> set the value of index 11 to 4
- If active power < 0 -> set the value of index 11 to 5

6.15 Descriptions of the configurable attributes

6.15.1 STATION_NAME

A selector type of an editable combo box to select or enter the name of the substation. By default, the maximum length of the text is 9 characters. The selection list shows up to ten latest entries.

The configurable attribute is read directly from the process object database, if there is any process object linked to the selected station picture function. Therefore, the configurable attribute within a picture function has only meaning when the process objects have not been created.

This text is used as a substation identifier of the common station objects on event list, alarm list, printout, database query, and so on. It is very important that this text is similar to all objects within the same substation.

6.15.2 P_OBJECT_LN

The logical name of the database process objects. The maximum length of the text is 10 characters. Allowed characters are letters A-Z, all digits and the underscore (_).

The logical names of process objects have to be unique throughout the entire system, that is, different objects cannot have the same logical name. Therefore, it is very important to have a proper naming convention for the process object database.

Certain naming convention is also required to speed up the opening time of the main dialogs. With the predefined naming convention it is possible to collect all station and bay related database objects without searching through the complete database. Therefore, it is advised to use the following naming convention:

- The first three characters identify the substation
- The following four characters identify the bay
- The last three characters can be freely used to identify the source of the signal, that is device, unit, and so on.

The following list is presented as an example of the proper naming convention within one bay. The station name is Green Bay and the bay name is HA12 Downtown:

- GRB, station
- GRBHA12BAY, bay
- GRBHA12Q0, circuit breaker Q0

- GRBHA12Q1, truck Q1
- GRBHA12Q9, earth switch Q9
- GRBHA12CT, measurements/current transformer
- GRBHA12VT, measurements/voltage transformer
- GRBHA12ALA, alarm indicator
- GRBHA12AR, auto-reclosing tag
- GRBHA12TRP, tripping tag
- GRBHA12R1, feeder terminal or protection unit 1
- GRBHA12R2, feeder terminal or protection unit 1

If some other naming convention than 3+4 is used, it usually takes a longer time to open the main control dialogs.

6.15.3 STATION_TYPE

A selector type of a combo box to select the station type of the objects.

The database process objects are created for the station type defined here. The list of supported protocols is as follows:

- SPA
- ANSI
- LON
- RP-570
- RP-570 with FTABs
- IEC 60870-5-101/104
- IEC 60870-5-103
- IEC 61850-8
- DNP 3.0
- Modbus RTU/ASCII/TCP
- PROCOL

6.15.4 CMD_PARAMETER

A selector type of a combo box to select a command parameter for the control method. This attribute has meaning only if IEC 60870-5-101/104 or DNP 3.0 protocol is selected. Otherwise this attribute is disabled.

The supported control parameters for the IEC 60870-5-101/104 protocol are:

- Single command (ASDU 45)
- Double command (ASDU 46)
- Single command with time tag (ASDU 58)
- Double command with time tag (ASDU 59)
- Normalized value (ASDU 48)
- Scaled value (ASDU 49)
- Short floating point number (ASDU 50)
- Normalized value with time tag (ASDU 61)
- Scaled value with time tag (ASDU 62)
- Short floating point number with time tag (ASDU 63)

and for DNP 3.0 protocol:

- Direct
- Direct - No ack.

6.15.5 CONTROL_SUPERVISION_TIMEOUT

Timeout starts to run after the execute command. If indication is not received within the timeout, an event is generated about a failed command. Timeout is given in milliseconds. 0 means that timeout is not in use. The expected open/close state can be configured by STA:BSM and/or APL:BSM attributes.

6.15.6 CONTROL_SUPERVISION_ALARM

Defines the alarm class (AC) for control supervision timeout.

6.15.7 OUTPUT_STATUS

An ON/OFF type of a toggle button to select whether there is need for output status process objects with the DNP 3.0 protocol. If STATION_TYPE is other than DNP 3.0, the configurable attribute is disabled, and the value has no meaning.

6.15.8 AUTHORIZATION_GROUP

A selector type of an editable combo box to select the authorization group for the object. The maximum length of the text is 63 characters. The given text has to be usable as a SCIL list attribute name. For more information about the SCIL list attribute name, see SYS600 Programming Language SCIL. The contents of the selection list is taken from the existing authorization groups.

The attribute needs to be defined if the object is wanted to be included in some authorization group. The authorization level of each user in this group can be defined afterwards by the system manager with User Management pictures.

6.15.9 STATION_LR_IN_USE

An ON/OFF type of a toggle button to select if the station local/remote switch is in use, that is, a physical switch exists.

By default, the station local/remote switch is in use.

If this selection is set OFF, the control authority is given to everyone whose personal authorization level is high enough for the control actions. This is gained by setting the state (value) of the station local/remote switch permanently out of use.

The process object for station local/remote switch is created regardless of the selection. If the station local/remote switch is NOT in use, the process object should be left as it is created, that is, switching state (SS) to manual and value (DB) as 3 (out of use).

6.15.10 IND_DOUBLE_BINARY

An ON/OFF type of toggle button to select whether the station/remote switch or bay local/ remote indication type is double binary or single binary. This selection is disabled and it has no meaning if the station/remote switch or bay local/remote is not in use, that is, STATION_LR_IN_USE or BAY_LR_IN_USE is not set.

Depending on the actual field device or communication protocol, the indication of the local/ remote switch is sent either as a single binary (0/1) or as a double binary (0/1/2/3) format.

6.15.11 Station:

Binary input:

0 = Remote

1 = Station

Double binary input:

0 = Local

1 = Station

2 = Remote

3 = Out of use

6.15.12 Bay:

Binary input:

Local or Remote. Depends on Bay L/R switch polarity

Double binary input:

0 = Disabled

1 = Local

2 = Remote

3 = Reserved

6.15.13 IND_ANALOG_INPUT

An ON/OFF type of toggle button to select whether the bay switch indication type is analog input. This selection is disabled and it has no meaning if the local/remote switch is not in use, that is, BAY_LR_IN_USE is not set.

Analog input:

0 = Off

1 = Local

2 = Remote

3 = Error

4 = Error

5 = All

6.15.14 LR_REMOTELY_CONTROLLABLE

An ON/OFF type of a toggle button for the selection, whether the station local/remote switch can be controlled remotely by SYS600 or not.

By default, the station local/remote switch is defined as manually controllable only.

Typically, if there is a remote control for the station local/remote switch, switching election between the station and remote can be done remotely, whereas switching election to or from local has to be performed locally.

Switching to and from local can be achieved by setting the attribute SS (switching state) of the local command process object to OFF (0). Setting the SS attribute to 0 will make the relevant Radio button in the dialog insensitive for selections.

6.15.15 EVENT_RECORDING

An ON/OFF type of a toggle button to select whether there is need for event recording process objects with the RP-570 protocol. If STATION_TYPE is other than RTU 2xx or RTU 2xx/FTABs, the configurable attribute is disabled and the value has no meaning.

With the RP-570 the time stamp from process device comes into the event recording process object. If no accurate time stamp is needed or available, or the device is emulating the RP-570 without an event recording possibility, do not set this attribute ON.

6.15.16 OPC_ITEM_PREFIX

Specifies the OPC Item Prefix used as an IEC 61850 instance information. The default is IEC61850 Subnetwork.IED1.LD1.

6.15.17 OPC_LN_INSTANCES

Specifies the OPC Logical Node names used as IEC 61850 instance information. The default is VECTOR("LLN0").

6.15.18 BAY_NAME

A selector type of an editable combo box to select or enter the name of the bay (feeder). By default, the maximum length of the text is 14 characters. The selection list shows up to ten latest entries.

The configurable attribute is read directly from the process object database, if there is any process object linked to the selected bay picture function. Therefore, the configurable attribute within a picture function has meaning only when the process objects have not been created.

This text will be used as a bay (feeder) identifier of the common bay objects on event list, alarm list, printout, database query, and so on. It is very important that this text is similar to all objects within the same bay (feeder).

6.15.19 BAY_LR_IN_USE

An ON/OFF type of a toggle button to select if the bay local/remote switch is in use, that is, a physical switch exists.

By default, the bay local/remote switch is in use.

It is typical that if RTU 200 or RTU 210 is used as a remote terminal unit, there is no bay local/remote switches at bay level. The control authority of a single bay is determined by using station local/remote switch only.

The process object for bay local/remote switch is created regardless of the selection. If the bay local/remote switch is NOT in use, the process object should be left as it is created, that is, switching state (SS) to manual and value (DB) as 2 (remote).

6.15.20 BAY_LR_POLARITY

The polarity of the bay local/remote switch input signal.

6.15.21 LR_REMOTELY_CONTROLLABLE

An ON/OFF type of a toggle button to select if the bay local/remote switch can be controlled remotely by SYS600. This selection is disabled and it has no meaning if the bay local/remote switch in not in use, that is, BAY_LR_IN_USE is not set.

By default, bay local/remote switch is defined as manually controllable only.

It is typical that if there is a remote control for bay local/remote switch, shift to local can be done remotely, whereas shift from local to remote has to be performed locally. This can be achieved by setting the attribute SS (switching state) of the local command process object to OFF (0). By setting the SS-attribute to 0, the relevant Radio button on dialog becomes insensitive for shifting.

6.15.22 DEVICE_NAME

A selector type of an editable combo box to select or enter the name (identifier) of the switching device. By default, the maximum length of the text is 5 characters. The selection list shows up to ten latest entries.

The configurable attribute is read directly from the process object database, if there is any process object linked to the selected switching device.

Therefore, the configurable attribute within a picture function has meaning only when the process objects have not been created.

This text will be used as an identifier of the switching device on event list, alarm list, printout, database query, and so on.

6.15.23 SWICHING_DEVICE_TYPE

A selector type of a combo box to select the type of the switching device. By default, the type is a circuit breaker.

The configurable attribute has a great effect since it defines the index range for process objects, the texts within database and dialogs, the functionality and methods on dialogs, and so on. It is also used by the event and alarm list when filtering certain types of switching devices.

Circuit breakers:

- Circuit breaker
- Circuit breaker with synchro-check
- Circuit breaker-disconnector (no full breaking capacity)

Disconnectors:

- Disconnector
- Fuse switch
- Load breaking switch

Earth switch

Fault-making switch

Truck

Three-state switch

6.15.24 SWICHING_DEVICE_PURPOSE

A selector type of a combo box to select the purpose of the switching device.

The configurable attribute has just an informative purpose since, by the default, it is not used by any function within LIB 5xx. The attribute is stored into process object database (the 25th or the 26th character of the RX attribute) for user defined purposes, for example, for special search condition on the alarm or event list.

Circuit breakers:

- Circuit breaker
- Main busbar circuit breaker
- Main busbar A circuit breaker
- Main busbar B circuit breaker
- Reserve busbar circuit breaker
- By-pass circuit breaker
- Bus-section circuit breaker
- Bus-coupler circuit breaker

Disconnectors:

- Disconnector
- Main busbar disconnector
- Main busbar A disconnector
- Main busbar B disconnector
- Reserve busbar disconnector
- By-pass disconnector
- Line disconnector
- Line disconnector A
- Line disconnector B
- Bus-section disconnector
- Bus-section A disconnector
- Bus-section B disconnector
- Bus-section C disconnector
- Bus-coupler disconnector

Earth switches:

- Earth switch
- Main busbar earth switch
- Main busbar A earth switch
- Main busbar B earth switch
- Reserve busbar earth switch
- By-pass earth switch
- Line earth switch

- Line earth switch A
- Line earth switch B
- CT-section earth switch

Fault-making switch

Truck

Three-state switch

6.15.25 SWITCH_SECTION

A selector type of a combo box to select the switch section for the truck and the three-state switch. If SWITCHING_DEVICE_TYPE is some other than truck or 3-state switch, the configurable attribute is disabled, and the value has no meaning.

In Power Process library this attribute is not relevant, but for backward compatibility reasons it is not removed.

6.15.26 INDICATION_TYPE

A selector type of a combo box to select the type of the position indication process object(s). By default, the process object type is defined as a double indication (DB).

The purpose of this configurable attribute is to support other data type than double indication as the position indication of the switching device. The supported data types are:

- Double indication (DB)
- Single indication (BI)
- 2 x Single indications (2 x BI)
- Analog input (AI)

The interface between the picture function and the process object database is always with double indication object, that is, there is an additional process object(s) receiving information and then passing it to double indication.

6.15.27 MOTORIZED

An ON/OFF type of a toggle button to select if the switching device is motorized and remotely controllable. By default, the switching device is defined as a motorized object.

By selecting Motorized option, the Process Object Tool (POT) will create a set of process objects for control purpose, and all configurable attributes related to control are being enabled.

6.15.28 CONTROL_TYPE

A selector type of a combo box to select the control method and the type of the control process object(s). If the configurable attribute MOTORIZED is not selected, this attribute is disabled, and the value has no meaning.

The purpose of this configurable attribute is to support other control methods and process object types than the conventional one. The conventional control method depends very much on the configurable attribute STATION_TYPE since there is a certain default method for each protocol to control the switching device. The supported control methods are:

- Secured command with four binary outputs
- Secured command with two binary outputs
- Secured command with two digital outputs
- Secured command with a single binary output (*)
- Secured command with a single analog output
- Secured command with five binary outputs
- Direct command with a single binary output

(*) Control method not supported with SPA and ANSI protocols

Normally, A RTU 2xx with RP-570 uses secured command with a single binary output.

6.15.29 CONTROL_PULSE_LENGTH

An input field for integer value to set the pulse length of the control pulses. The configurable attribute is enabled, if the attribute MOTORIZED is set and CONTROL_TYPE is secured control with two digital outputs.

6.15.30 CONTROL_BITS

A vector type of an input field for integer values to specify the bits to be used to send a control command to the switching device. The configurable attribute is enabled if the attribute MOTORIZED is set and CONTROL_TYPE is a secured control with an analog output.

The user must specify two or four elements for that vector. If two elements are specified, the first element represents the bit for Open execute while the second element is for Close execute.

When four elements are used, the first represents the bit for Open select, the second represents Close select, the third represents Execute and the fourth represents Cancel.

When five elements are used, the first represents the bit for Open select, the second represents Close select, the third represents Open execute, the fourth represents Close execute and the fifth represents Cancel.

The bit vector shall be given to the SCT in the following way: (0,1) or (9,10,11,8) or (0,0,1,2,3).

6.15.31 AUXILIARY_PLUG

An ON/OFF type of a toggle button to select whether the device utilizes an auxiliary plug or not. If the SWITCHING_DEVICE_TYPE is truck or three-state switch, the configurable attribute is disabled, and the value has no meaning.

The process object for an auxiliary plug informs if the control cable from the switching device to the cubicle has been disconnected, that is, if the switching device is fully racked out from the cubicle. In that case, no switching device symbol is shown in the single line diagram.

The process object can also be used as an internal tag without the process connection. In that case, the control dialog asks the auxiliary plug state when the object state is intermediate.

6.15.32 INTERLOCKING_BYPASS

An ON/OFF type of a toggle button to select whether the switching device has interlocking bypass capability. This attribute is enabled only when IEC 61850-8 protocol is selected.

6.15.33 SYNCHROCHECK_BYPASS

An ON/OFF type of a toggle button to select whether the switching device has synchrocheck bypass capability. This attribute is enabled only when IEC 61850-8 protocol is selected.

6.15.34 QUALIFIERS

Qualifier is a value that is written to the QL attribute of process object when writing the control command. The qualifiers are given as a vector of integers. Minimum length is 0 (default) and maximum length 5. The vector items have the following meanings:

Item Meaning

1 = Qualifier for open select

For example, (1,2,3) defines qualifier 1 for open select, qualifier 2 for close select and qualifier 3 for open execute.

It is not possible to give empty items e.g. (,,,,3), but all the values must be given, e.g. (0,0,0,0,3).

6.15.35 MIN_POSITION

The minimum position of the tap changer.

6.15.36 MAX_POSITION

The maximum position of the tap changer.

6.15.37 TRANSFORMER_TYPE

Selection for the 2-winding or 3-winding transformer.

6.15.38 DIRECT_TAP_POS_SETTING

An ON/OFF type of a toggle button to select whether it is possible to directly control tap changer to a certain tap position instead of stepwise raise or lower operations.

6.15.39 OPERATION_MODE

Selection for the operation mode, where regulator is running. Depending on this attribute the following indications/settings are possible in control dialog:

- Single
  - Auto
  - Manual
- Parallel - Master/Slave

    - Parallel
    - Single
    - Auto/Manual, when regulator is in Single mode
    - Master/Slave, when regulator is in Parallel mode
- Parallel - Negative Reactance Principle
    - Parallel
    - Single
    - Auto
    - Manual
- Parallel - Minimizing Circulating Current
    - Parallel
    - Single
    - Auto
    - Manual

Single/Parallel setting is available only with IEC 61850 protocol, otherwise it is indication only.

With IEC 61850 protocol Master/Slave setting can be enabled/disabled by setting the IU attribute of indexes 33 and 34 to in use/not in use.

6.15.40 AUTO_MANUAL_IND_TYPE

Either BI or DB type of process object can be created for the auto/manual indication. This attribute must be configured before the process objects are created.

6.15.41 TYPE_MEAS_'number'

The type of the measurement. The type identifies the measurement for the creation of database (indexes, object texts, units...). In case no suitable alternative can be found, the indexes 27...37 can be used. The indexes 27 ... 32 are for user defined analog (AI) objects, and the indexes 33 ... 37 are for user-defined pulse counter (PC) objects.

The database objects are not created until the Process Object Tool has been used to create them.

6.15.42 FOLDER_'number'_TITLE

The name of the measurement to be shown in the measurement dialog. Attribute is optional, if it is not given, the measurement is referred to as Meas. 'number' in the dialog.

If the configurable attribute TYPE_MEAS_'number' is not configured, this attribute is disabled, and the value has no meaning.

6.15.43 MEAS_'number'_DECIMALS

The number of decimals used when presenting the measurement value in the dialog.

If the configurable attribute TYPE_MEAS_'number' is not configured, this attribute is disabled, and the value has no meaning.

6.15.44 OPC_LN_INSTANCE_1

Specifies the OPC Logical Node names used as IEC 61850 Instance information for the measurement type 1. The default is VECTOR("MMXU1").

6.15.45 OPC_LN_INSTANCE_2

Specifies the OPC Logical Node names used as IEC 61850 Instance information for the measurement type 2.

The default is VECTOR("MMXU1").

6.15.46 OPC_LN_INSTANCE_3

Specifies the OPC Logical Node names used as IEC 61850 Instance information for the measurement type 3.

The default is VECTOR("MMXU1").

6.15.47 OPC_LN_INSTANCE_4

Specifies the OPC Logical Node names used as IEC 61850 Instance information for the measurement type 4.

The default is VECTOR("MMXU1").

6.15.48 TYPE_OF_SIGNAL'nr'

A selector type of a combo box to select the type of the signal. The list of supported input signals is as follows:

- Binary Input (BI)
- Double Binary (DB)
- Analog Input (AI)
- Linked

If any of the first three choices is selected, the process objects are created by using the Process Object Tool. When Linked is selected, the logical name and the index of the process object where the signal is connected to has to be given. When an empty string is selected, it means that the signal is discarded by the alarm indicator.

6.15.49 SIGNAL'nr'_TEXT

This text is used as an identifier of the signal on event list, alarm list, printouts, and so on.

6.15.50 ALARMING_STATE_OF_SIGNAL'nr'

The alarming state of the signal is defined by this attribute. The value of the attribute is related to the type of the signal. The following examples clarify how this attribute should be defined:

- Type of signal is BI, alarming state is 1 value to be set: (1)
- Type of signal is DB, alarming states are 0 and 3 values to be set: (1,0,0,1)
- Type of signal is AI.

6.15.51 LINK_SIGNAL'nr'_TO_LN

The logical name of the process object to which the signal is linked.

6.15.52 LINK_SIGNAL'nr'_TO_IX

The index of the process object to which the signal is linked.

6.15.53 AR_WITH_MULTIPLE_CYCLES

An ON/OFF type of a toggle button to select the object type for the indication auto-reclosing in progress. The configurable attribute is enabled if the attribute P_OBJECT_LN has been configured. The database process objects will be created either as a binary input or as an analog input depending on the selection.

6.15.54 SHOW_AR_IN_USE

An ON/OFF type of a toggle button to select if the tag is shown when the auto-reclosing is in use. The configurable attribute is enabled if the attribute P_OBJECT_LN has been configured.

By default no tag is shown. This is useful when the auto-reclosure is normally OFF and a warning should be produced if the auto-reclosing is in use.

6.15.55 SHOW_AR_NOT_IN_USE

An ON/OFF type of a toggle button to select if the tag is shown when the auto-reclosing is not in use. The configurable attribute is enabled if the attribute P_OBJECT_LN has been configured.

By default, no tag is shown. This is useful when the auto-reclosure is normally ON and the warning should be produced if the auto-reclosing is not in use.

6.15.56 LINK_AR_TAG_TO_LN

The logical name of the database process objects of the auto-reclosing module. The maximum length of the text is 10 characters.

Allowed characters are the letters A-Z, all digits and the underscore (_). However, an object name cannot begin with a digit or an underscore.

The configurable attributes LINK_AR_TAG_TO_LN and LINK_AR_TAG_TO_MODULE are utilized to create a link between existing auto-reclosing objects from the other Object Browser function (for example, auto-reclosing module) and the auto-reclosing tag. It even supports linking of user defined process objects as LINK_AR_TAG_TO_MODULE is an editable selector.

6.15.57 LINK_AR_TAG_TO_MODULE

An editable selector type of a combo box to select the type designation of the auto-reclosing module. The configurable attribute is enabled if the attribute LINK_AR_TAG_TO_LN has been configured.

By default, all auto-reclosing modules from the SPACOM family have been implemented for the users to choose from. The supported modules are:

SPCT 2C5

SPCT 2D38

index 70 HSAR started/reset (E1/E2)

SPCT 2D46

index 91 HSAR forward started/reset (E1/E2)

During the relay installation and configuration, the following indexes have to be created and SPACOM events have to be enabled as a minimum requirement for the auto-reclosing tag by using the SPACOM Relay Configuration Tool:

SPCT 2C5

index 65 auto-reclosing started/reset (E1/E2)

SPCT 2C17

index 151 auto-reclosing started/reset (E1/E2)

SPCT 5D54

index 32 auto-reclosing interrupted by ARINH (E30)

SPTO 6D3

index 16 auto-reclosing interrupted (E7)

If there is no appropriate auto-reclosing module to choose, or if freely defined process objects are going to be used, it is possible to link those objects by using the following syntax:

"IDENT","INDX1","REPR1","INDX2","REPR2", ...

IDENT = module ID - (used only as a comment)

As an example, the auto-reclosing module SPCT 2D46:

"SPCT 2D46","91","F_REPR/F_ARRUN4","95","F_REPR/ARRUN5"

The type designation of the auto-reclosing module is not necessary, it is just additional information for users.

6.15.58 LINK_TRIP_TAG_TO_LN_x

The logical name of the database process objects of the protection module x (x=1..5). The maximum length of the text is 10 characters. Allowed characters are the letters A-Z, all digits and the underscore (_).

The configurable attributes LINK_TRIP_TAG_TO_LN_x and LINK_TRIP_TAG_TO_MODULE_x are utilized to create a link between the existing protection tripping indication objects from the other picture function (for example, protection module) and the trip indication tag. It even supports linking user defined process objects as LINK_TRIP_TAG_TO_MODULE_x, which is an editable selector.

6.15.59 LINK_TRIP_TAG_TO_MODULE_x

An editable selector type of a combo box to select the type designation of the protection module x (x=1..5). The configurable attribute is enabled if the attribute LINK_TRIP_TAG_TO_LN_x has been configured.

By default, all protection modules from the SPACOM family have been implemented for the users to choose from. The supported modules are:

SPAM 05x

The minimum requirement for the tripping tag to function is that the following indexes have been created and SPACOM events have been enabled by using the Relay Configuration Tool:

SPAM 05x

index 122 thermal trip (E5/E6)

SPCD 2D55

index 26 TS2 (E31/E32) Note!

SPCD 3C21

index 32 dI> L1 (E3/E4)

SPCD 3C22

index 34 dI> L2 (E3/E4)

SPCD 3C23

index 36 dI> L3 (E3/E4)

SPCD 3D53

index 44 TS2 (E27/E28) Note!

SPCJ 1C7

index 12 I0> trip (E3/E4)

SPCJ 1C8

index 16 I0> trip (E3/E4)

SPCJ 2C30

index 23 dI0> trip (E1/E2)

SPCJ 3C3

index 20 I> trip (E3/E4)

SPCJ 3C48

index 57 I> trip (E3/E4)

SPCJ 3D35

index 51 TS2 (E23/E24) Note!

SPCJ 4D24

index 44 TS2 (E25/E26)

SPCJ 4D28

index 80 TS2 (E25/E26) Note!

SPCJ 4D29

index 57 TS2 (E25/E26)

SPCJ 4D34

index 78 TS2 (E41/E42)

SPCJ 4D36

index 39 TS2 (E25/E26)

SPCJ 4D40

index 95 TS2 (E33/E34)

SPCJ 4D44

index 108 TS2 (E25/E26)

SPCJ 4D61

index 135 TS2 (E31/E32) Note!

SPCP 3C2

index 62 U>, U>> trip (E3/E4)

SPCS 2D26

index 95 TS2 (E19/E20) Note!

SPCS 2D32

index 107 TS2 (E22/E30) Note!

SPCS 2D37

index 22 TS2 (E23/E24) Note!

SPCS 3C4

index 29 I0> trip (E3/E4)

SPCS 4D11

index 121 TS2 (E25/E26)

SPCS 4D12

index 134 TS2 (E25/E26)

SPCS 4D13

index 147 TS2 (E25/E26)

SPCU 1C1

index 12 U> trip (E3/E4)

SPCU 1C6

index 16 U0> trip (E3/E4)

SPCU 1D39

index 126 TS2 (E23/E24) Note!

SPCU 1D47

index 142 TS2 (E23/E24) Note!

SPCU 3C14

index 20 U> trip (E3/E4)

SPCU 3C15

index 20 U< trip (E3/E4)

SPEF 3A2

index 101 I> alarm (E3/E4)

As a presumption, these modules are using TS2 as the indicator of latched output.

If there is no appropriate protection module to pick, or if freely defined process objects are going to be used, it is possible to link those objects by using the following syntax:

"ID","I1","R1","I2","R2","I3","R3", ...

ID = module ID - (used only as a comment)

If the module does not support sending of an event of a latched output relay, the vector element I1 has been left as an empty element, that is " ".

As an example, non-directional overcurrent and neutral earth-fault of REF 54x feeder terminal:

"REF54x","","F_REPR/F_LATCH1","1","F_REPR/F_TRIP3",-

The type designation of the protection module x (x=1..5) is not necessary, if there is no logical name for the protection module in question.

6.15.60 LINE_INDICATOR_COLORED_BY

A selector type of combo box to select the color of line indicator.

The color of the line indicator is determined either from DMS600, Network Topology Coloring or with manual settings.

6.15.61 ADD_CAUSE_IX

The ADD_CAUSE_IX attribute defines the index of a process object created for showing the possible failure of select and execute operations. AddCause functionality is available only with IEC 61850-8 protocol.

The AddCause feature is informative, that is, the AddCause process object value does not prevent an execution operation in control dialog.

Values -35- -30, -25- -22, 1- 18, 1000-1018, 2000-2018 and 3000-3018 have a descriptive text for AddCause, otherwise a number is displayed as a description.

Assuming that the AddCause value is 1001, the following text is shown if the AddCause value changes after the selected operation: Control failed: Not supported

(1001).

Assuming that the AddCause value is 1020, the following text is shown if a descriptive text is not available: Control failed: 1020.

6.15.62 LIB_OBJECT_TYPE

A selector type of a combo box to select the lib object type for the object. The lib object type can be used to configure:

- Index convention of process objects.
- Value semantic of process objects.
- Predefined custom methods.
- Object specific process object.

6.15.63 DEFINE_ITEM_NAME

An ON/OFF type of a toggle button to select whether the filling of OPC item path (to IN attribute) is done when the process objects are created. This attribute is enabled only when IEC 61850-8 protocol is selected.

6.15.64 OPC_ITEM_PREFIX

Defines the OPC path to the subnetwork level. This path has to be same as the object path in IEC 61850 OPC server configuration. When this attribute is filled, the information can be utilized when automatically generating External OPC DA Client configuration.

6.15.65 OPERATOR_PLACE_HANDLING

Defines the operator place handling for the Bay, Switch Device and Tap Changer objects. Depending on the configuration the operator place can be switched between Local/Station/NCC.

- None, there is no operation place checking
- MicroSCADA Internal, fictive process object is created for operator place
- Loc, process object for device level L/R switch is created
- LocSta, process objects for device level Station/NCC switch are created
- Loc and LocSta, process objects for device level L/R switch and device level Station/NCC are created

Loc and LocSta are available only for IEC 61850-8 protocol.

6.15.66 MULTILEVEL_CONTROL

By enabling multi level control, process object for indication multilevel operator place is created. Multilevel control means that both Station and NCC are authorized operator places.

6.16 Group configuration

The power process libary functions share some attributes, which can be configured at the same time for several selected objects. Multiple objects can be configured with the Group Configuration Tool. Group configuration is useful, for example, when the authorization group must be changed for a several objects after they have already been installed.

The Group configuration is intended for experienced users.

To configure multiple objects:

1. Go to the Object Navigator.
2. Click Process Objects and select **View/Process Objects in Groups**.

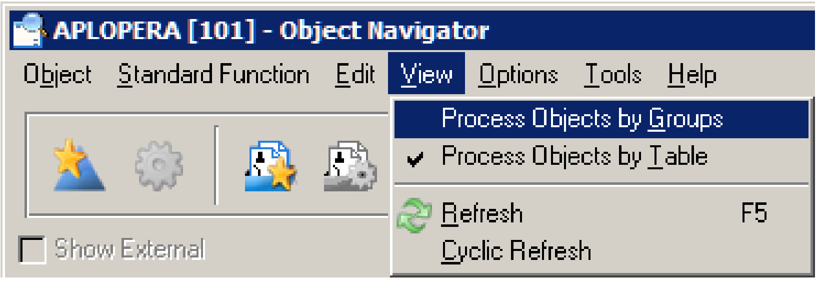


Figure 81: Viewing object by groups

3. Select the process objects groups you want to configure.
4. Select **Standard Function/Group Configure...**to open the Group Configuration Tool.

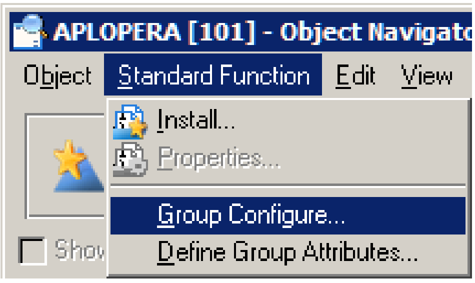


Figure 82: Opening the Group Configuration Tool

5. Click **OK** in the warning dialog box. Type the new values for the attributes in the text boxes, or select one of the shared selector values from the combo boxes.

Only values accepted by all the selected Power process libary functions can be selected for the selector type attributes.

Clicking the enable/disable button next to each text or combo box includes or excludes the attribute to/from the configuration. See Figure 83.

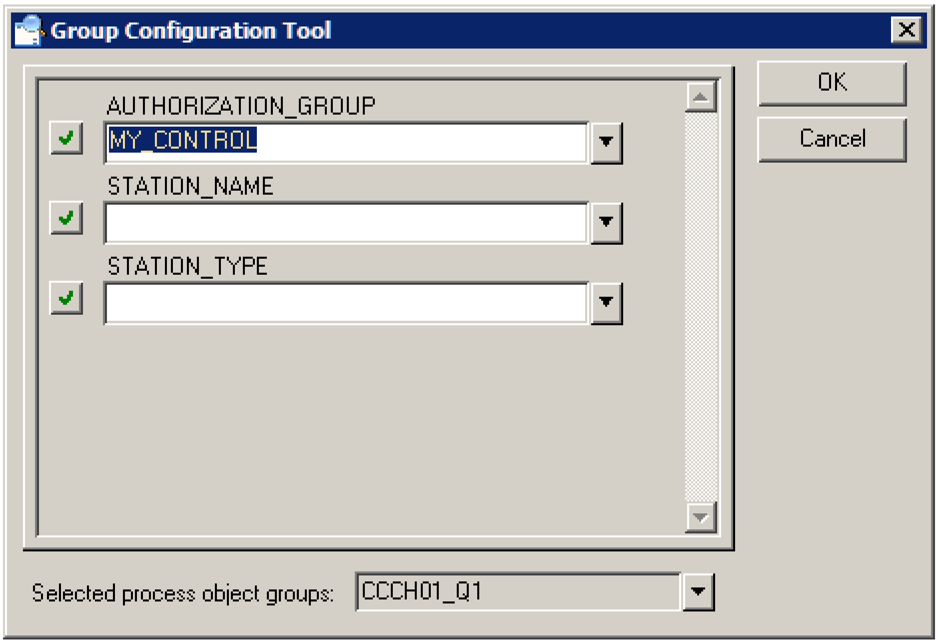


Figure 83: Group Configuration Tool

6. Click **OK**. A dialog for confirming the group configuration action appears. Click **OK** to accept the new configuration. Once the group configuration operation is finished, a log dialog appears showing the missing attributes and/or attributes excluded from the configuration.

> When changing the value of attributes related to object identification, for example STATION_NAME, the new value is written both to the CD attribute and to the correct part of the OI-attribute.

To define the group configuration attributes:

1. Select **Standard Function Configuration/Define Group Attributes...**. The Group Configuration Attributes dialog, shown in Figure 84 appears.

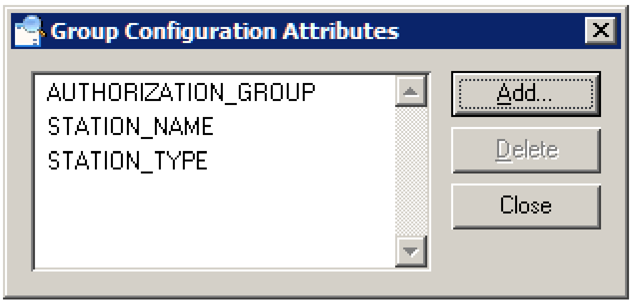


Figure 84: Group Configuration Attributes dialog

2. To add an attribute, click **Add** and type the name of the new attribute to the text box.
3. To delete an attribute, select the attribute to be deleted and click **Delete**.
4. When the attributes to be configured are defined, click **Close** to close the dialog.

6.17 Object types

6.17.1 General

There may be project specific needs to make adaptations to control dialogs or to the structure of the process database. With object type, it possible to affect to the behavior of standard Power Process Library objects both in the configuration phase and during runtime. The object type can be used to configure:

- Index convention of process objects.
- Value semantic of process objects.
- Predefined custom methods.
- Device specific process object.

The basic principle of object type is shown below:

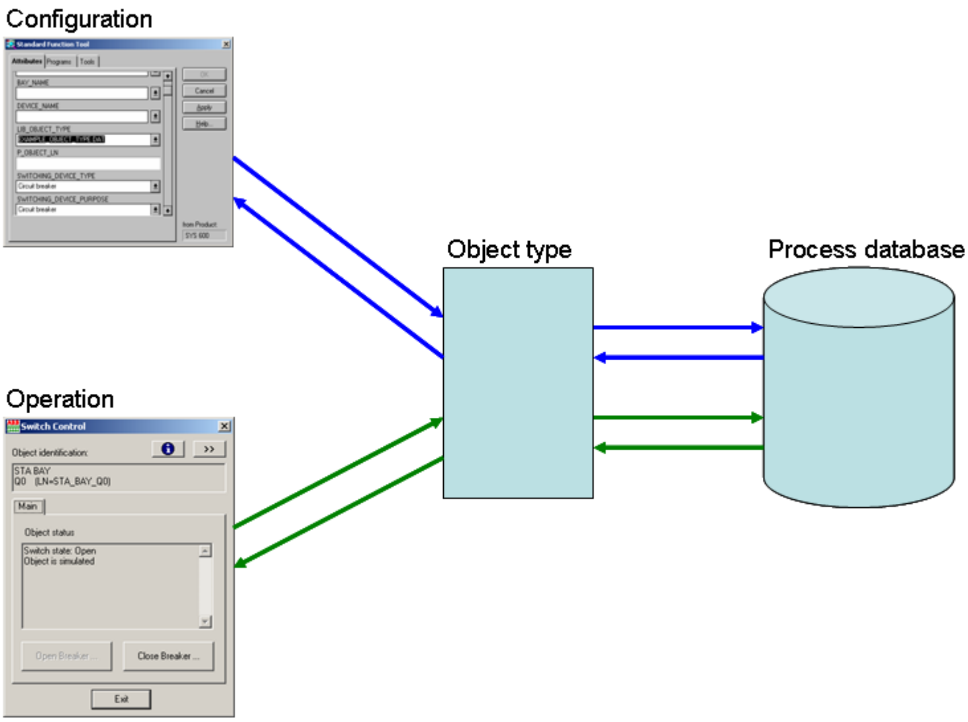


Figure 85: The main principle of object type

6.17.2 Object type file

Object type is an ASCII file with suffix .DAT. Object types are located in directory \sc\sa_lib \defaults\Object_Types or alternatively in \sc\apl\'apl name'\APLMOD4\Defaults \Object_Types.

The content of the object type file is a SCIL list dump. From object type of view, the content can be divided as attributes and methods.

6.17.2.1 Attributes

Attributes can be used for defining the index convention of the object (at run time). This is useful, for example, if the control dialog is connected to process database built with convention, where indexes differ from the ones that are created by Power Process Library function.

Attributes can also be used to override selections made in Standard Function Tool. All attributes of the standard function do not have to be defined in the object type.

6.17.2.2 Methods

Object type methods can be divided into two categories: ones that are executed in configuration phase and the ones that affect the run time functionality of the control dialog. The run time methods are dependent on the Power Process Library function.

Object type methods can be written directly into the object type file or method can be a reference to an external file which is executed with #DO READ_TEXT … statement.

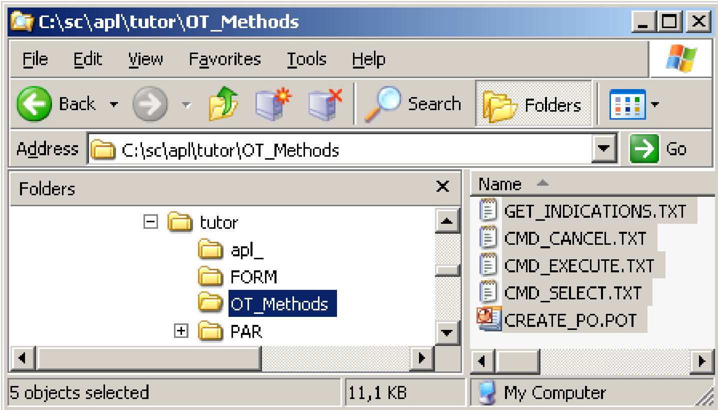


Figure 86: Example location of object type files

Characters ' and " have to be duplicated in when writing methods. For example

```
...
METHOD_OPEN_SELECT=VECTOR("#DO READ_TEXT(""SAGR_OBJT/
CMD_SELECT.TXT"")"),-
...
```

The name of the object type file should be carefully considered in order to avoid duplicate file names. File name could contain e.g. some device specific or protocol specific information, for example REx5xx_IEC104.DAT.

6.17.3 Object type for switch device

6.17.3.1 Attributes

Object type for switch device has the following attributes to modify the index convention.

Table 120: Attributes for index convention

Process object meaning	Default Power Process Library index	Attribute in object type
Position indication	10	INDICATION_DB_IX
Open select command	11	OPEN_CMD_IX
Close select command	12	CLOSE_CMD_IX
Execute command	13	EXECUTE_CMD_IX
Open execute command	13	EXECUTE_OPEN_CMD_IX
Close execute command	13	EXECUTE_CLOSE_CMD_IX
Cancel command	14	CANCEL_CMD_IX
Device control block	15	EXT_CNTR_BLK_IX
Open interlocked	16	OPEN_ILOCK_IX
Close interlocked	17	CLOSE_ILOCK_IX
Cause of interlocking	18	ILOCK_CAUSE_IX
Selection on monitor	19	SEL_ON_MON_IX
Command event	20	CMD_EVENT_IX
Auxiliary plug	21	AUX_PLUG_IX
Breaker Synchrocheck	22	SYN_IND_IX
Device open blocked	41	OPEN_BLOCK_IX
Device close blocked	42	CLOSE_BLOCK_IX
Bay L/R switch	10	BAY_LR_OBJECT_IX
Station Remote switch	10	STA_LR_OBJECT_IX

For example, if the open and close interlocked signals would come from indexes 71 and 72, the definition in the object type is done in a following manner.

```
LIST(-

OPEN_ILOCK_IX=71,-

CLOSE_ILOCK_IX=72)
```

6.17.3.2 Methods for adding/modifying process objects

These methods can be utilized when a process database is created with Process Object Tool.

Table 121: Methods for modifying process database

Method	Meaning

METHOD_INIT_SIGNALS	Additional signals defined in object type. These signals appear in process object tool (POT) for the standard function. This can be used, for example, for creating additional event signals for the standard function. The method must return a vector each item contains list defining an object:
	`@v_Object_Type_Process_Object = VECTOR(-`
	`LIST(LN=%LN,IX=201,...),-`
	`...`
	`LIST(…))`
	`#return %v_Object_Type_Process_Objects`

METHOD_CUSTOMIZE_OBJECTS	The method is executed after the SAI_SSW.POT file is executed. Therefore, vector %OBJECTS contains all process objects to be created, so it is possible to edit final objects that were created. Variable %OBJECTS is a vector, whose each element is a list containing the configuration of the process object. Note that modifying %OBJECTS does not change the object configuration, because objects are already created.

6.17.3.3 Method for value conversions

Depending on IED or protocol, certain signals can have a different value for same meaning. For example, some IED can send value 1 when Device is interlocked, whereas some other IED can send value 0 in a same situation.

In order to support different value conventions in control dialog, it is possible to define value conversions in the object type.

Table 122: Method for value conversion

Method	Meaning
METHOD_GET_INDICATIONS	This method is executed when user opens control dialog and used for value conversion.

Table 123: Signals supporting value conversion

Process object meaning	Return variable	Interpretation in control dialog
Position indication	INDICATION_OV	0=intermediate, 1=open, 2=closed, 3=faulty
Device control block	EXT_CNTR_BLK_OV	0=control allowed, 1=blocked
Open interlocked	OPEN_ILOCK_OV	0=interlocked, 1=control allowed
Close interlocked	CLOSE_ILOCK_OV	0=interlocked, 1=control allowed
Cause of interlocking	-	-
Breaker Synchrocheck	SYN_IND_OV	0=synchrocheck inhibits, 1=control allowed
Bay L/R switch	BAY_LR_OBJECT_OV	0=control allowed, 1=not authorized to control the bay

6.17.3.4 Methods for customized control

Table 124: Control methods for switch device

Method	Meaning
METHOD_OPEN_SELECT	This method is executed when the user makes the Open Select operation in control dialog.
METHOD_CLOSE_SELECT	This method is executed when the user makes theClose Select operation in control dialog.
METHOD_OPEN_EXECUTE	This method is executed when the user makes the Open Execute operation in control dialog.
METHOD_CLOSE_EXECUTE	This method is executed when the user makes the Close Execute operation in control dialog.
METHOD_CANCEL	This method is executed when the user makes the Cancel operation in control dialog.

6.17.3.5 Method for changing texts

Table 125: Method for text conversions

Method	Meaning
METHOD_GET_STRING_REPLACEMENTS	By using this method, dialog texts can be changed in object type. When the control dialog is opened, this method is executed if it exists, and strings returned by it are applied to control dialog.

It is possible to define object type specific localization of the strings for control dialog according to user language, see REPLACE_STRINGS.DAT, where languages EN and FI are implemented as an example.

String ID:s are returned by method METHOD_GET_STRING_REPLACEMENTS of object type. To replace, e.g. id 100, define attribute ID100 in list returned by the method. See REPLACE_STRINGS.DAT how to do this.

Contents of REPLACE_STRINGS.DAT:

```
LIST(METHOD_GET_STRING_REPLACEMENTS=VECTOR(-
"#if %language==""EN"" #then #block",-
"@strings=LIST(-",-
"ID296=""Hand..."",-",-
"ID297=""Auto..."",-",-
"ID272=""Switch state is Hand"",-",-
"ID273=""Switch state is Auto"",-",-
"ID315=""Are you sure you want to change the switch to Hand state?"",-",-
"ID316=""Are you sure you want to change the switch to Auto state?"")",-
"#block_end",-
"#else #if %language==""FI"" #then #block",-
"@strings=LIST(-",-
"ID296=""Kasi..."",-",-
```

```
"ID297=""Automaatti..."",-",-

"ID272=""Kytkin on kasitilassa"",-",-

"ID273=""Kytkin on automaattitilassa"",-",-

"ID315=""Oletko varma että haluat muuttaa kytkimen kasitilaan?"",-",-

"ID316=""Oletko varma että haluat muuttaa kytkimen automaattitilaan?"")",-

"#block_end",-

"#RETURN %strings"))
```

Resulting appearance when REPLACE_STRINGS.DAT object type is used for switching device:

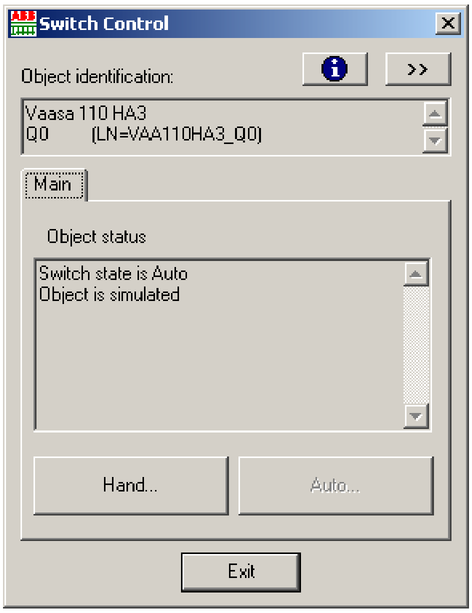


Figure 87: Changing titles in control dialog with object type

6.17.3.6 Configuration

Once the object type description is made, it can be taken in use while installing Switch device.

In Standard Configuration Tool there is an attribute LIB_OBJECT_TYPE. The appropriate object type can be selected from the combo pop down.

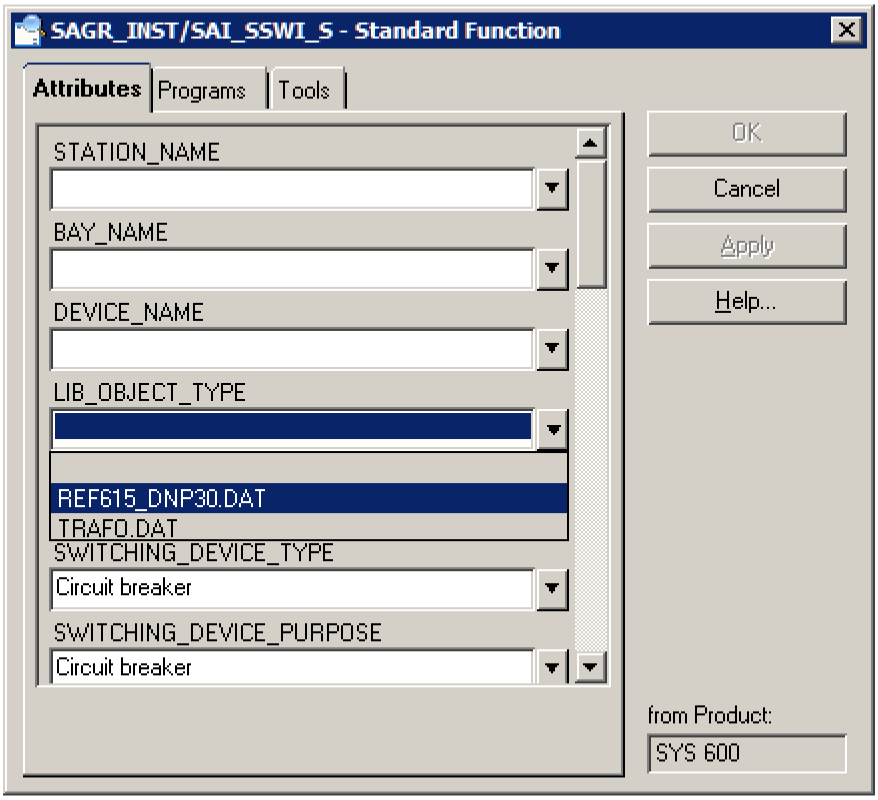


Figure 88: Selection of object type

If the object type has own control methods, CONTROL_TYPE attribute has to be configured "Use from Object Type". When this is selected object type has to have all control methods defined.Section 6.17.3.4.

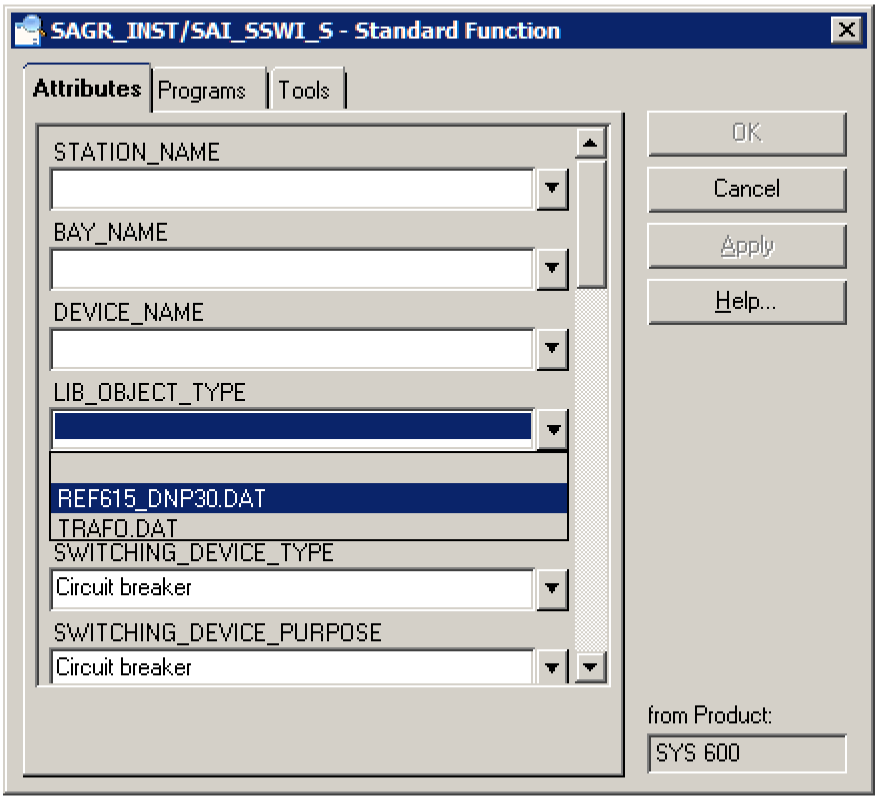


Figure 89: Selection of control type for object type

Section 7 Sequence Configurator

7.1 Overview

SYS600 Sequencer offers creation, execution and monitoring of switching device command sequences in MicroSCADA X. Sequencer has two separate tools one for creation and configuration of sequences (Sequence Configurator) and another tool for execution of sequences (Sequence Executor). This section describes sequence creation and configuration using Sequence Configurator. More details regarding Sequence Executor tool can be found from SYS600 Operation Manual.

A Sequence is a collection of one or more steps/commands for switching type devices. These sequences can be created and configured by a MicroSCADA user with sufficient engineering rights using Sequence Configurator/SeqConf tool. For each step of the sequence, the command to be executed is defined together with optional precondition and postcondition checks or delays. Currently supported commands are switching device open/close command or message box display. For information on how to execute sequences see SYS600 Operation Manual.

Sequences can also be started from internal and external triggers in addition to an interactive user. Internal triggers include event channel and Schedule function of the MicroSCADA X Calendar. Using external triggers, the sequences can be started from NCC via COM500i functionality or from DMS600.

Sequencer also supports creation and execution of sequences from external system via MicroSCADA OPC server interface. This is used by DMS600 for dynamic sequence creation and execution.

7.2 Terminology

This section defines the terminology used in relation to Sequencer

Sequence Configurator – a tool for creation and configuration of sequences

Sequence Executor - a tool intended for execution and monitoring of configured sequences

Sequence – List/collection of steps

Step – a configured command to execute or a message to display to an interactive user. The configuration of a step consists of:

- Precondition or command execution delay
- Command
- Postcondition or proceed with next step delay

Precondition/Postcondition – several comparisons of process objects with defined values. If the condition is fulfilled the sequence execution continues. If the condition fails the further sequence execution will be interrupted.

Command – can be of two types:

- Command to control a switching device as e.g. circuit breaker or disconnector: OPEN or CLOSE
- Message Box – show the defined message box for an interactive user. For other user types this type of command will be ignored during execution.

Validation check – predefined conditions that can be checked with Sequence Configurator.

Controllability check – predefined conditions that are automatically checked when loading a new sequence in Sequence Executor. These controllablity checks are also performed at the beginning of each sequence step execution.

SIF File - Every sequence is stored in SCIL database file with a .sif extension.

User Name – User Name shown in the Event List for sequence execution. The real user name is only shown for interactive users. For other sequence triggers, the trigger names such as DMS and SYS are used as User Names.

Sequencer Trigger Four trigger types are allowed to start and execute a sequence:

- INT – MicroSCADA Interactive User
- SYS – MicroSCADA command procedure or event channel
- DMS – DMS600
- COM – COM500i / NCC

Internal Triggers – MicroSCADA internal events that can start sequence execution. These include event channels and command procedures.

External Triggers – A trigger in which sequence execution is started from an external source such as from DMS or NCC

7.3 Installation and removal

Sequencer installation and removal can be done using LIB5 Application Initialization Tool. To Install Sequencer Select **Tools > Engineering Tools > Tool Manager >** in Monitor Pro and navigate to **Application Objects > Application Initialization Tool** in Tool Manager. Selecting Sequencer checkbox and clicking Prepare selected packages button installs sequencer and clearing the checkbox uninstalls it. However, Sequencer cannot be installed if not enabled in the applied license.

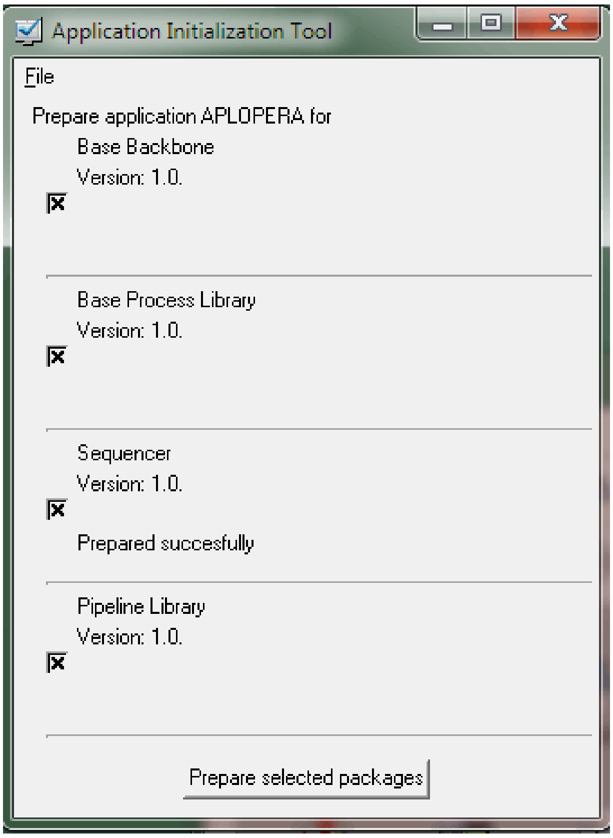


Figure 90: Sequencer installation

During installation the following line gets added to LIB5INIT_1:C command procedure

#exec epu_seq_main_mng_clear_queue:c

epu_seq_main_mng_clear_queue:c command procedure resets possible interrupted sequence executions due to abnormal system halt.

The following line gets added to LIB5INIT_H:C command procedure.

#exec epu_seq_main_continue_executing:c

The above command procedures ensures that running sequence will continue to execute when a switch-over occurrs in a Hot-Standby system. Both the above calls are removed automatically during sequencer uninstallation.

7.4 User Interface

Sequence Configurator can be launched from Tool Manager by Navigating to Monitor Pro **Tools>Engineering Tools>Tool Manager>Application Objects>Sequence Configurator**. See Figure 91 below.

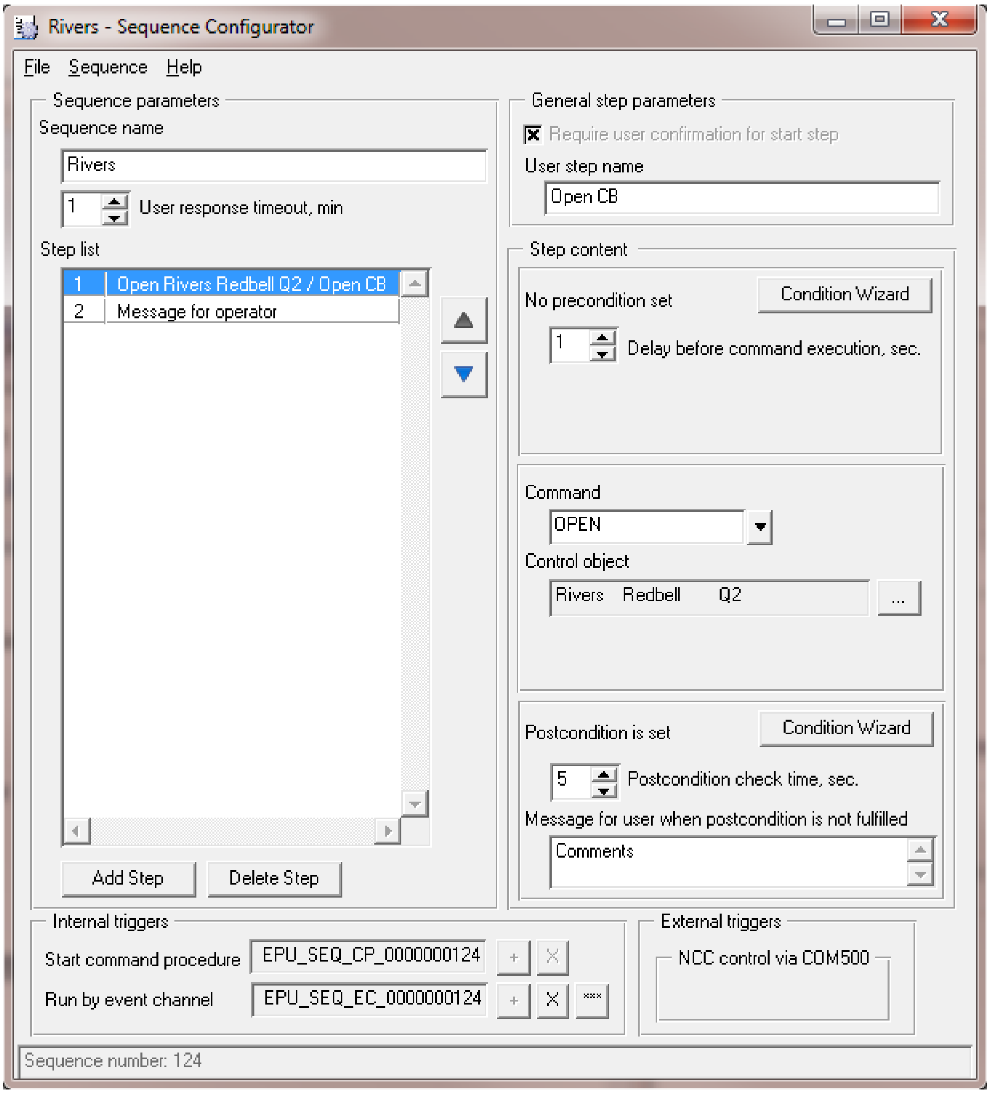


Figure 91: Sequence Configurator

Menu bar

The menu bar in Sequence Configurator consists of different menus.

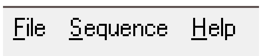


Figure 92: Sequence Configurator's Menu bar

The descriptions of the each menu functions are described in Table 126 below.

Table 126: Sequence Configurator's File menu functions

Menu	Function	Description
File	New	Prepares a new sequence instance
	New Based on Current Template	Creates a new sequence based on the opened sequence instance
	Open	Opens an existing sequence file
	Save	Saves a sequence
	Duplicate	Duplicates a currently opened sequence (Makes a copy of the currently opened sequence)
	Delete	Used to delete sequence files from the list of available sequences
	Exit	Exits Sequence Configurator Tool
Sequence	Validation Check	Makes a validation check to an already prepared sequence
	Modification Log	Shows timestamps and modification information about the available sequence files
Help	About	Information about the tool and system

7.5 Configuring a Sequence

7.5.1 Configuration Parameters

Sequences are used to control specific switching objects that are configured in sequence steps. Table 127 below lists all the sequence parameter which can be configured using the Sequence Configuration tool.

Table 127: Sequence configuration parameters and corresponding descriptions

Parameter	Description	Type	Values/Range
Sequence Name	Textual identification of the sequence	Text	1-30 character
User Response Timeout	Timeout for waiting user response during sequence execution. If no user response received within the timeout, the sequence execution is aborted	Integer	1-60 Minutes Default: 1 min
User defined step name	Textual identification of the step.	Text	1-30 character Default: empty
User confirmation	Definition if the steps will be executed automatically or if some user confirmation is needed before the step is executed	Boolean	True/False toggling checkbox
Table continues on next page			

Parameter	Description	Type	Values/Range
Precondition	Configurable precondition check. Condition is created with condition wizard. Condition can include multiple comparisons of process object values. Logical OR and AND operations can be used with comparisons to advanced conditions		Up to 10 OR branches. Each OR branch may have up to 10 AND conditions
Precondition Check Time	The retry time for the condition check. Check is repeated in 1 second interval until it is passed or the time limit is reached. If the time limit is reached and check has not passed, the check is considered as failed.	Integer	0-60 Seconds Default: 1 s
Delay Before Command Execution	Delay before command execution, if no precondition is configured.	Integer	0-60 Seconds Default: 1 s
Command	Command to be executed in the step. Command can be either switch object open/close operation or message output for the user. Command is configured using a command wizard	Integer	open or close command, message output
Postcondition	Configurable postcondition check (see precondition).	Integer	
Postcondition Check Time	The retry time for the condition check. Check is repeated in 1 second interval until it is passed or the time limit is reached. If the time limit is reached and check has not passed, the check is considered as failed. Value 0 means condition checked only once immediately after command execution	Integer	0-60 Seconds Default: 1 s
Delay After Command Execution	Delay after command execution, if no postcondition is configured.	Integer	0-60 Seconds Default: 1 s
Postcondition Check Failed Message	Message to be displayed to the user, when check is not fulfilled	Text	

7.5.2 Major Functions

The following major functions are available in Sequence Configuration Tool:

1. Create new empty sequence (see Figure 93 first item)
2. Create new sequence based on existing. Only sequences with Process Object references to one bay only can be used as source. In this case a list of bays containing the same amount and type of switching devices will be provided. Multiple bays can be selected to create sequences for the selected bays (see Figure 93 and Figure 94).
3. Create and save a new sequence as a copy from an existing (duplicate). Possible configured internal triggers are also created as new triggers to the new sequence ID.
4. Save sequence
5. Open sequence
6. Delete sequence

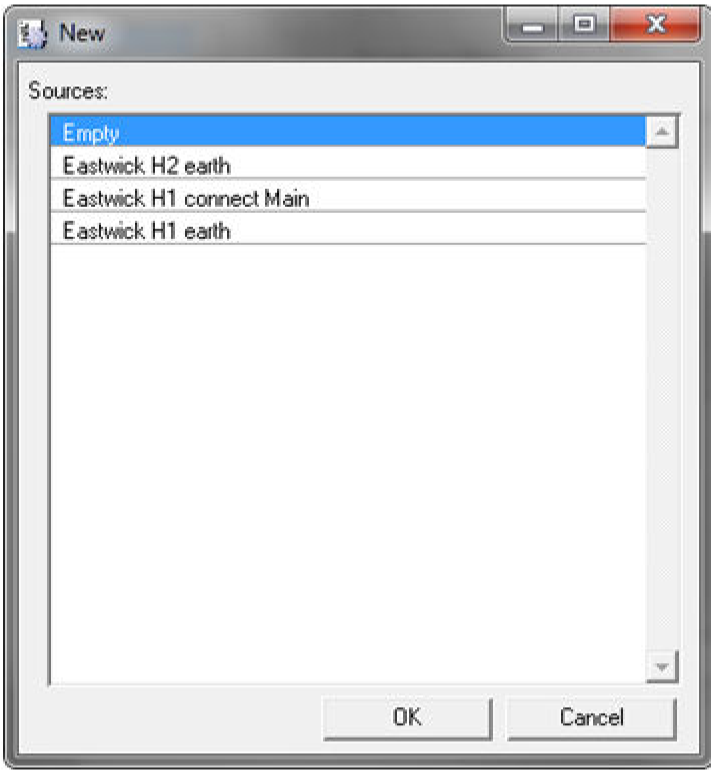


Figure 93: New sequence dialog

A sequence containing only Process Object links to one bay (based on OI) can be used as template to duplicate the sequence to other equivalent bays.

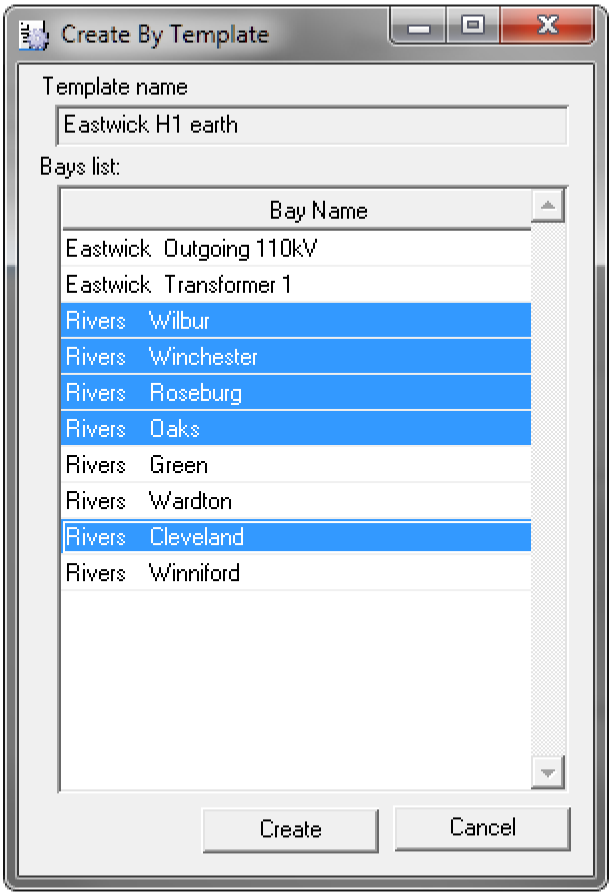


Figure 94: New sequence based on template

Sequence Configuration Functions

For each sequence the following configuration functions are available:

1. Edit sequence name
2. Edit user response timeout in minutes
3. Create, Edit, delete, move steps
4. Define internal and external triggers

Step Configuration Functions

For each step, the following configuration functions are available:

1. Edit step name
2. Edit user confirmation required flag. If the flag is set, user confirmation is required before the step is executed. This flag is used only when the sequence is executed by an interactive user.
3. Edit delay before command execution in seconds. This option is available in case no precondition is set
4. Edit precondition with the help of the condition wizard (see Figure 97)
5. Edit precondition check time in seconds if precondition is set
6. Edit text for INT user when precondition is not fulfilled
7. Edit step command, with the following options available:
   - CLOSE for closing switching object (see Figure 96)
   - OPEN for opening switching object (see Figure 96)
   - MESSAGE output

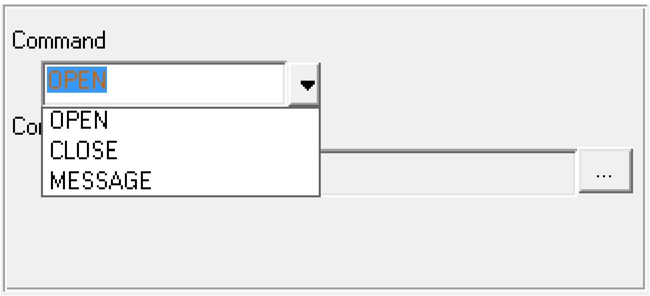


Figure 95: Command Type selection

8. Edit delay after command execution in seconds. This option is available in case no postcondition is set
9. Edit postcondition in help of the condition wizard (see Figure 97)
10. Edit postcondition check time in seconds if postcondition is set
11. Edit text for INT user when postcondition is not fulfilled

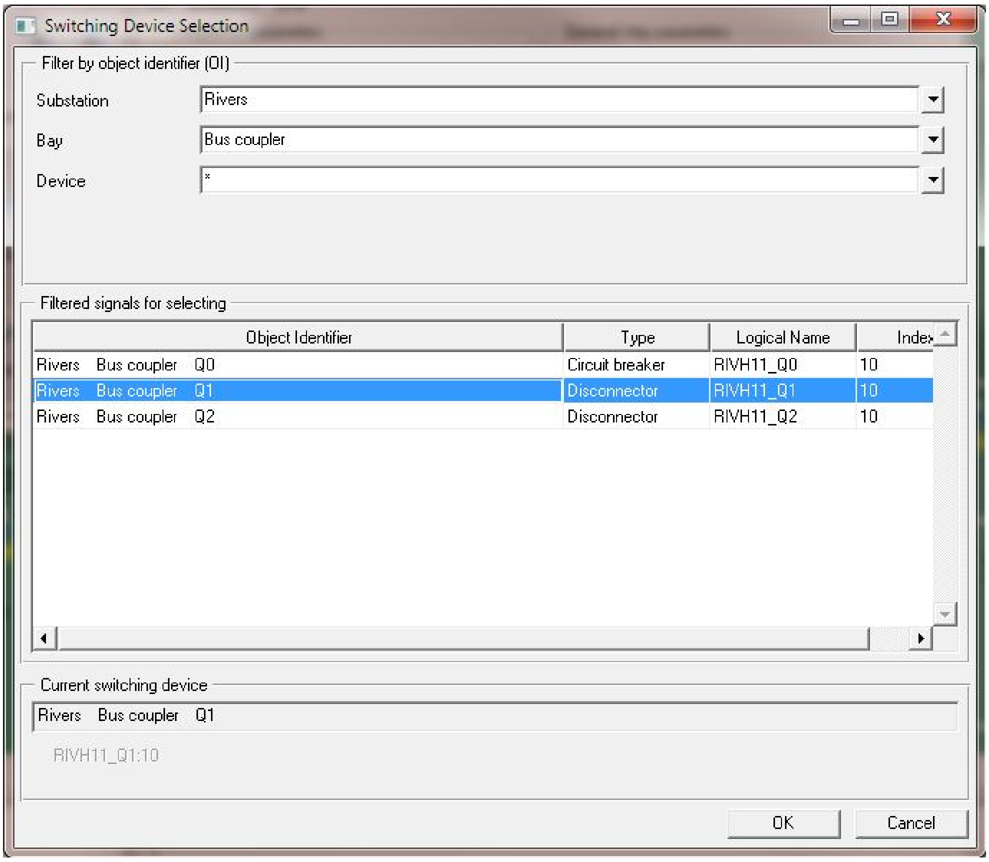


Figure 96: Switching Device Selection for commands

The precondition and postcondition can be constructed in help of the condition wizard.

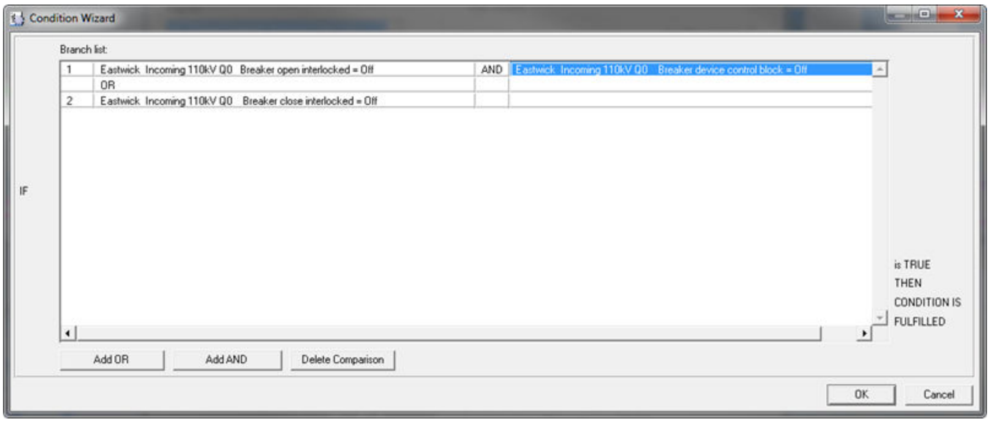


Figure 97: Condition wizard

7.5.3 Sequence Identification and Managing

ID_APL

ID_APL is an INTEGER, 1.. 2147483647 (MAX_INTEGER), sequence number, unique within MicroSCADA application. This value is used in the command to start the sequence from NCC via COM500i. ID_APL value is stored in the index table which is saved in \sc\apl\<apl name>\SEQUENCER\SEQUENCES\index.csv file.

Sequence ID_APL numbers are generated by the sequencer configuration tool. Sequencer will ensure that the ID_APL number of the executable sequence is unique within an application. If the user copies the sequence file from another application, Sequencer assignes a new unique and unused ID_APL for the copied sequence file.

7.5.4 Sequence Modification Log

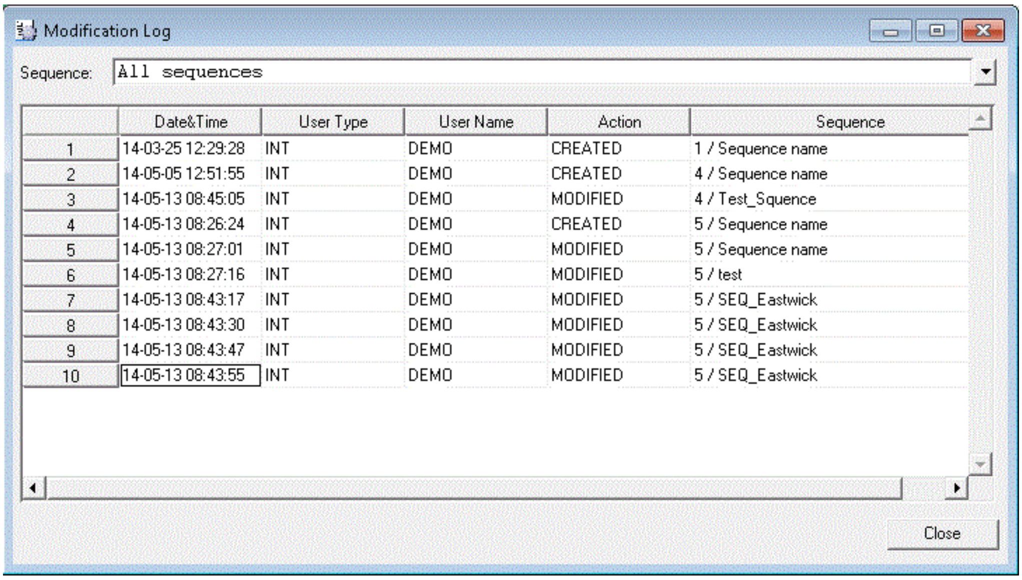
The sequence modification log is stored in the file

\sc\apl\<apl name>\PAR\APL\SEQ_CONF_HIS.sdb

The log consists of the following log details:

1. <Date&Time> - date and time of operation
2. < File_Name > - Name of SIF-file
3. < UserType >
   - INT – if modified by SeqConf
   - DMS –if procedure is created automatically by DMS request
4. < UserName > - User name
5. < OperationType > - includes the following operation type events:
   - Event “CREATED” is saved to the history table when a user or a DMS600 request creates a new sequence
   - Event “MODIFIED” is saved to the history table when a user modifies an existing sequence
   - Event “DELETED” is saved to the history table when a user deletes a sequence

The sequence modification log content can be viewed in Sequence Configuration tool. To launch Modification Log dialog Select **Sequence>Modification Log** from the menu.



	Date&Time	User Type	User Name	Action	Sequence
1	14-03-25 12:29:28	INT	DEMO	CREATED	1 / Sequence name
2	14-05-05 12:51:55	INT	DEMO	CREATED	4 / Sequence name
3	14-05-13 08:45:05	INT	DEMO	MODIFIED	4 / Test_Squence
4	14-05-13 08:26:24	INT	DEMO	CREATED	5 / Sequence name
5	14-05-13 08:27:01	INT	DEMO	MODIFIED	5 / Sequence name
6	14-05-13 08:27:16	INT	DEMO	MODIFIED	5 / test
7	14-05-13 08:43:17	INT	DEMO	MODIFIED	5 / SEQ_Eastwick
8	14-05-13 08:43:30	INT	DEMO	MODIFIED	5 / SEQ_Eastwick
9	14-05-13 08:43:47	INT	DEMO	MODIFIED	5 / SEQ_Eastwick
10	14-05-13 08:43:55	INT	DEMO	MODIFIED	5 / SEQ_Eastwick

Figure 98: Sequence modification log dialog

7.5.5 Configuring internal and external triggers

In addition to an interactive user, sequences can be started by internal or external triggers.

The internal and external trigger settings can be configured only for saved sequences. When a new sequence is not saved the following message will be shown:

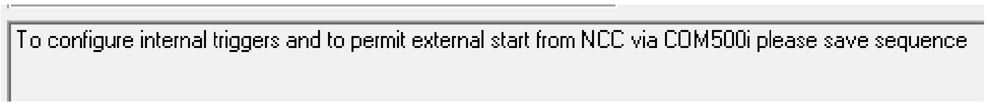


Figure 99: Trigger information box

7.5.5.1 Internal Triggers

Internal triggers are based on these application objects:

- Command procedure to start sequence execution for example to run from calendar tool
- Event channel to start command procedure (EPU_SEQ_CP_<ID_APL>:C) sequence execution

If the user creates the event channel the command procedure is created automatically.

All application objects (command procedure and event channel) have predefined names:

- Command procedure: EPU_SEQ_CP_<ID_APL>:C
- Event channel: EPU_SEQ_EC_<ID_APL>:A

See below example for standard sequence with ID_APL=1.

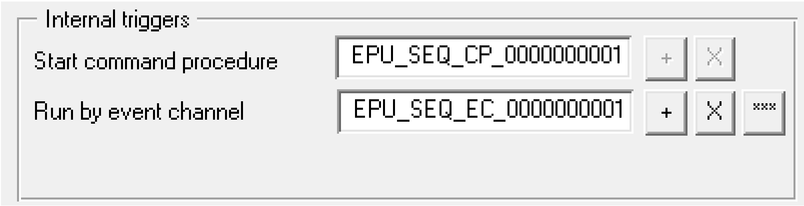


Figure 100: Internal Trigger Configuration

See details in Table 128 Below.

7.5.5.2 External triggers

External triggers can be used e.g. to start a sequence from NCC via COM500i.

There is a check box for each configured NCC connection in COM500i, that is used to allow (when checked) or deny the execution of the sequence from the corresponding NCC.

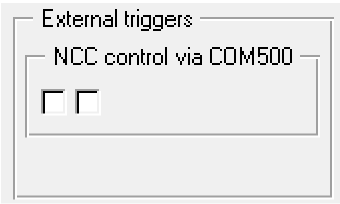


Figure 101: Configuring external triggers

For user types INT and SYS the external trigger permissions are not checked. See details in Table 128 Below.

Table 128: Summary of sequence execution triggers

Trigger source	Internal User Name	Type	Activation	Output Results Available
INT	<Interactive user name>	Interactive User	Sequence Execution Tool	• Sequence Execution Tool • Event and Alarm List • SYS600 Notify • Execution log
SYS	SYS	Internal trigger	• By Command procedure • By Event Channel	• Event and Alarm List • SYS600 Notify • Execution log
DMS	DMS	External trigger	Command from DMS	• Event List and Alarm List • SYS600 Notify • Execution log
COM	NCC Name/NCC Number	External trigger	Command from NCC via COM500i up to 8 directions	• Event and Alarm List • SYS600 Notify • Execution log

7.5.6 Configuring for Hot Stand-by system

When a sequence starts by an interactive user (”INT” User Type), sequence execution gets aborted during HSB takes over. However, when a sequence starts by internal trigger (SYS user) or by an external trigger (“COM” or “DMS” user type), sequence execution continues automatically on HSB take-over.

Note: During installation procedure SCIL-code is already added in LIB5INIT_H:c for Sequencer HSB to work:

#exec epu_seq_main_continue_executing:c

;THIS STRING IS CREATED AUTOMATICALLY.

;PLEASE DO NOT EDIT IT MANUALLY

7.5.7 User Authorizations

The level of access rights is defined in help of the SEQ_HANDLING authorization group.

If the SEQ_HANDLING group does not exist the GENERAL group will be used to determine the access levels. The details of authorization levels are explaind in Table 129 Authorization level.

Table 129: Authorization level

Level	Sequence Configuration	Sequence Execution
0	Opening the tool is not possible	Opening the tool is not possible
1	Opening the tool is not possible	View mode for current state
2,3,4,5	Editor rights	View and execute sequence mode

7.6 Sequence Validation and Controllablity Checks

Sequencer has two kinds of checking methods namely validation and controllability checks. The checkings are explained in the following sections in detail.

7.6.1 Sequence Validation Check

Validation checking ensures that the sequence fullfils the valid startup requirments e.g. needed Process Objects and standard function configuration data exists. Validation checking can be performed manually from Sequence Configurator's menu by selecting **Sequence>Validation Check**. When an interactive user selects a sequence to run it in Sequence Executor, validation is checked before the selected sequence is loaded for execution. If validation checking fails the sequence cannot be loaded.

After validation checking is completed, user can see three type of checking messages in Validation Check dialog:

- OK – Value or Validation checking passed
- FAIL – Unexpected values
- ERR – Validation checking error

When a validation check fails, the checking can be repeated from validation check dialog by pressing the **Repeat Check** button.

7.6.2 Sequence Controllablity Check

Sequence controllablity check is done in addition to validation check and it is done only during execution. Every single step of the sequence will be checked before the execution will start.

Controllability check includes:

- All station, bay and device local/remote keys and operation place indications are in the expected position
- Switching Device to be controlled is not interlocked or blocked by the process
- All Process Objects included:
  - must exist
  - are in use
  - have proper Object Status
  - are not blocked

The failed checks are visible in Figure 102.

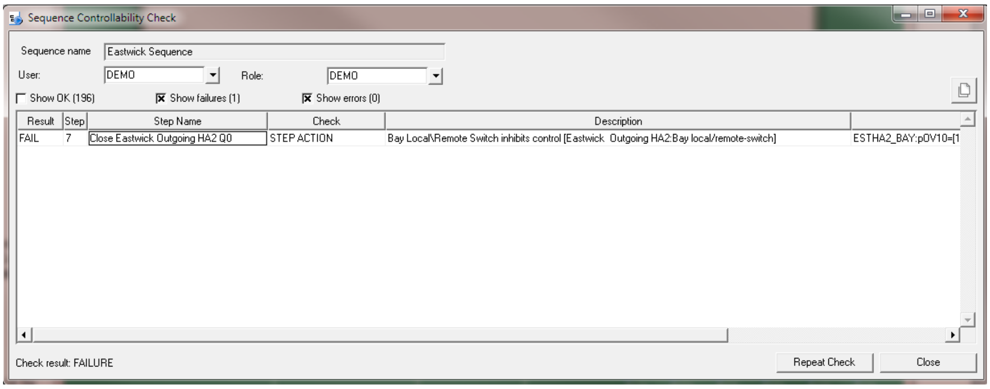


Figure 102: Controllability Check Dialog showing a failure due to Bay local/remote switch inhibition of control

Section 8 Creating a single-line diagram

Display Builder is an editor that lets the user create complete graphical interfaces. It is the primary tool that is used when Process Displays are created. The user can draw the displays and map the signals before or after process objects have been created to the process database, except that the Object Browser requires an existing and running application.

Symbols representing the process objects can be created to an empty display from the palette, and the signal mappings can be done manually. Alternatively, the symbols and mappings can be created with the Object Browser. After adding the symbols for process objects and busbars to the display, the network topology coloring can be built manually or by using Display Builder's automatic functions. For more information on the Display Builder, see SYS600 Process Display Design.

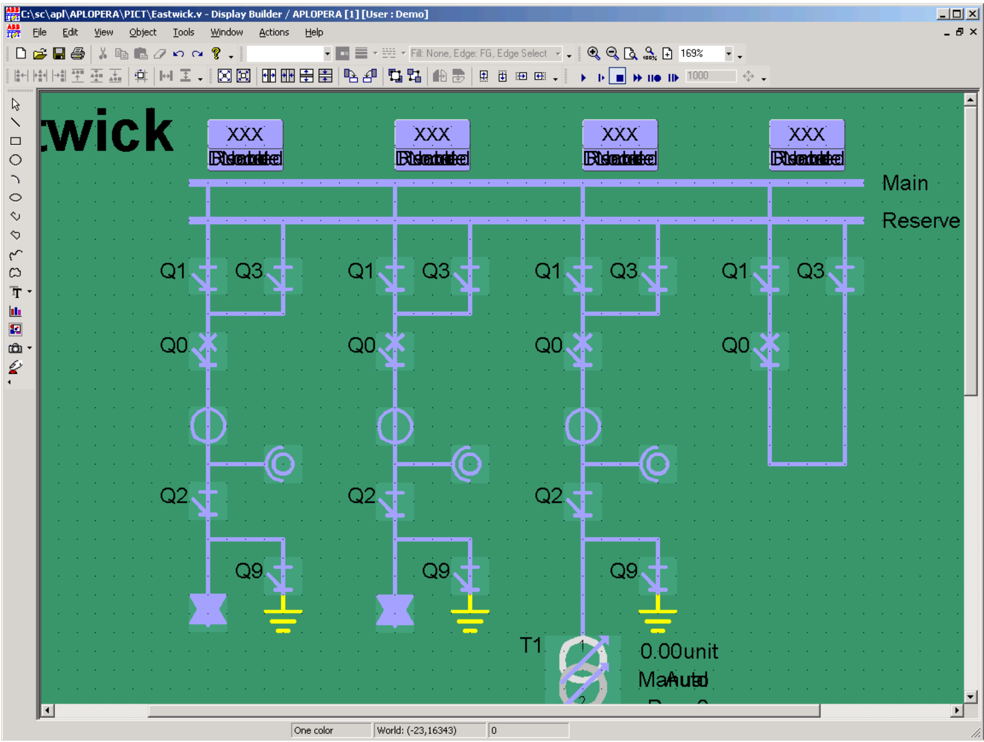


Figure 103: Main window of Display Builder

8.1 Connecting signals

The data variable properties can be edited by selecting **Edit/Data Variables**, see Figure 104. Usually, the Object Browser tool automatically generates the data variables for symbols that are dragged into the display from the object browser tree, and connects the symbol to the variables. Sometimes, it is necessary to edit these settings manually.

In the data source settings, the application number can be set to all data variables that exist in the data source. This makes it possible to create more than one data source in the same picture and define the application for them separately.

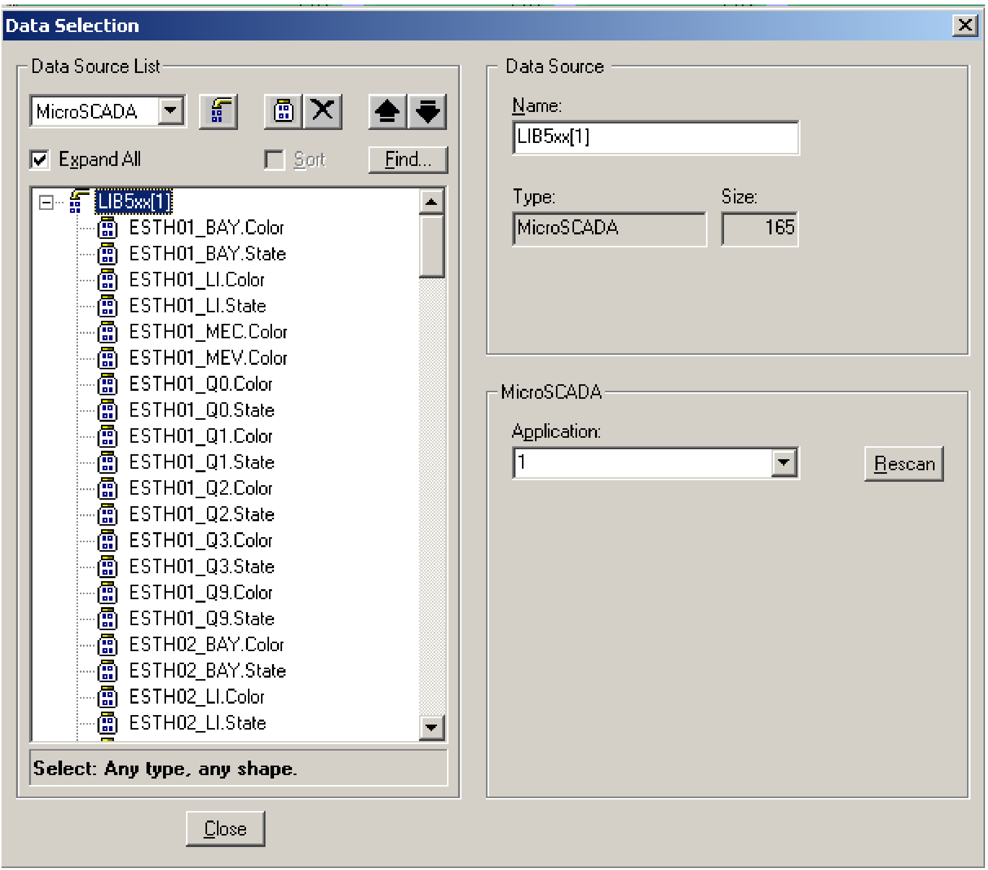


Figure 104: Creating data source in Data Selection dialog

8.1.1 Creating data sources

To create a data source:

1. In the Display Builder, open the **Data Variables** dialog by selecting **Edit/Data Variables**.
2. Select MicroSCADA from the New Data Source drop-down menu and click the **Add New Data Source** button.
3. Give a name for the new data source in the **Name** field.
4. In the **MicroSCADA** field, select the application number from the **Application** drop-down menu.
   This application number is applied to all data variables that are created under this data source.
5. Click **Rescan**.

8.1.2 Creating data variables manually

To create data variables manually:

1. In the Display Builder, open the **Data Variables** dialog by selecting **Edit/Data Variables**.
2. In the **Data Variables** dialog, select the MicroSCADA data source from the data source hierarchy and click the **Add New Data Source** button.

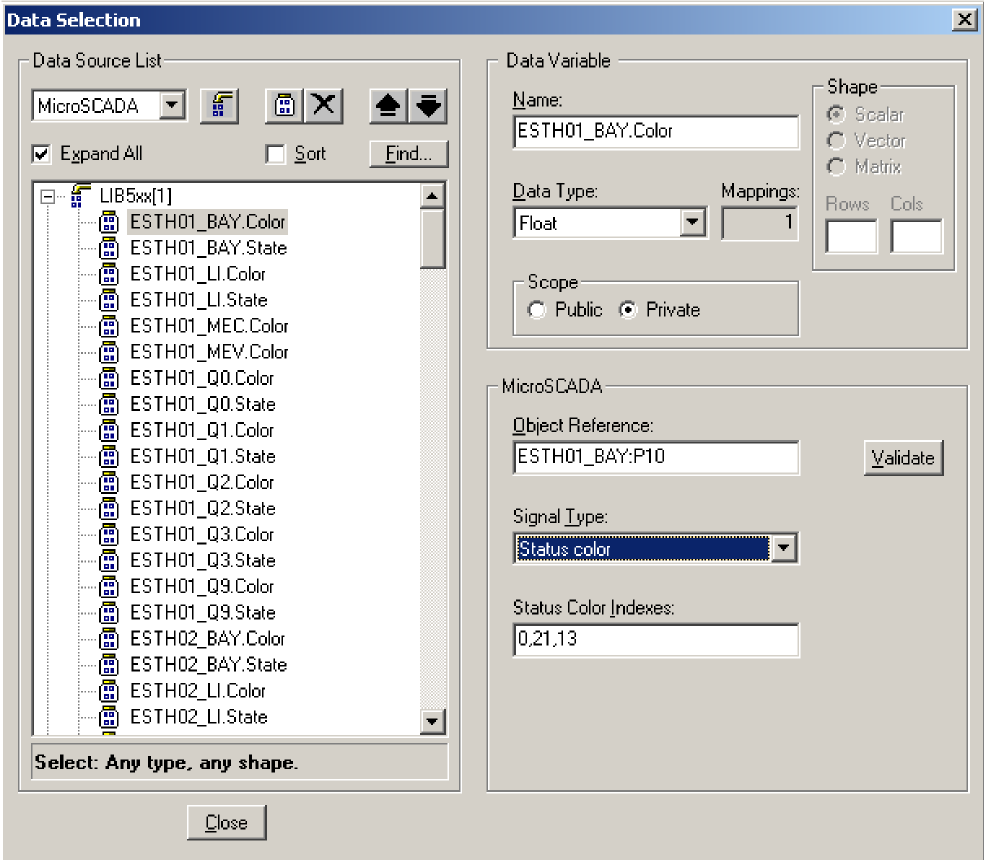


Figure 105: Creating data variables in Data Selection dialog

3. Define the data variable settings in the **Data Variables** field.
   These settings are related to a so called view data variable.
4. Set the type to **Float** and scope to **Private**.
5. Click **Validate**.

8.1.3 Creating data variables with vector and matrix data

Data variables can also be created with vector and matrix data from MicroSCADA.

To create data variables with vector and matrix data:

1. In Display Builder, open **Edit**-> **Data Variables**.
2. Create a MicroSCADA data source and connect it to an application.
3. Create a data variable to MicroSCADA data source.
4. Give an object reference in MicroSCADA data variable settings.
5. Select **Generic signal** as variable type.
6. Select **Create** and edit new file... in Run SCIL combo box.
7. Modify the result value.
8. Click **Apply**. Shape options are activated in Data Variable group.
9. Select vector or matrix shape. For vector data source, Length setting is activated. For matrix data source, rows/cols setting is activated.
10. Type appropriate length or row/column value.
11. Create a graph object and select **Graph Properties** from context menu of the object.
12. Select suitable type from **Types** tab sheet of Graph Properties window (e.g. surface type for matrix).
13. Connect the graph to data variable and set suitable range in **Variables** tab sheet.

When matrix data type is used, the data must be a vector set. Each inner vector represent one column. For example, when defining a matrix that has two rows and three columns, it is represented following way:

```
VECTOR(VECTOR(1,2),VECTOR(3,4),VECTOR(5,6))
```

As SCIL code this would be

```
#modify L_INPUT:V=LIST(VALUE=VECTOR(VECTOR(1,2),VECTOR(3,4),VECTOR(5,6)))
```

```
#RETURN %L_INPUT.VALUE
```

Following example returns vector of length 3:

```
#modify L_INPUT:V=LIST(VALUE=VECTOR(1,2,3))
```

```
#RETURN %L_INPUT.VALUE
```

When an OPC item with vector data type is used to launch SCIL code, the VALUE attribute is not created in %L_INPUT, and %L_INPUT. DATA_TYPE value is NONE, and therefore it cannot be used as an argument. Instead, create the value with SCIL. This way, it can be passed to data variable.

Only numerical items are supported in the vector and matrix data.

8.1.4 Connecting data variables to process database

To connect the data variable to the process database:

1. In the **Data Variables** dialog in the **MicroSCADA** field, type the process object reference in the **Object Reference** text field.
   The format is <process object logical name>:P<process object attribute><process object index>. For example, ESTH01_Q0:POV10.
   Type the object reference in upper case letters.
2. Select the Signal Type. For more information, see Section 8.2.
3. Click the **Validate** button to check that the name was typed correctly and the process object exists, see Figure 106.

Figure 106: Validation notice for existing process object

8.2 Data variable types

The data variable type is selected from the **Type of Variable** drop-down menu. Three types are available:

- State indication.
- Status color.
- Generic signal.

The State indication is used for integer state values. There is a possibility to swap values 1 and 2. If the signal value read from the process object is, for example, 1, it can be changed to 2 and written into the view data variable.

The Status color is used when the data variable should represent the status color value for the symbol. Internally, this is done by subscribing several process objects' attribute values from the OPC Server, and the status color is calculated based on the attribute values. Some attribute values are also subscribed from other process indexes, such as values for command event, selected on monitor status and control blocked signal. These values are subscribed from indexes <current index> + 10, <current index> + 9,<current index> + 3, respectively.

When Generic signal is used, no type conversion is done when moving data from the OPC Server to the data variable. This variable type can be used with measurement values and other floating point data types, or text values.

8.3 Customizing used process object indexes

It is possible to configure the process object indexes that graphical symbols use to indicate status colors. The status colors are based on the following process object values:

- State indication object
  The data source uses attributes of the object to define the device state, for example object state, blockings and alarms.
- Command event object
  The data source uses the OV attribute of the object to define if the device is selected for the operation.
- Selected on monitor object
  The data source uses the OV attribute of the object to define if the control dialog is open for the device in another monitor.
- Control blocked object
  By default, the index is set to 13. Data source uses the UB attribute of the object to define if the control is blocked.
- Alarming object
  The data source uses the alarm attributes of the object to define if the device is alarming.
- Externally blocked object
  The data source uses the OV attribute of the object to define if the device is externally blocked.
- Open blocked object
  The data source uses the OV attribute of the object to define if the device is open blocked.
- Close blocked object
  The data source uses the OV attribute of the object to define if the device is close blocked.

To configure the indexes, modify the SA_LIB_INDEXES attribute contents to the standard function configuration list that is located on the attribute CD. The attribute CD is located on the process object group of the standard function in Object Navigator. Run the rescan function to update the changes to the Process Display as described in the following.

For example, if the SA_LIB_INDEXES attribute is added to the standard function configuration, the attribute sets the command event index to 20, the selected monitor index to 19 and the control block index to 13:

SA_LIB_INDEXES="20,19,13"

After the SA_LIB_INDEXES attribute has been added or modified in the standard function configuration list on the CD attribute, update the Process Display with Display Builder as follows:

1. Select **Edit/Data Variables** to open the **Data Selection** dialog.
2. Select the appropriate data source on the left pane. See Figure 107.
3. Click **Rescan** in the Data Source Settings field to scan the configuration and to update and save the index configurations to the display, see Figure 107.

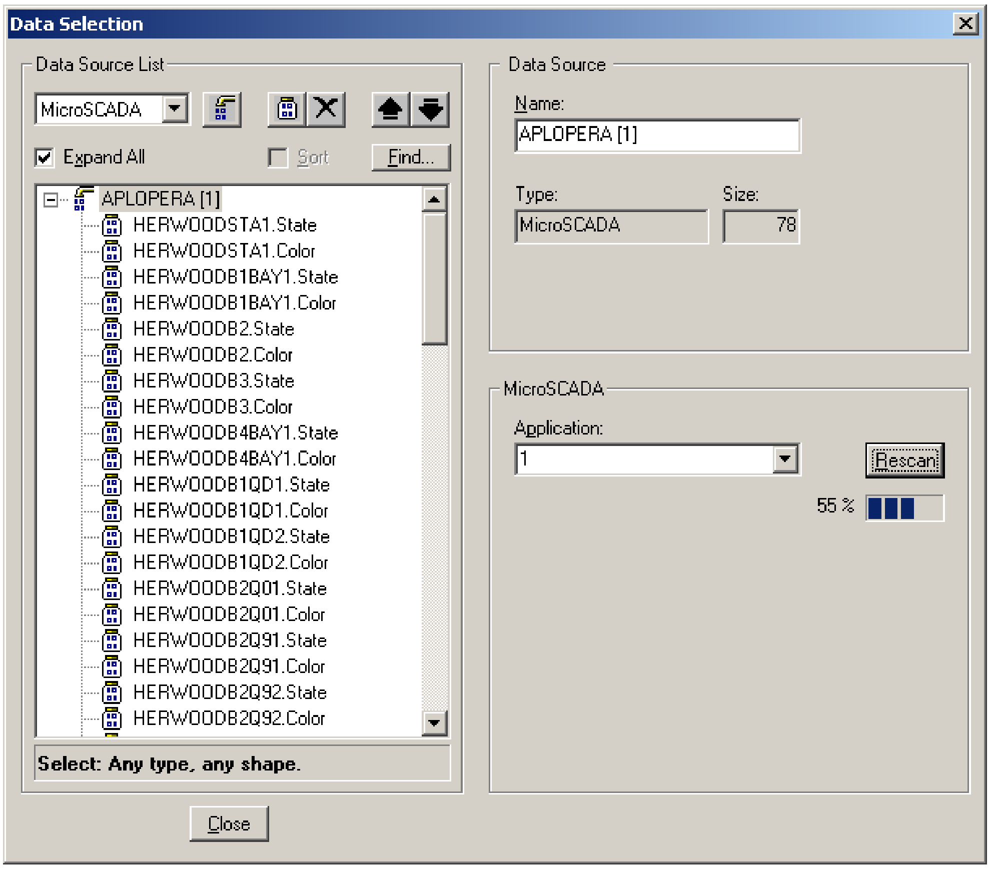


Figure 107: Rescanning configuration and updating and saving index configurations

4. Ensure that the indexes were updated correctly, see Figure 108.
   - Select a Status color Signal Type data variable on the left pane.
   - Used indexes are displayed in the Status Color Indexes field and semantics as a tool tip.

This applies only partly to Power Process switch device symbols. Some indexes are calculated by internal logic based on switching device type, indication type and control type. Logic is in command procedure BGU_CONTROL:C (GET_OFFSETS).

The number of used indexes may be different for separate standard functions. If, for example, some circuit breaker does not have open or close blocking process objects, the indexes are not added to the Status Color Indexes list.

Changes do not take effect before clicking **Rescan** and saving the picture in Display Builder.

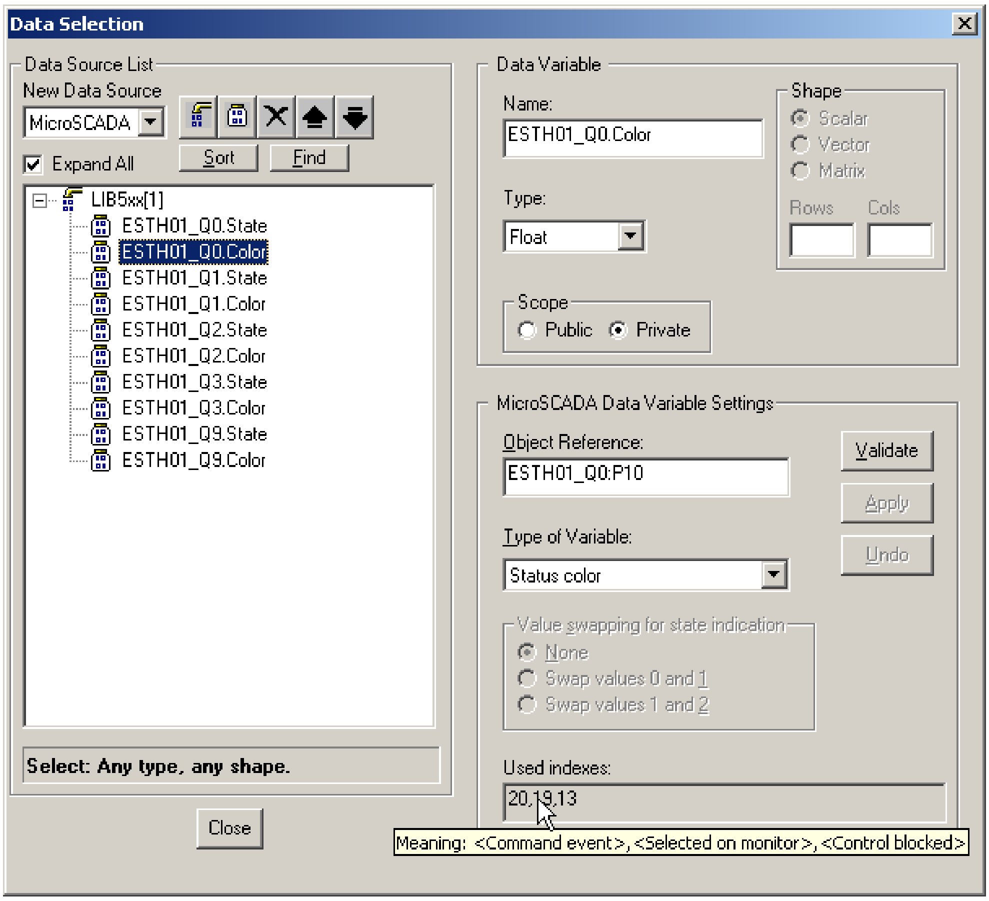


Figure 108: Ensuring that data variables are updated correctly

8.4 Status color logic

The status color is calculated based on several indexes and attribute values. Details for switching device can be found from the table below. Values are in priority order.

Table 130: Status color logic

Description	Condition (LN = Logical Name)	Color index
Unmapped device	'LN':PIU10 = 0	29
Not sampled	'LN':POS10 = 10	38
Selected, under command	'LN':POV20 = 1	42
Selected, on display	'LN':POV19 >= 1	41
Substituted, Internal	'LN':PSU10 = 1	43
Update blocked*, Obsolete or not updating	'LN':PUB10 = 1, 'LN':POS10 = 2 or 'LN':POS10 = 4..9	39
Substituted, External	'LN':PSB10 = 1	43
Blocked	'LN':PBL10 = 1	39
Invalid value	'LN':POS10 = 1	35
Alarm, Unacknowledged	'LN':PAR10 = 0	46
Alarm, Acknowledged	'LN':PAL10 = 1 and 'LN':PAR10 = 1	45
Table continues on next page		

Description	Condition (LN = Logical Name)	Color index
Control blocked*	'LN':PUB13 = 1	47
Alarm blocked	'LN':PAB10 = 1	34
Externally control blocked**	'LN':POV15 = 1 or (non IEC 61850) 'LN':POV15 = 2 or (IEC 61850) 'LN':POV41 = 1 or (open blocked) 'LN':POV42 = 1 (close blocked)	47
History blocked or Printout blocked or Reprocessing blocked	'LN':PHB10 = 1 or 'LN':PPB10 = 1 or 'LN':PXB10 = 1	34
Faulty time***	'LN':POS10 = 3	50
Not in use	'LN':PSS10 = 0 or 'LN':PIU = 0	29
Manually entered	NOT('LN':PSS10 >=2 and 'LN':PUN10 <> 0)	37
Normal		32

* Depending on control type index can vary.

** Available in 9.3 FP1. Depending on switching device type, open blocked and close blocked indexes can vary.

*** Status coloring is used only if the Use status colors setting is enabled in network topology coloring settings.

Table 131: Status color logic measurements

Description	Condition	Color Index
Measurement High Alarm	'LN':PAZ'IX' = 2	28
Measurement Low Alarm	'LN':PAZ'IX' = 1	24
Measurement High Warning	'LN':PAZ'IX' = 4	27
Measurement Low Warning	'LN':PAZ'IX' = 3	25
Warning	'LN':PAZ'IX' = 3 or 'LN':PAZ'IX' = 4	44

8.5 Configuring status signals

A status signal can be connected to another process object than the indication signal. For example, the status signal is read from index 110 in the ANSI environment. Control dialogs use the index automatically for switch devices with the ANSI protocol.

To take the feature in use in another protocol, define an object type that has the STATUS_IX attribute defined. Object types can be found on \sc\sa_lib\defaults\object_types. The object type is saved in this directory as a text file with a .dat extension.

File contents is a dump list containing the attributes, for example LIST(STATUS_IX=110). To take the object type in use, define the LIB_OBJECT_TYPE attribute in the standard function file.

The attribute value is an object type file name with the file name extension and without path, for example ANSI.DAT. Standard function files can be found on \sc\apl\<application name>\POBJCONF.

8.6 Customizing the Object Browser

Object Browser displays objects that can be inserted as a symbol to the Process Display, see Figure 6. It automatically creates symbols to the display and also establishes data mappings (static or dynamic) according to the configuration file. Insertable objects are defined in the Object Browser configuration file.

The configuration file is located in [drive]\sc\prog\graphicsEngine\ etc\ObjNav.ini. The file contains settings related to Object Browser and mappings between the process objects and the symbols. Both numeric and text values can be mapped to symbols. Table 132 shows the possible options for the Object Browser configuration file.

When the settings are modified in the configuration file, it is required to restart Display Builder. There is, however, a shortcut for this: select the root node in the browser and press CTRL+R to reload the settings from the configuration file. To apply general settings and filter settings, press the **Select** button once.

8.6.1 General settings

General settings are used to control some features of the Object Browser. Normally, it is not necessary to modify these.

The RX (or LN or ST) values are separated with the colon (:). Whitespace (except spaces) between separators and RX (or LN or ST) values are not skipped.

Table 132: Options for the ObjectBrowser configuration file

Key	Value
UsesReversedSingleIndications	Automatically swap values 0 and 1 for single state indications when dragging a switching device from Object Browser to the display. This setting is not used for local/remote switches. 0 = No swap (default) other value = Swap
UsesReversedDoubleIndications	Automatically swap values 1 and 2 for double state indications when dragging a switching device from Object Browser to the display. This setting is not used for local/remote switches. 0 = No swap other value = Swap (default)
UsePtSpecificMappingRxValues	Lists object types, which require process value type to be used in the object type identification, e.g. ASTA_12. PT attribute value is appended to the end of the object type.
LocalRemoteSwitchRxValues	Lists object types, which identify local/remote switches.

8.6.2 Filtering settings

Filtering settings can be used to show and hide process objects in the Object Browser.

The RX (or LN or ST) values are separated with the colon (:). Whitespace (except spaces) between separators and RX (or LN or ST) values are not skipped.

The process object must have RX attribute value to be shown in the Object Browser.

Table 133: Filtering settings

Key	Value
IncludeIx	Used for showing process objects only with a certain IX value in the Object Browser. This setting overrides IxMax setting. 0 = not used <n> = only include this IX value To include other IX values use the IncludeAllIndicesForRxValues and IncludedSpecifiedIndicesForRxValue settings.
IxMax	Used for hiding process objects with IX value greater than a certain value from the Object Browser. This setting is useful for skipping LIB510 event recording objects that all have IX > 100. 0 = not used <n> = Skip objects with larger IX To include larger IX values for some objects use IncludeAllIndicesForRxValues and IncludedSpecifiedIndicesForRxValue settings.
IncludeAllIndicesForRxValues	Shows all IX values for certain RX values in the Object Browser. This setting overrides IncludeIx setting.
IncludedSpecifiedIndicesForRxValue	Specified indices for the specified RX values, which are shown in the Object Browser. This setting overrides other RX and IX filter settings. Indices are separated with the comma (,). Indices are listed after RX value and the # character. Note that whitespace (except spaces) between separators are not skipped. Range of indices is defined with a dash e.g. 10-17
ExcludedRxValues	RX values that will be hidden from the Object Browser. This setting overrides the IncludeAllIndicesForRxValues setting.
ExcludeLNsStartingWith	Hides all LNs that start with these substrings from the Object Browser. This setting overrides both RX, IX, and ST filter settings.
ExcludedStValues	ST values that will be hidden in the Object Browser. This setting overrides RX and IX filter settings.

8.6.3 Symbol definitions

In addition to general and filtering settings, the configuration file contains symbol definitions for process objects. These are under [RX_Settings] and [RX_Mapping] sections in the configuration file. [RX_Settings] section defines what part of the RX value is used for identifying devices in the configuration file, and [RX_Mapping] section defines what graphical symbols are available for the process object/device. It is possible to add, remove, and modify symbol definitions in the configuration file.

Table 134: RX_Settings

Key	Value
StartIndex	Used to specify the start position of the substring in the RX attribute value that contains the object type. The default value is 23.
Length	Used to specify the length of the RX attribute substring that contains the object type. The default value is 4.
DetailedLength	Used to specify the length of the RX attribute substring that contains the detailed object type. The detailed object type can be used to differentiate devices that have the same value specified by Length setting but different value when more characters are taken to substring. The default value is 8.

In a SYS600 product, a device has an instance in the process database. The device has both static and dynamic data. This data is represented in the process display using graphical symbols. Graphical symbols available in the Object Browser for the device and the data linked to the symbol are configured to the configuration file. The syntax of the symbol definition is in its simplified form.

8.6.3.1 Single symbol

device_id = [symbol], e.g. QB01 = 02 - SA_IEC\Breaker.sd

8.6.3.2 Symbol group

device_id = [symbol_group], e.g. QE01 = <02 - SA_IEC\Disconnector.sd:0,0:IEC Earth Switch, vertical & 01 - SA_Common\Earth.sd::0,-1000:Earth symbol>

It is possible to define the following properties for symbols in the configuration file:

- Name
- Path to symbol file
- Coordinates
- Data mappings

8.6.3.3 Symbol Definition Syntax

The device_id element is the substring of the RX value of the device, specifically, the object type. Syntax for the symbol definition of a device and its data mapping to the graphical symbol is presented in the table below. The following notations are used in the table:

{1} = must have the preceding character/string

? = can be empty or existing

* = repeats

+ = once or more

^ = exclusive or

Table 135: Symbol definition syntax

<table>
<tr><th>Syntax</th><th>Regular expression/valid values</th><th>Description</th><th>Example</th></tr>
<tr><td>device_symbol_definition</td><td>[device_id]=[symbol_definition] ^ [device_id_reference]</td><td>This is the main definition.</td><td>WSYS = Server.sd</td></tr>
<tr><td>Syntax for device identification</td><td></td><td></td><td></td></tr>
<tr><td>device_id</td><td>[device_type]([_][device_value_type])?</td><td></td><td>QB01, ABAY_3, FMFUISTR</td></tr>
<tr><td>device _type</td><td>Any string value</td><td></td><td>PMEA</td></tr>
<tr><td>device_value_type</td><td>[3 | 6 | 9 |12]</td><td>3 - Single indication, 6 - Digital input, 9 - Analog input, 12 - Double indication. If this syntax is used, see also the UsePtSpecificMappingRxValues setting.</td><td>3</td></tr>
<tr><td>device_id_reference</td><td>[[]{1}[device_id][]]{1}</td><td>This is useful when a device has same symbols as another device.</td><td>[QB01], e.g. QB02 = [QB01]</td></tr>
<tr><td>Syntax for symbol definition</td><td></td><td></td><td></td></tr>
<tr><td>symbol_definition</td><td>(([single_symbol]^[symbol_group])[|])*</td><td>Single symbol or symbol group. Alternative symbols are separated using the | character.</td><td>Server.sd | Database Server.sd</td></tr>
<tr><td colspan="4">Table continues on next page</td></tr>
</table>

<table>
<tr><th>Syntax</th><th>Regular expression/valid values</th><th>Description</th><th>Example</th></tr>
<tr><td>single_symbol</td><td>[symbol path]{1}[:][data mapping]?[:][coordinate]?[:][symbol name]?</td><td>Symbol definition for a single symbol with other detailed information.</td><td>Value FG.sd : POV : 0,0 : Measurement Value</td></tr>
<tr><td>symbol_group</td><td>[<]{1} ([symbol][&]?)*{1} [>]{1}</td><td>Symbol definition for a symbol group.</td><td>Symbol1.sd:::Symbol 1 & Symbol2.sd:::Symbol 2</td></tr>
<tr><td>symbol path</td><td>Any valid file path. Also, a relative path according to graphics engine can be used.</td><td>Absolute or reference file path to the symbol.</td><td>06 - SA_Supervision\Monitor.sd</td></tr>
<tr><td>symbol name</td><td>Any string</td><td>The name shown for the symbol in the Object Browser.</td><td>Circuit Breaker ANSI Horizontal</td></tr>
<tr><td>coordinate</td><td>([-16732-16372][,][-16732-16372])?</td><td>XY-coordinate value to determine symbol coordinates. Used in symbol groups. Can be empty, default value is 0,0.</td><td>-1000,1000</td></tr>
<tr><td>Syntax for data mappings</td><td></td><td></td><td></td></tr>
<tr><td>data mappings</td><td>([data mapping]([,])?)*</td><td>Determines which static/ dynamic data is mapped to the data variables of the symbol.</td><td>POV, NPOV, POV[10], "REF630", 1234,-,, TPCX</td></tr>
<tr><td>data mapping</td><td>(([parent][\])?[data type]?[object type][attribute][\[]{1} [process object index][]]{1})? |
[constant data mapping] |
[empty data mapping]</td><td>Determines the path to process point or gives a constant, static value.</td><td>See [data mappings]</td></tr>
<tr><td>empty data mapping</td><td>[-]?</td><td>Determines whether the symbol data variable is skipped.</td><td>See [data mappings]</td></tr>
<tr><td>constant data mapping</td><td>(["][a-zA-Z0-9]*["]) | [0-9]*</td><td>Constant text or numeric mapping.</td><td>See [data mappings]</td></tr>
<tr><td>attribute</td><td>[A-Za-z]{2}</td><td>Two-letter process object attribute.</td><td>OV, CX</td></tr>
<tr><td>process object index</td><td>[0-9]*</td><td></td><td>10, e.g. POV[10]</td></tr>
<tr><td>data type</td><td>T|N</td><td>T= text, N=numeric. By default, numeric is used if nothing is specified. For text attributes, T is mandatory.</td><td>TPCX,</td></tr>
<tr><td>object type</td><td>P|H|X|D|C|T|A|E|V|F|B</td><td></td><td>POV, BCX</td></tr>
<tr><td>parent</td><td>APL|SYS</td><td>These are used to indicate that top level attributes are read.</td><td>APL\BNA, SYS\BCX</td></tr>
</table>

8.6.4 Default settings for the Object Browser

The following listing shows the default settings for the Object Browser:

```
[Options]

UsesReversedSingleIndications=0

UsesReversedDoubleIndications=1

LocalRemoteSwitchRxValues = ASTA:ABAY:AGRP
```

```
PcmObjectSubdrawing = 01 - SA_Common\IED_pcm.sd

UsePtSpecificMappingRxValues = ASTA:ABAY:FARC:AGRP

[Filters]

IncludeIx=10

IxMax=99

IncludeAllIndicesForRxValues =
PMEA:WDDD:WDDS:FMFUISTR:FMFUITRP:FTRPISTR:FTRPITRP

IncludedSpecifiedIndicesForRxValue = HTCI#255 : XRST#10000 : QT01#30 :
FARC#12 : HALA#10-17 : Q301#10,11 : XTAG#49 : WSTA#210 :
WNETILIN#220-235 : WNETXLIN#220-235 : WNETXNET#210 :
WAPL#100-350,1000-26250

ExcludedRxValues = ASYS:ESYS:PCOS:MFUI:WMFU:WNETINET

ExcludeLNsStartingWith = BNCC:SAGR_:TOPO_

ExcludedStValues = IN_COLOR

[RX_Settings]

StartIndex=23

Length=4

DetailedLength=8
```

8.6.5 Show/hide process objects shown in the browser

To hide all objects:

1. Empty all excluding filters.
2. Empty all including filters.
3. Set IncludeIx to 100000 (or any other non-existing Ix value).
4. Set IxMax to 0.

To show all objects:

1. Empty all excluding filters.
2. Empty all including filters.
3. Set IncludeIx to 0.
4. Set IxMax to 0.

To include some objects:

1. Modify Include* settings to show some object or remove some of the settings in Exclude* settings.

To exclude some objects:

1. Modify Exclude* settings to hide some object or remove some of the settings in Include* settings.

8.6.6 Create a symbol definition for the object

8.6.6.1 Create a single symbol for the object

Normally, each object presented in the browser has at least one symbol presentation. The configuration file contains default symbols for common process objects. The user has the possibility to create symbol definitions for any object shown in the browser that have unique device identification.

In general, the syntax for symbol definition for one symbol is the following:

device_id = [symbol]

To create new symbol definition for the object, use the following instructions:

1. Open the configuration file and locate symbol definitions. The new line to be added is a key-value pair. Here, WSYS = 06 - SA_Supervision\Server.sd is used as an example.
2. Start a new line and enter an object identification followed by the = character, e.g. WSYS=
   - 2.1. Object identification consists of an object type and possibly an object IO type. The object type is taken from the RX value of the object. The object IO type is taken from the ST value of the object. Examples are shown in the table in section 0.
   - 2.2. The RX value can be more detailed as well, see the DetailedLength setting.
3. After the = character, enter a file name for the symbol, e.g. 06 - SA_Supervision\Server.sd.
   - 3.1. Note that relative paths can be used. For example, directories and symbols in the Palette directory can be found using relative paths. For more information, see Section 25.1.
4. Items 1-3 are mandatory and the symbol definition is now ready. However, the symbol definition misses optional definitions for process mapping, as well as a descriptive display name of the symbol. Process mapping is introduced in more detail in Section 8.6.9, and display name in Section 8.6.7.
5. Save the configuration file and restart Display Builder.

8.6.6.2 Create a symbol group for the object

Object Browser makes possible to instantiate many symbols at the same time for the object shown in the browser. The symbols may include an image of the device, status symbol, and some labels, for example, for identifying the device. These symbols form a symbol group, which can be manually configured to the configuration file. The initial design of the symbol group is done using an empty process display. Note that it is not possible to export the design automatically to the configuration file. The design in the empty process display is only used to help in finding valid coordinates for each symbol of the symbol group. These coordinates can then be used in the symbol definition.

The symbol group can be written directly to the configuration file by using a text editor, but the procedure described below makes it easier to get correct coordinates for the symbols in the group. In general, the syntax for symbol definition for a symbol group is the following:

device_id = <[symbol] & [symbol] & …>

Symbol coordinates inside the symbol group is based on the width and height of the symbols in the symbol group, as well as to the coordinates of the main symbol. Symbol coordinates is an optional definition of the symbol, see Section 8.6.3.

Example:

Designing a symbol group having symbols with different dimensions.

This example illustrates how to configure the symbol group containing three other symbols (Label) on the right side of the main symbol (Server). The labels are distributed vertically even to the height of the Server symbol.

1. Create all symbols that form the symbol group from the Palette and place the main symbol in the middle of the new process display, i.e., the anchor point of the main symbol is in coordinates (0,0). Move other symbols as they would appear surrounding the main symbol. This will help to define the exact coordinates for other symbols surrounding the main symbol.
    1.1. The symbols in this example have the following properties as seen in the Display Builder:

Server (main symbol in the group)	Label L (left justified text)
Width: 2000	Width: 1500
Height: 2000	Height: 400
Anchor point: middle center	Anchor point: middle left

    1.2. The anchor point of the symbol can be seen in Display Builder when selecting the symbol. The anchor point is shown with a dot. Observe the anchor points of all symbols and how they relate to the main symbol width/height.

2. Next, the difference between the X-axis and Y-axis values of the symbols are calculated. Notice that the main symbol is now located in the coordinate (0,0).
    2.1. The difference between the anchor points of the main symbol and other symbols in the X-axis is the main symbol width divided by two, i.e., 1000. Therefore, the X-axis value for other symbols is 1000.
    2.2. The Y-axis values are also calculated taking the anchor points into account:

Label	Y-axis value
Topmost label	Server symbol height divided by two =1000, the anchor point of the label symbol is in the middle, therefore 400/2=200. The vertical distance between the anchor points is 1000-200=800. The Y-axis value for the topmost label is 800.
Middle label	The vertical difference of anchor points of the Server symbol and the middle label is zero (0).
Bottommost label	See topmost label, but negative Y-axis value is used, i.e., -800.

3. The calculated values can be verified by observing the symbol's X-axis and Y-axis values in the symbol properties. Replace the symbol's X-axis and Y-axis values with the calculated values. There should be only minor changes in the symbol positions. Otherwise, there is something wrong with the calculated values.
4. When the coordinate values are calculated and verified, symbol group definition can be added to the configuration file.
    4.1. WSYS=<Server.sd & Label L.sd::1000,800:Server with labels & Label L.sd::1000,0 & Label L.sd::1000,-800>
5. Save the configuration file.
   Note that the process mapping definition, located before the coordinate definition, was skipped, hence the marking "::".

Symbol group is created as separate symbols on the process display. This means that all symbols of the symbol group must be selected separately in order to move them as whole. This functionality, however, makes it possible to move the symbols of the symbol group easily afterwards.

8.6.6.3 Create symbol definition containing alternative symbols for the object

Each object may have alternative symbols, and the user can select a symbol to be created when the object is inserted into process display. In general, the syntax for symbol definition for alternative symbols is the following:

device_id = [symbol] | [symbol]

The | operator is used to determine alternative symbols. To create an alternative symbol definition for the object, use the following instructions:

1. Open the configuration file and locate symbol definitions.
2. Find an existing symbol definition and go to the end of the definition. Insert the | operator to the end of the line and then enter the new symbol definition.
3. Save the configuration file and restart Display Builder.

8.6.6.4 Use same symbol definition for multiple objects

When an object uses the same symbols as another object, it is possible to define a reference to the other object's symbol definition. This reduces duplicate symbol definitions and maintenance of the configuration file. This functionality is helpful when similar device types should have the same symbol definitions.

In general, syntax for symbol definition reference is the following:

device_id = [device id]

A referenced symbol definition is surrounded with brackets, [and]. To create a symbol definition reference for the object, use the following instructions:

1. Open Object Browser configuration file.
2. Find a device identification, whose symbol definition is to be referenced and write it down, e.g. WSYS.
3. Instead of repeating the symbol definition of another device, use device reference syntax:
   - E.g. let device QB02 to use same symbols as device QB01:
   - QB02 = [QB01]
4. Save the configuration file and restart Display Builder.

When the device reference is encountered, the referenced device is searched. The search continues until a symbol definition is found. If there is a circular reference or the reference hierarchy is too deep, an error message is shown.

8.6.7 Change the symbol name in the browser

Once the object has been selected, all the symbols available for that object are shown in a drop-down list. The name shown in the list can be changed to give the user more information about the symbol e.g. shape, color or other properties.

The symbol definition is the following:

symbol = [symbol path] : [data mappings] : [coordinate] : [symbol name]

1. Open the configuration file and locate symbol definitions.
2. The symbol name is the fourth parameter for the symbol.
3. Enter a descriptive name for the symbol e.g. Nuclear Power Plant.
4. Save the configuration file.

If the symbol definition is a symbol group, the symbol name is read from the first symbol of the group. If the symbol name is not defined at all, the symbol path is shown instead.

8.6.8 Change the symbol order in the browser

When there are several symbols defined for the object, it is possible to change the order of the symbols shown in the drop-down list. This functionality enables the user to place frequently used symbols to the beginning of the list. The symbol definition is the following:

device_id = [symbol] | [symbol] | …

The | operator is used to separate different symbols. The more left the symbol is, the higher the order in the drop-down list. Symbol order can be changed by swapping the symbol definitions between two | operators.

To change the symbol order:

1. Open the configuration file and locate symbol definitions.
2. Locate symbol definition for the object whose symbol order needs to be changed.
3. Swap symbol definitions between two | operators, e.g.:
   Original: WSYS = Communication Server.sd | Database Server.sd
   Swapped: WSYS = Database Server.sd | Communication Server.sd
4. Save the configuration file.

The drop-down list is generated by reading the symbol definition and parsing all symbols separated by the | character. Symbols are shown in the order they are specified in the symbol definition.

8.6.9 Create a process mapping definition for the symbol

Symbols have some dynamic functionality based on process data mapped to the symbols. The process mappings are defined in the Object Browser configuration file. There are four types of data mapping:

- No data mapping.
- Map to constant.
- Map to dynamic value.
- Map to calculated dynamic value (status and color).

A symbol may have many data items mapped. The count of data items is relative to the data variables published by the symbol. The data items defined for the symbol must be entered in the same order as they will populate the data variables of the symbol. For example, the first data variable of the circuit breaker symbol is Position and the second is Status Color, the data items are defined so that the first item presents the position and the second the status color. In the configuration file, the delimiting character for data items is the comma (,).

Data mapping types are introduced below. The syntax of data mapping is specified in Section 8.6.3.

8.6.9.1 No data mapping

It is possible to define that nothing is mapped to the data item of the symbol. This is indicated with the - character. Leaving data mapping section of the device empty means that there are no data mappings at all.

8.6.9.2 Map to constant

It is possible to give a constant value for the data item. The constant value can be numeric, alphabets, or alpha numeric. Constant string value is surrounded with quotation marks ("example").

8.6.9.3 Map to dynamic value

The dynamic value has a special syntax, which is used to construct a valid mapping.

8.6.9.4 Map to calculated dynamic value

In addition to generic dynamic values, Object Browser supports status and status colors.

The status is calculated automatically and it swaps the process value if so defined in Display Builder. A reserved word S is used for indicating that the mapping is status. It is also possible use process object index with status, e.g. S[10].

The status color is calculated automatically by the system and it can be used to color some graphical elements of the symbol. A reserved word C is used for indicating that the mapping is status color. It is also possible use process object index with status color syntax, e.g. C[10].

8.6.9.5 Example: Creating process mapping definition for the symbol

In this example, an invented device identification (BREAKER) and symbol called CircuitBreaker.sd is used. It has the following data variables: Status (type: numeric), Status Color (numeric), Identification (text), OperationCount (numeric), Manufacturer (text), ManufactureYear (numeric), where to map static and dynamic process data.

1. Open the configuration file and locate symbol definitions and add a symbol definition for the circuit breaker.
   1.1. BREAKER=CircuitBreaker.sd
2. Decide what data items are mapped to the data variables of the symbol:
   2.1. Data variable Status from S[10]
   2.2. Data variable Status Color from C[10]
   2.3. Data variable Identification from TPCX
   2.4. Data variable OperationCount: skipped/not used
   2.5. Data variable Manufacturer: use constant string "ABB"
   2.6. Data variable ManufactureYear: use constant number 2009
3. Enter a process mapping definition to the symbol definition of the circuit breaker.
   3.1. BREAKER=CircuitBreaker.sd:S[10],C[10],-,TPCX,"ABB", 2009
4. Save the configuration file and restart Display Builder.
5. Verify that process mapping works:
   5.1. Open Object Browser and select an object that has an object type BREAKER and instantiate the symbol to the process display.
   5.2. Select the symbol in the process display and select **Properties**.
   5.3. In the **Properties** dialog open the Subdrawing tab page and verify that data variables are set as defined in item 2.

In case of syntax errors, error messages might appear. The most common error is a mismatch between data types of the process data item and the data variable of the symbol.

8.6.10 Change the object name in the browser

Object names are automatically read from the MicroSCADA process database, and there is no setting to change the format or the name of the object in the configuration file. The object names can be changed by changing the process object attributes in the process database, for example by using Object Navigator.

The object name is generated from the following information found from the process database:

- Object Identifier: The last substring of OI attribute value.
- Object Text: The value of OX attribute.

The format of the object name in the browser is following:

%Object Identifier% (%Object Text%)

The changes are applied when the process database is updated by using the **Select** button. The object name is changed in the browser.

8.7 Network topology coloring

Network topology coloring visualizes the topological state of a network. Line segments and devices can be colored with different colors based on their electric state (powered, unpowered, earthed etc.), their voltage level, or by voltage source. The data can also be used for alarms, external systems etc. A process display with network topology coloring is shown in Figure 111.

Network topology coloring is based on topology models. A network topology model is a model that contains all the objects whose topological state is calculated, and objects that are affected by the calculation. There may be more than one topological model but the models do not affect each other. Each object can be in one model only.

The topological state is a calculated state that should match the real world state of the network. Also, voltage level is calculated for each component.

The relationship between the process database, network topology models, and process displays is visualized in Figure 109. The red dots are normal process objects, for example state indications for breakers. Each model refers to a group of those process objects. Informally, it can be said that those process objects are a part of the topology model. Each process display can refer to those process objects, and objects in these displays will be colored according to the topological state of the object. One display may use process objects from multiple models, and each object can be in several displays. A process object is created for a line segment.

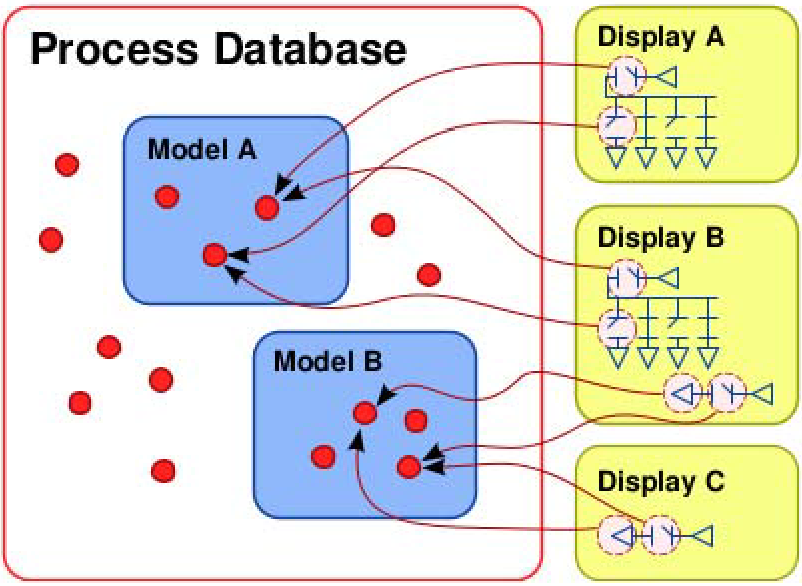


Figure 109: The relationship of the topology model, process objects, and process displays (red dots illustrate process objects)

The data flow of the functionality is shown in Figure 110. The network topology model is built from displays and it is entered into the process database. It is combined with the data from the process. Then, the process database calculates the topological state which is used for coloring the different components in the process display.

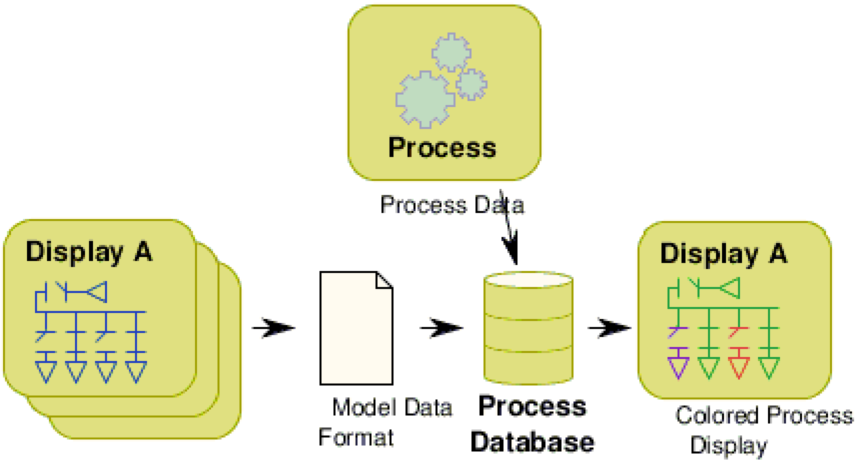


Figure 110: Data flow of the network topology coloring

Topologies may contain several types of network items, such as line segments, different kinds of switching devices, transformers, generators, line indicators and earths.

Line indicators can be used for connecting embedded displays to an overview display, to act as a power feed, or as an output. To connect an overview display to an embedded display, add a line indicator to the embedded display and snap a line segment to it in the overview display.

Transformers can be used to convert a voltage level to another. In network topology coloring, the transformer always emits the configured voltage level. It does not matter what the input voltage is. The earth state will be transferred through transformers.

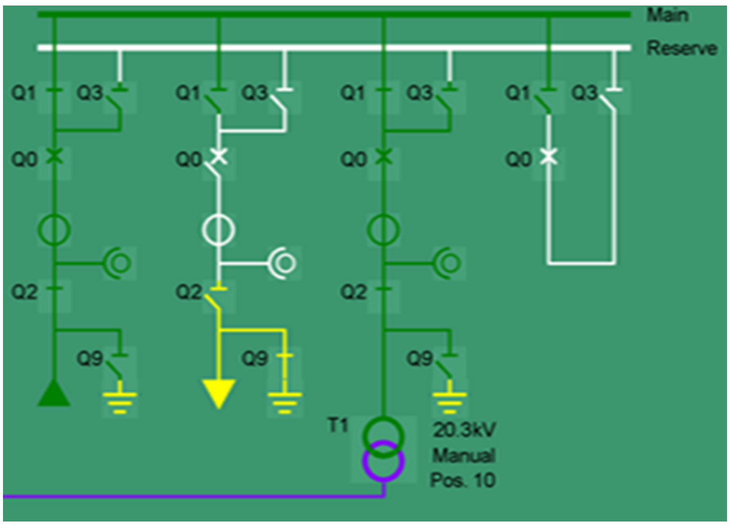


Figure 111: Process Display with network topology coloring

8.7.1 Network topologies

It is possible to design a large display that contains all network objects, but, normally, subnetworks are designed as small displays which are then used to design a large overview display. This section introduces several ways to build a network topology. These include putting all network objects to one large display, or dividing a network to small displays and then combining them to one large overview display. For each option, a description is given, and pros and cons are listed in the aspect of engineering, operation, and maintenance.

It is strongly suggested to divide large network to small process displays first. See Section 8.7.1.4.

Multilayered/hierarchical displays are supported by network topology coloring system. There are no limitations to how many layers a display can have.

A network overview display can be added to the model. All displays that are part of the overview display are then calculated in the process database and colored in Monitor Pro. Displays can also be added separately to the model.

A network can contain one or more network topology models. For most applications, using only one model leads to the best solution. For a large application that may have several hundreds of process displays, and if the network contains non-interconnected parts, it may be useful to divide a network to several different models. Importing a model with a large number of displays may take a while. Having many models does not have a performance impact on a live system.

8.7.1.1 Enabling network topology coloring for a display

To enable network topology coloring for the display:

1. Select **Actions/Manage Topology Models...**.
2. Click **Add New...** to create a new model.
3. Click **Add Display...** to browse for a display.
4. Check for error and warning messages and fix the causes.
5. Press **Save** to import model to the process database.

Monitor Pro can be used to verify that the coloring of the display is correct.

8.7.1.2 One large display

All network objects are put to one large display, there are no other displays. There are two different aspects to this option. This option is suitable for small networks, where it is not necessary to view subnetworks separately. This option is not recommended for large networks.

Configuration
There is only one large display, which is built in Display Builder. This is the only display that is used to construct the network topology model.

Pros

- Engineering: Easy for small networks.
- Operation: One display, and therefore network status is easy to see and status is propagated to whole network.
- Maintenance: Only one display to maintain.

Cons

- Engineering: If the network is large, designing needs excessive zooming and shrinking of graphical symbols. Finding engineering errors in the large network is a time taking process.
- Operation: Awkward navigation and controlling of large network.
- Maintenance: -

8.7.1.3 Many small displays

A large network is divided into subnetworks, and a small display is designed for each subnetwork. A small display normally presents a substation. There is no overview display presenting the large network.

Configuration
Topology model is built of all the displays. If the displays have common items that exist in two or more displays, the state is displayed identically in all displays.

Pros

- Engineering: It easier to design a subnetwork when it is presented in a small display. The engineer can easily see the subnetwork with one glimpse without zooming and panning the display.
- Operation: Zooming and panning of displays is unnecessary. Controlling the network objects is easy because graphical symbols are larger.
- Maintenance: It is easier to maintain a subnetwork, because there is one small display corresponding each subnetwork.

Cons

- Engineering: -
- Operation: There is no overview display.
- Maintenance: -

8.7.1.4 Many small displays and large overview display

A large network is divided into subnetworks. A small display is designed for each subnetwork. A small display normally presents a substation.

Additionally, an overview display is designed to embed small displays into an overview display and to connect them to each other using line indicators and line segments.

Configuration
The network can be displayed as a larger Process Display containing the whole network or as smaller Process Displays containing a part of the larger network. This section describes how the information can be consistent in different Process Displays and how the subnetworks are connected to the network overview.

Start building the network overview by creating the subnetworks and adding them to the network overview.

Each subnetwork refers to the network overview. Changes made in the subnetwork also change the network overview display.

To build the network overview:

1. Build a Process Display for a small geographical area, for example, for a station.

   1.1. Possible topology errors (unconnected symbols) are highlighted when the symbol is attached/snapped to the adjacent symbol and there is a problem with the connection by enabling**View/Highlight Topology**.
2. Connect a line indicator symbol to a line segment that is to be connected to a larger network.
3. Identify the subnetwork with a unique name. This is done automatically, but can be changed manually by setting a custom attribute for the process display.
   3.1. Create attribute **SADisplayName** by selecting **View/Properties/Custom Attributes**.
   3.2. Give the Process Display a unique name.
4. Save the display.
5. Repeat steps 1 to 4 until the large network is divided into small enough Process Displays.
6. Create a new Process Display used as a network overview.
7. Embed the small Process Displays in the large Process Display as objects or symbols.
   7.1. Select **Objects toolbar/Subdrawing**.
   7.2. Open a V file to embed a Process Display.
8. Repeat step 7 for each Process Display that is required to be embedded in the network overview.
9. Connect the small Process Displays to the network overview by normal busbars.
   9.1. Add a busbar symbol.
   9.2. Use the Snap-to-Symbol or Snap-and-Stretch-to-Symbol function to connect the symbol to the line indicator of the small Process Displays.
10. Identify the network overview with a unique name. This is done automatically, but can be changed manually by creating a new attribute SADisplayName by selecting **View / Properties /Custom Attributes** and setting the value to, for example, Overview.
11. Save the V file.
12. Add the network overview display to the model and run it.
   12.1. Select **Actions/Manage Topology Models...**.
   12.2. Click **Add New...** to create a new model.
   12.3. Click **Add Display...** to browse for a display.
   12.4. Check for error and warning messages and fix the causes.
   12.5. Press **Save** to construct the model to the process database.

Monitor Pro can be used to verify that the coloring of the display is correct.

To verify the successful configuration:

1. Login to Monitor Pro.
2. Make sure that Network Topology Coloring is enabled from the Settings dialog.
3. Open the network overview display and verify that coloring is working.
4. Open one of the small embedded process displays and verify that the coloring is working.

Embed the small Process Displays as a subdrawing, as described above. Unique identifications for network objects are lost if a display is embedded in the network overview display as separate symbols using **File/Merge** instead of using **Object Toolbar/Subdrawing**.

Pros
See also Section 8.7.1.3.2 in Section 8.7.1.3.

- Engineering: Designing the overview display is easy, just embed small displays and connect them to each other with busbar symbols. Small displays are references in the

overview display, thus, it is easier to handle small display in the overview display when only one symbol is selected.
- Operation: There are small displays and an overview display that can be used for viewing network status. The coloring in small displays is always in accordance with the status calculated from the overview display.
- Maintenance: -

Cons

See also Section 8.7.1.3.3 in Section 8.7.1.3.

- Engineering:-
- Operation: -
- Maintenance: -

The SADisplayName value is added automatically but the custom key itself must exist in the embed small display file.

Simple small displays
Description

Simple small display contains all line segments, switching devices, transformers and measurement transformers. A simple small display is normally a display presenting a bay of a substation. This display is designed for purposes where the display should not have any additional detailed information such as measurements. The display may also contain a link where it is possible to open a detailed small display.

Configuration

1. Create line segments, switching devices, transformers and measurement transformers.
2. Add line indicators where upper network is about to be connected.
3. Save the display.
4. See Detailed small displays to get more information on how to use simple small displays.

Detailed small displays
Description

In addition to network objects shown in the simple small display, detailed small displays normally contain measurement data; values and bars. Detailed small display is built using a simple small display and measurement value/graph symbols.

Configuration

1. First build simple small display, see section Simple small displays.
2. Create a new process display for detailed small display.
3. Embed simple small display.
4. Create measurement values and bars beside the simple small display.
5. Save detailed small display.

Detailed small displays are not included in the network overview display since detailed information in this display is not part of the topology. Instead, simple small displays are included in the network overview display.

8.7.1.5 Summary overview display

There might be a need for a small overview display, which is used to get a quick view of the large overview display. Usually, this display contains less than fifty network objects, which are referring to the large overview display. Normally, the network objects selected to this display are important nodes in the network. When the important node has an unusual state, it usually means this affects the whole network.

Configuration

Summary overview display requires that a valid overview display is configured to the network topology coloring system. To create a summary overview and to reference network objects in the overview displays, the following steps should be followed:

1. Create a new process display, which will be a summary overview.
2. Open a network overview display.
3. Copy network objects from the network overview display.
    3.1. If network objects are inside the embedded display, open the display and copy the network objects.
    3.2. An embedded display can also be copied.
4. Paste the network objects to the summary overview display.
   Use the Paste Special feature and select the Preserve option. This option will also copy the data variables used for the topology state mappings.
5. Repeat steps 3 and 4 to each network object to be referenced.
6. Save the summary overview display and open it in Monitor Pro.
7. Verify that the network objects in the summary overview are colored with the same color as in the network overview.

Note that referred network objects cannot be used to propagate their state to adjacent network objects. Also, if there is a network object that is not part of the network overview display, the line segment is not colored.

8.7.2 Getting network status outside displays

A network overview display is normally a subnetwork of a larger network. For example, the network overview display is a distribution network, which is a part of a transmission network. Therefore, status information is missing from the outer network, that is, normally a transmission network.

Get the status information from outer networks in the following ways:

- Use external color handling for voltage sources provided by DMS600.
- Use an additional circuit breaker and earth switch.
- Use measurement values.

8.7.2.1 Using external color handling

This scenario is suitable in systems where:

- DMS600 is installed and line indicators are configured to be colored by DMS600 (standard function).
- The network is larger than the used network in SYS600.

To color the line indicators and line segments with the status information received from DMS 600:

1. Select **Settings/Network Topology Coloring** in Monitor Pro.
2. Select **DMS 600 external color handling**.
3. Click **OK**.

Use the Color Setting Tool to synchronize colors with DMS 600.

The line indicator foreground color is determined by using the value of the **Position** data variable. The valid values are shown in Table 136.

External color values 20-29 are mapped to color indexes 90-99. These indexes overlap with Voltage Level color definitions.

Table 136: Valid foreground color values

Color	Values
Unpowered	0
Powered	1-5 and 20-29
Uncertain	11
Looped	12
Earthed	13

8.7.2.2 Additional circuit breaker and earthing switch

This scenario is mostly suitable in electric networks. An additional circuit breaker and an earthing switch can be used after an incoming line indicator. The devices can be virtual or real. In this scenario, virtual devices are created. These two switches simulate the state of the upper level network enabling to simulate all the states that a feeding line may have, except for looped the state.

1. Open **Tool Manager/Object Navigator**.
2. Select **Object/Install Standard Function**.
3. Browse for Standard Functions/SA_LIB/Process/Switching Device and click **Install**.
4. Fill in STATION_NAME, BAY_NAME, DEVICE_NAME and P_OBJECT_NAME.
5. Click **Apply**.
6. Open Process Object Tool from the **Tools** tab.
7. Click **Create All**.
8. Repeat steps 1. to 7. to create more virtual circuit breakers.
9. Use Object Browser in Display Builder to create symbols to the station Process Display.

It is recommended to separate virtual circuit breakers from the station picture.

An example of an earthed upper level network (Rivendell 110 kV) is shown in Figure 112.

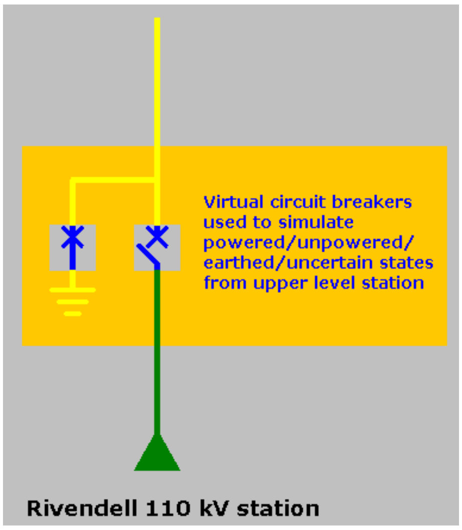


Figure 112: Example of an earthed upper level network

8.7.2.3 Using measurement values

Measurements can be used to control the direction and color of the line indicator (incoming/outgoing, or powered/not powered, respectively) from an outer network.

1. Install and configure a line indicator with the standard function installation and configuration tool from the Object Navigator.
2. Create a new Process Object of type Analog Input(AI) or use an existing, e.g. the power value from the incoming feeder connected to the outer network.
3. Create an Event Channel and a Command Procedure. Connect the Command Procedure to the Event Channel and the Event Channel to the measurement Process Object. Set the Event Channel activation attribute AA to New Value.
4. The Command Procedure is used to calculate the line indicator direction and color based on the measurement Process Object value.
5. Install the line indicator symbol with the Display Builder/Object Browser.

Because the value of the line indicator is propagated to the network, the Event Channel/ Command Procedure should be designed so that it only changes the value when necessary.

8.7.3 Setting up the configuration

See Section 8.7.1.1.

8.7.4 Removing the configuration

Network topology coloring of a display can be removed by removing the display from the model. This will inform network topology coloring system to not to calculate the state of the network for that display.

To remove the configuration:

1. Open Display Builder.
2. Select **Actions/Manage Topology Models...**.
3. Select a model in which the display is.
4. Select a display from the list and click **Remove display**.

Verify the removal by opening the removed display in Monitor Pro.

8.7.5 Network topology coloring modes

Network topology coloring is enabled if it uses dynamic coloring. Network topology coloring is disabled if it uses static coloring.

1. Disabled network topology coloring
   If the network topology coloring is disabled, the switching device's position does not affect the network topology coloring. Therefore, a static foreground color is used for objects in a Process Display.
2. One color for all powered line segments
   Define color called Powered that is used for all powered line segments.
3. Coloring based on level
   In a level based network topology coloring, a feeding point has a level, for example a voltage level. After the feeding point, a level color applies to the network's powered line segments.
4. Feeding point type and its color determine the coloring of powered line segments
   There are predefined colors for feeding points in SYS600. In SYS600, feeding points are voltage sources, such as generators, transformers, and incoming line indicators. A priority order of the feeding point type is applied for this mode.

8.7.5.1 Used states and colors

States used in network topology coloring are defined in Table 137. Configure the state's color by using the Color Setting tool, see SYS600 Process Display Design. The network object can have one of the following states:

- Powered
- Unpowered
- Looped
- Uncertain
- Error
- Earthed

Table 137: States and colors

Color name	Description	Color
Earthed	The object is connected to ground.	Yellow
Error	The color is used in error situations. For example, powered line segment is earthed.	Magenta
Looped (meshed)	The color is used when there is a loop, e.g. two or more feeding points.	Red
Powered/active, common	Default color used on powered objects.	Green
Powered, generator	Color used for line after generator.	Green
Powered, incoming line indicator	Color used for line segment after line indicator.	Green
Powered, transformer (primary)	Color used for line segments when power flows to transformer primary winding side.	Cyan
Powered, transformer (secondary)	Color used for line segment after transformer secondary winding.	Dark Green
Powered, transformer (tertiary)	Color used for line segment after transformer tertiary winding.	Olive Green
Powered, transformer (quaternary)	Color used for line segment after transformer quaternary winding.	Dark Cyan
Powered, custom 1-5 and 20-29	Reserved for external color handling.	Several colors
		Green
Uncertain	Color used for line segment when the state of a switching device is uncertain.	Magenta
Unpowered/inactive	Line segment is cold, no power flow.	White
Level colors	Colors that are used for level based coloring. The default voltage levels in kilovolts (kV) are 0, 3, 6, 10, 15, 20, 30, 45, 66, 110, 132, 150, 220, 400, 800 and 1600.	Several colors Green
Line segments (Network topology coloring disabled)	One color used for all line segments when network topology coloring is disabled.	Black

8.7.5.2 Color priorities

In the network, situations can arise where a line segment can have two or more colors that could be used in the line segment, for example, generator and transformer colors. A default priority order of colors are shown in Table 138. Some of the priorities are not configurable and the user can only configure priorities of feeding point types.

Table 138: Color priority order

State	Priority	Configurable	Description
Error	1	NO	
Uncertain	2	NO	The used color after uncertain switching device.
Looped	3	NO	
Earthed	4	NO	Earthed color is always indicated before powered and unpowered.
Powered, feeding line	5	YES	The priority of a feeding point type color in powered line segments is configurable. Functionality can be seen in the line segments, which are looped. The coloring mode must be set to feeding point or voltage source type to enable priorities.
Powered, generator	5		
Powered, transformer, windings (primary, secondary, tertiary, quaternary)	5		
Unpowered	6	NO	Unpowered objects have lowest priority

8.7.6 Configuring voltage levels

The Network Topology Manager tool is used for configuring the voltage levels of the power process (network topology schema POWER) in the current application. It shows network topology models of the power process in the current application.

To configure the voltage levels:

1. Open the NT Manager tool by selecting **Tools/Engineering Tools/Tool Manager/ Application Objects/NT Manager**.
2. View the default voltage levels by clicking **Levels/Restore Defaults**.

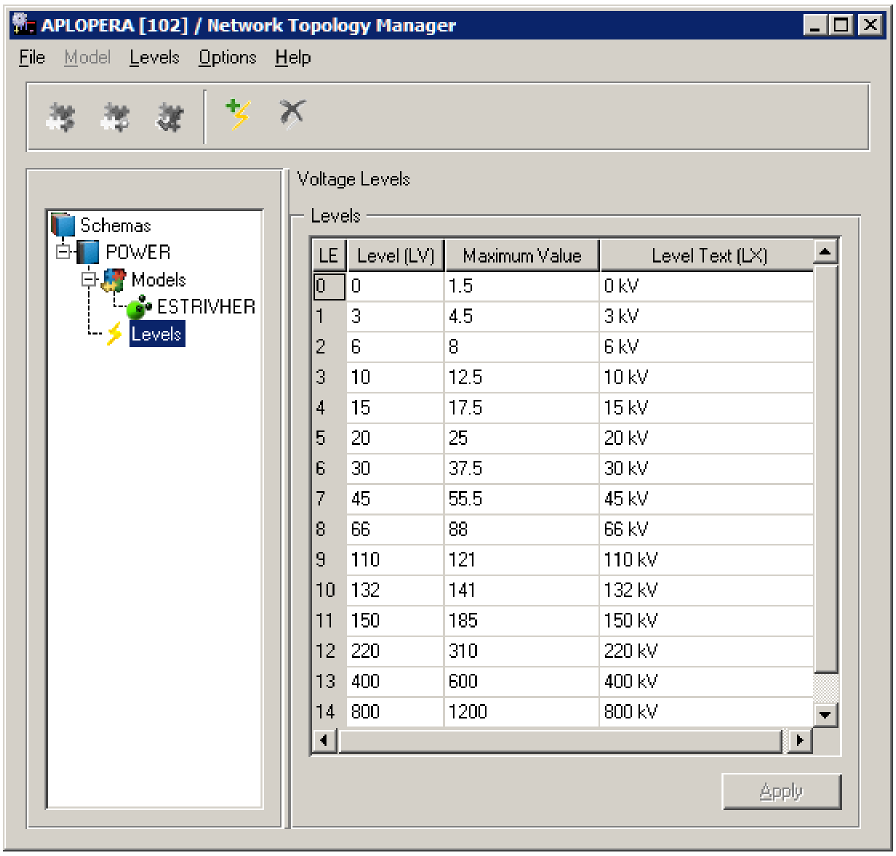


Figure 113: Network Topology Manager dialog

Power process may have up to 20 voltage levels. Levels are numbered starting from zero. Each level consists of three items:

- **Level (LV)**, the value of the nominal voltage level.
- **Maximum Value**, the item defines the maximum level value that is considered to belong to the level.
- **Level Text (LX)**, the level text that describes the nominal value. The tool automatically suggests that a level text is a voltage level value as kilo volts, for example, 800 kV.

8.7.7 Using Network Topology Coloring Process Objects in application design

In addition to visualization of topological state of line segments in single line diagrams, Network Topology Coloring Process Objects can also be used for the following purposes:

- generating events and alarms
- calculating interlocking conditions

The following is an example of interlocking program that can be used in switch device control dialogs U_INTERLOCKING method (see Section 6.2).

Program prevents the closing of switch device if Powered state meets Earthed or Uncertain state.

```
; By this program it is possible to create internal interlockings by
utilizing SCIL

;
```

```
; as OUTPUT you can give variables:
; @OPEN_INTERLOCKED, boolean, (if true then opening gets disabled)
; @CLOSE_INTERLOCKED, boolean, (if true then closing gets disabled)
; @INTERLOCKING_CAUSE, text, cause to be shown on the dialog infobar
;
; None of these is obligatory!
@t_LN = %PF_ID
@i_IX = %PF_INDICATION_INDEX
@b_Earthed = False
@b_Prevent_Closing = True
@t_Additional_Text = ""
;prevent closing if earthed, opening of breaker should be always possible
;if switch is open
#if 't_LN':PTS'i_IX' == 2 #then #block
;network object connection
@l_NO = 't_LN':PNO'i_IX'
;Connected network object(s)
@l_CN = 'l_NO:vLN':PCN'l_NO:vIX'
#loop_with i = 1 .. length(%l_CN)
@tmp_LN = %l_CN(%i).LN
@tmp_IX = %l_CN(%i).IX
;if line_segment
#if 'tmp_LN':PNS'tmp_IX' == 3 #then #block
;if earthed
#case 'tmp_LN':PNT'tmp_IX'
#when 0 #block
@b_Prevent_Closing = True
@t_Additional_Text = "Unknown"
#block_end
#when 1 #block
@b_Prevent_Closing = False
@t_Additional_Text = ""
#block_end
#when 2 #block
```

```
@b_Prevent_Closing = True

@t_Additional_Text = "Uncertain"

#block_end

#when 3 #block

@b_Prevent_Closing = True

@t_Additional_Text = "Powered"

#block_end

#when 4 #block

@b_Prevent_Closing = True

@t_Additional_Text = "Error"

#block_end

#when 5 #block

#if %b_Earthed #then @b_Earthed = False

#else @b_Earthed = True

#block_end

#case_end

#block_end

#loop_end

#if %b_Earthed and %b_Prevent_Closing #then #block

@CLOSE_INTERLOCKED = TRUE

@INTERLOCKING_CAUSE = "#Earth meets " + %t_Additional_Text + "#"

#block_end

#block_end

#RETURN LIST(OPEN_INTERLOCKED=%OPEN_INTERLOCKED, CLOSE_INTERLOCKED=
%CLOSE_INTERLOCKED, INTERLOCKING_CAUSE=%INTERLOCKING_CAUSE)
```

8.7.8 Troubleshooting Network Topology Coloring

Table 139: Network Coloring Topology troubleshooting

Problem	Reason	Solution
Only some items are colored.	The topology is built from an older version of the display. Some items may be added to the display afterwards.	Rebuild the topology.
Only subdrawings are colored.	Only subdrawings are added to the topology. If two or more displays contain the same object, all connection to this object will be added to the model. For example, if there are components A and B in two displays and in the first display they are connected by a line segment and in the second display not, the current flows from component A to B even when viewing the second display.	The main display should also be added to the topology. To add the main display, the user must first remove subdrawings that are in the model, and then add the main display.
Odd loops are detected or objects are colored oddly.	Two or more displays may contain conflicting topologies. If two or more displays in the same network topology model contain same objects, the model will include all logical connections between these objects. This may not represent the physical reality.	Use the same objects only once in the displays which constitute a network topology model.
Error: "Switch must be connected to two non-ambiguous items"	The same object is used more than once in one model but with different connections, e.g. the same object is used within different embedded bay displays.	Remove the non-ambiguous object and rebuild the topology.
Error: "Display already exists in this model with the same SADisplayName"	• Different display files using the same SADisplayName custom key value • Same display used more than once in the model, e.g. embedded bay displays	• SADisplayName custom key value must be unique within one model. • This error message will disappear after the topology model is saved.

8.8 Launching external application from Process Displays

There is a possibility to open external programs, like relay tools from the Process Displays of the SYS600 Monitor Pro. See Process Display Design manual for more information about Tool Launcher.

8.9 Modifying data variable with SCIL

Modify the data variable values by running SCIL when the process object value changes, and place the result to the data variable. It is also possible to run SCIL when data variable changes, for example when the user clicks a Process Display object. Place the result to the process object. A SCIL code is located in \sc\apl\[application name]\PICT\.

LIST is given as a predefined variable, whose name is %L_INPUT, to the code:

LIST(ITEM=<Item ID of OPC item>

The same SCIL file is run in both cases, when

- The process object value changes. The DIRECTION list attribute has value READ.
- The data variable value changes. The DIRECTION list attribute has value WRITE.

When writing, the VALUE list attribute is written to the process object. When reading, the VALUE list attribute is written to the Process Display data variable.

The OPC item name and data type are also passed to the SCIL code, and they are available in the ITEM and DATA_TYPE attributes, respectively.

To modify the data variable value when process object changes:

1. Create a Process Display.
2. Create a data variable to the data source.
3. Define process object or process index where the variable is connected to.
4. Set data variable's type to Generic signal.
5. Select **Create and edit new file** from the Run SCIL drop-down menu.
6. Notepad opens the created SCIL file for editing.
7. Write the SCIL code to modify the variable value and save the file.

   The file suffix must be scil.

8. Run the Process Display in Monitor Pro.

To modify the process object value when the data variable changes:

1. Create a Process Display.
2. Create a data variable to the data source.
3. Define process object where the variable is connected to.
4. Set the data variable’s type to Generic signal.
5. Select **Create and edit new file** from the Run SCIL drop-down menu.
6. Notepad opens the created SCIL file for editing.
7. Write the SCIL code to modify the variable value and save the file.

   The file suffix must be scil.

8. Define input object that modifies the data variable value.
9. Run the Process Display in Monitor Pro.

Section 9 Event Display

This section describes the technical aspects of the Event Display. For information on the usage of the display, see the SYS600 Operation Manual.

Event Display provides a list of events. The purpose of the list is to provide the operator a view of what is happening in the system. It is also possible to receive information, such as activities carried out by other users, acknowledging of alarms, editing of limit values, and logins.

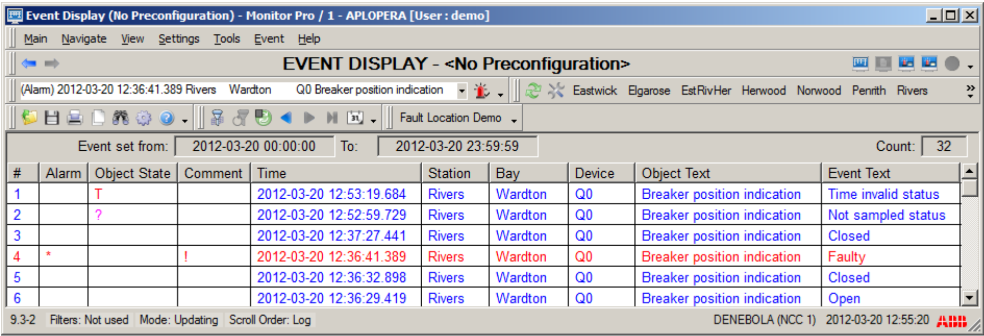


Figure 114: A typical Event Display

List of displayed columns can be configured, and the list can be filtered using different criteria. The configuration of the list can be readymade for operators. Events can be commented. When an event is double clicked or the **Comment...** option is selected from a context menu, a comment dialog appears. Each event can contain one comment.

Sections Section 9.1, Section 9.3, Section 9.4, Section 9.5, and Section 9.6 are applicable to Event Displays, Alarm Displays and Blocking Displays.

9.1 Attributes

Almost all attributes can be subscribed to the list. However, there are a few exceptions. Attributes that cannot be subscribed to the list are described in Table 140.

Table 140: Attributes which cannot be subscribed to the list

Attribute	Description
CN	Connected Network Objects
ND	Network Topology Data
NF	Network Feeds
NO	Network Object Connection
RO	Referenced Objects

The content of the attributes are described in the SYS600 Application Objects manual. There are, however, some additional attributes. Attributes that are common to all lists are described in Table 141.

Table 141: Common list specific attributes

Attribute	Description
OI1-OI5	OI1 The 1st level of Object Identifier (OI) attribute
OI2	The 2nd level of Object Identifier (OI) attribute
OI3	The 3rd level of Object Identifier (OI) attribute
OI4	The 4th level of Object Identifier (OI) attribute
OI5	The 5th level of Object Identifier (OI) attribute
SX	Translated text (TX) is used instead of SX
Time attributes	For all time attributes, the millisecond part of the time is automatically appended.

9.2 Event Display specific attributes

All attributes in the history database are available in Event Displays. In addition, attributes listed in Table 142 can be selected to the list.

Table 142: Attributes specific to the Event Display

Attribute	Description
STATUS	Set of three columns: 1. Alarming state. If the object was in alarming state when the event is generated, an asterisk is displayed. 2. Object status. If the signal was in non-normal state, a state indication is displayed. 3. Comment character. If the event contains a comment, and exclamation mark is show. For more information on how to name status field list captions, see the SYS600 Operation Manual. Note that the status describes the status of the process object, not the status of the event. For example, faulty time means that the process object had a faulty time.
UTC_TIME	The time used for sorting the list. If scroll order setting is LOG, then HT is used. If scroll order is EVENT, then ET is used.
EVENT	Event text. Alias for MX attribute.
CU	Custom attribute.

9.3 General Settings

Each list display has General settings, which can be used to modify the appearance of the list. General settings can be accessed by selecting **Settings/Display Settings/General Settings**. The engineer can define the general application level settings by selecting **Visibility/Application**. All users can use these settings, and they are the default settings for users in the View authorization group. For more information on the available options and list specific display settings, see the SYS600 Operation Manual.

9.4 Layout Settings

The layout settings can be configured by selecting **Settings/Display Settings/Layout Settings**. The engineer can define the display layout for application level settings by selecting

Visibility/Application. All users can use these settings, and they are the default settings for users in the View authorization group. For more information about customizing the column layout, see the SYS600 Operation Manual.

9.5 Locate Object

When a signal is displayed on a list, the user can open a process display containing the signal. This is done by Locate Object in Monitor Pro function in the context menu. The location of each signal must be configured. The configuration is either done by using the PV (Process Views) or a SCIL program (GET_DISPLAY_NAME).

A display is opened and the selected object is presented with a highlight symbol. See Figure 115.

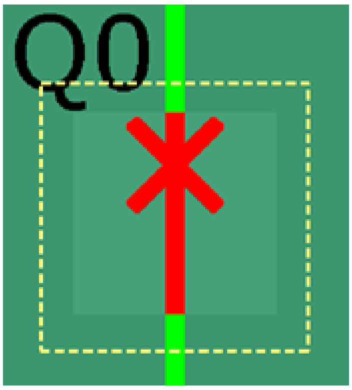


Figure 115: Located object is highlighted in a display

If DMS600 software is installed, it is also possible to locate objects in DMS 600. In this case, a **Locate object in DMS** function is available in the context menu. Locate functions can be disabled from the Settings dialog.

9.5.1 Configuration using PV (Process Views) attribute

The configuration is done by using the PV (Process Views) attribute of the Process Object.

- In Object Navigator, right-click the Process Object and select **Group Properties**.
- Select the Process Views (PV) tab and click **New**.
- Select the Process Display(s) to be added to Process Views.

It is possible to add several Process Views. In such cases, an Option dialog is shown in Monitor Pro for selecting the appropriate Process Display.

9.5.2 Configuration using SCIL program

The configuration is done using a SCIL program GET_DISPLAY_NAME.TXT. The program is located in \SC\APL\<application name>\PICT-folder. The program gets the identification of the signal as parameter and returns the display name without extension. Parameters can be seen in Table 143. An example program can be seen in "An example of a locate object procedure". The example program returns the station name element of the object identifier. This configuration works if the system is built so that there is a separate process display for each station, and the name of the process display matches the name of the station.

Table 143: Parameters of the locate object procedure

Number	Parameter Name	Description
1	LN	The Logical Name (LN) of the selected signal
2	IX	The Index (IX) of the selected signal
3	OI	The Object Identifier (OI) of the selected signal

An example of a locate object procedure

```
@LN=argument(1)
@IX=argument(2)
@IL='LN':PIL'IX'
#return %IL.STA
```

If the `GET_DISPLAY_NAME.TXT` procedure file does not exist or its execution fails, the function reads the PI attribute value of the object that produces the alarm or event. The PI attribute value contains a display name without the file extension.

9.6 Filters and Preconfigurations

The lists can be filtered. The user can create a filter, or filters can be created beforehand. A filter can be created using the **Filter Settings** dialog. Condition can be based on the OI of the signal, on the function of the device, or on alarm class. An arbitrary SCIL condition can be entered as custom condition.

Filters can be created using an OI based filtering condition, signal function, alarm class and custom SCIL condition. Complex conditions can be added as custom SCIL conditions. For example, condition (AC == 2 AND OI==”Eastwick*”) OR ((AC == 1 or AC == 2) and OI==”Rivers*”) fetches all events from Eastwick station with Alarm Class (AC) 2, and all events from Rivers station with Alarm Class (AC) 1 or 2.

It is possible to save filter conditions as a preconfiguration. Start creating a preconfiguration by creating a filter. Then, select **Filter Settings/Preconfigurations...**. Enter a name for preconfiguration and click the **Save** button.

Preconfigurations can be either user specific or application specific. Application specific preconfigurations can be used by all users, and user specific preconfigurations can be used only by the user who has created it. The engineer can create readymade lists for operators using preconfigurations. It is possible to configure the used credentials so that operators do not have permission to enter a filter, but only select one from the readymade preconfigurations.

9.7 Color Settings

Color settings can be accessed from **Settings/Display Settings.../Color Settings...**. Signals can be colored based on different conditions. The syntax for conditions is the same as in filter configuration. All signal attributes can be used for defining the coloring condition. Also, several general color settings, such as background of the grid, are available. If a signal matches multiple coloring conditions, the first matching color rule is used.

For each rule, there are options to specify foreground color, optional background color and blinking. The blinking affects the background color of the row.

Other color settings are explained in Table 144. For some color settings, the Use Color Scheme option is available. If checked, the color defined in the color scheme is used.

Table 144: Color settings

Number	Parameter Name
Background color	Default background color for the grid.
Gridline color	Color of the gridline.
Filter in use color	Color in the Filter in Use sign on the bottom of the screen.
Frozen mode color	Color in the Mode sign on the bottom of the screen when frozen mode is used.
Day Break color	If rows are sorted using a time column, the day break color is used as default background for every other day.

9.8 Blockings

The list provides a quick access for controlling signal blockings. For signals, there is a context menu item for opening the **Signal blocking state** dialog. With the dialog, the user can control the blockings of the selected signal.

The blockings dialog can be disabled from the lists' General settings.

9.9 Authorization Groups

Lists use different authorization groups. Authorization groups are used for verifying the user's privilege to perform actions. If the authorization level for a specific authorization group is not defined for the user, the authorization level of authorization group GENERAL is used. Table 145 describes the authorization groups and their usage. For more information about authorization, see Section 24.

Table 145: Authorization groups related to event display

Authorization Group	Level	Enabled Action
BLOCKING_HANDLING	>= 1, Control	Change blocking states
EVENT_HANDLING	>= 1, Control	Modify event comments
PRO_EVENTS_CUSTOMIZE	>= 2, Engineering	Change Application level settings Modify preconfigurations

Section 10 Alarm Display

This section describes the technical aspects of the Alarm Display. For more information on the usage of the display, see the SYS600 Operation Manual.

SYS600 Alarm Display displays a summary of the present alarm situation of the supervised process. There are two Alarm Display templates, and both templates share the same settings and preconfigurations. Template 1 contains separate lists for persisting alarms and fleeting alarms. Persisting alarms are alarms that are in active and unacknowledged state. Fleeting alarms are alarms that are in inactive and unacknowledged state. Template 2 contains one list for all alarms. An example of Alarm Display template 1 is shown in Figure 116.

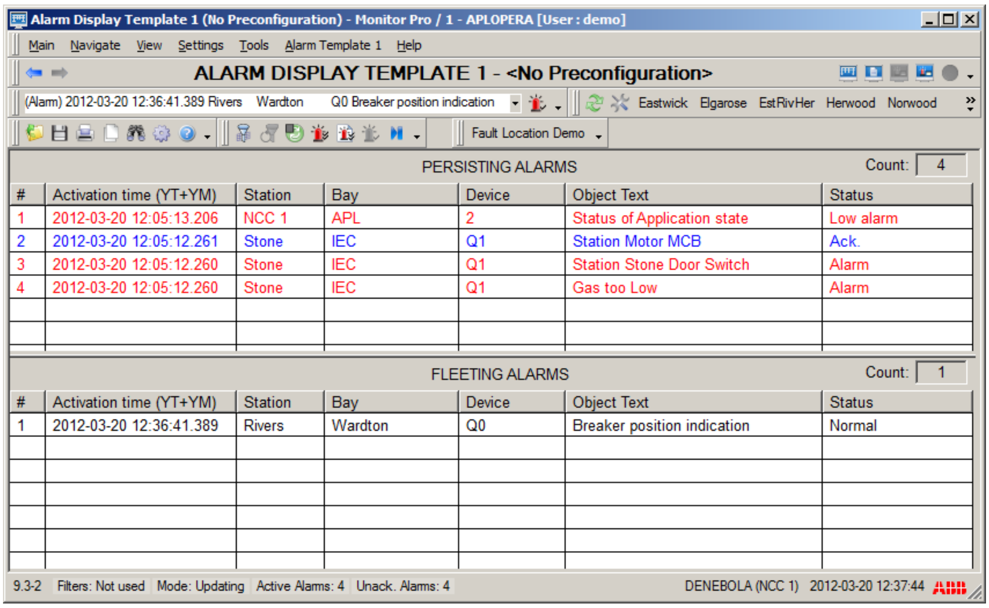


Figure 116: A Typical Alarm Display

One signal can have only one alarm. If a new alarm is created for a signal with an existing alarm, the new alarm overrides the earlier alarm.

Lists allow the user with adequate authorization level to acknowledge alarms. The user can acknowledge either all alarms, a single alarm, all visible alarms or selected alarms. Most of the process object attributes can be added to the list. There is also an alarm list specific attribute STATUS, which contains a textual explanation of the alarming state.

Both alarm list templates support coloring, filtering, preconfigurations and signal blocking. These features are explained in sections from Section 9.6 to Section 9.8.

Authorization groups used in Alarm Displays are described in Table 146. If authorization level for a specific authorization group is not defined for the user, the authorization level of authorization group GENERAL is used. For more information about authorization, see Section 24.

Table 146: Authorization groups related to Alarm Display

Authorization Group	Level	Enabled Action
ALARM_HANDLING	>= 1, Control	Acknowledge alarm
PRO_ALARMS_CUSTOMIZE	>= 2, Engineering	Change Application level settings in alarm list Modify preconfigurations
BLOCKING_HANDLING	>= 1, Control	Change blocking states

Section 11 Blocking Display

This section describes the technical aspects of the Blocking Display. For more information on the usage of the display, see the SYS600 Operation Manual.

SYS600 Blocking Display displays a summary of the present signal blocking situation of the supervised process. The Blocking Display is used for viewing and controlling signal blockings. The list can contain either signals with blockings or all signals. Signals can be filtered. Blocking columns are editable. If the user has adequate authorization, the user can either block or deblock signals. An example of a Blocking Display is shown in Figure 117.

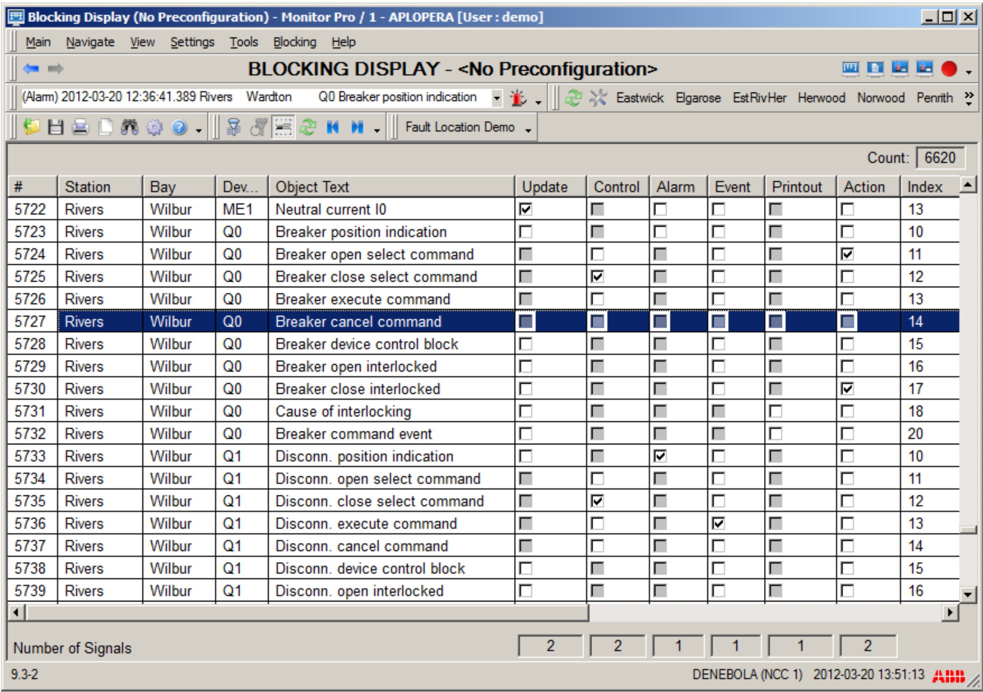


Figure 117: A typical Blocking Display

With default settings, an event is generated when a blocking attribute is changed. This can be changed with a configuration option.

The Blocking Display supports coloring, filtering, preconfigurations and signal blocking. These features are explained in sections from Section 9.6 to Section 9.8.

Authorization groups used in the Blocking Display are described in Table 147. If authorization level for a specific authorization group is not defined for the user, the authorization level of authorization group GENERAL is used. For more information about authorization, see Section 24.

Table 147: Authorization groups related to Blocking Display

Authorization Group	Level	Enabled Action
BLOCKING_HANDLING	>= 1, Control	Change blocking states
PRO_BLOCKINGS_CUSTOMIZE	>= 2, Engineering	Change Application level settings Modify preconfigurations

Section 12 User Activity Log Display

This section describes the technical aspects of the User Activity Log Display. For more information on the usage of the display, see the SYS600 Operation Manual.

SYS600 User Activity Log Display displays important user activated events for later analysis. The User Activity Log Display is a read only list.

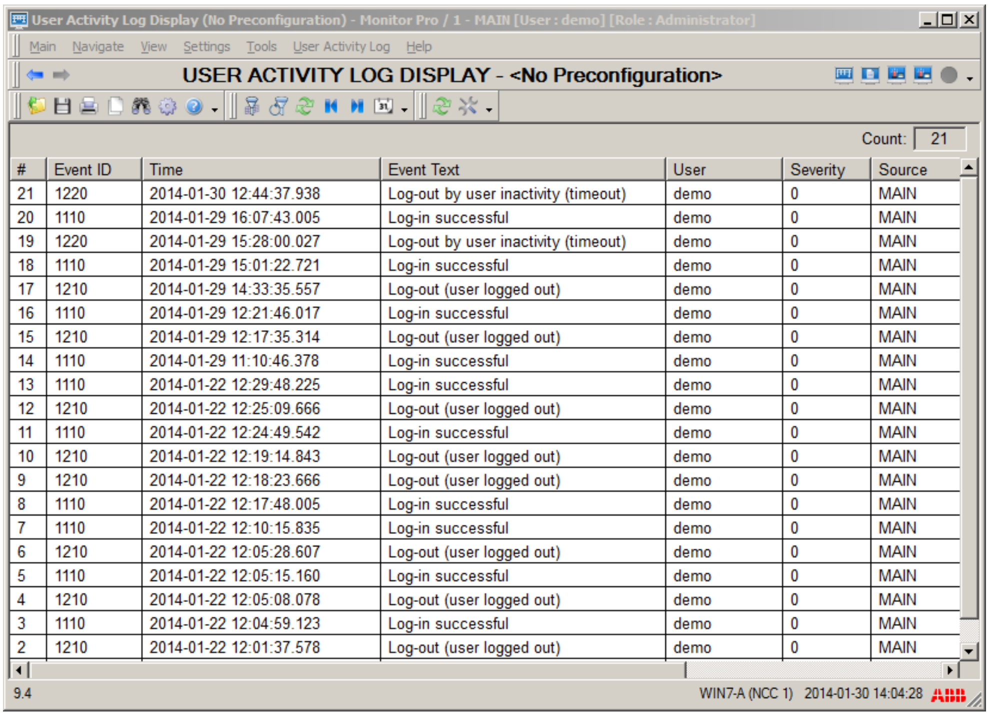


Figure 118: A typical User Account Log Display

The User Account Log Display supports coloring, filtering and preconfigurations. These features are explained in sections from Section 9.6 to Section 9.8.

Authorization groups used in the User Account Log Display are described in table below. For more information about authorization, see Section 24.

Table 148: Authorization groups related to User Account Log Display

Authorization Group	Level	Enabled Action
GENERAL	5, System Management	View User Account Log Display

Section 13 Trend Basket

With the Trend Basket the user can select data from the process database to be logged. When trend logging is activated, process object values are stored to Data Objects in a cyclical manner. You can view the logged data in the Trend Display.

To open the Trend Basket dialog login to application with **SYS600 Tool Launcher** and select **Application Configuration > Trend Basket**.

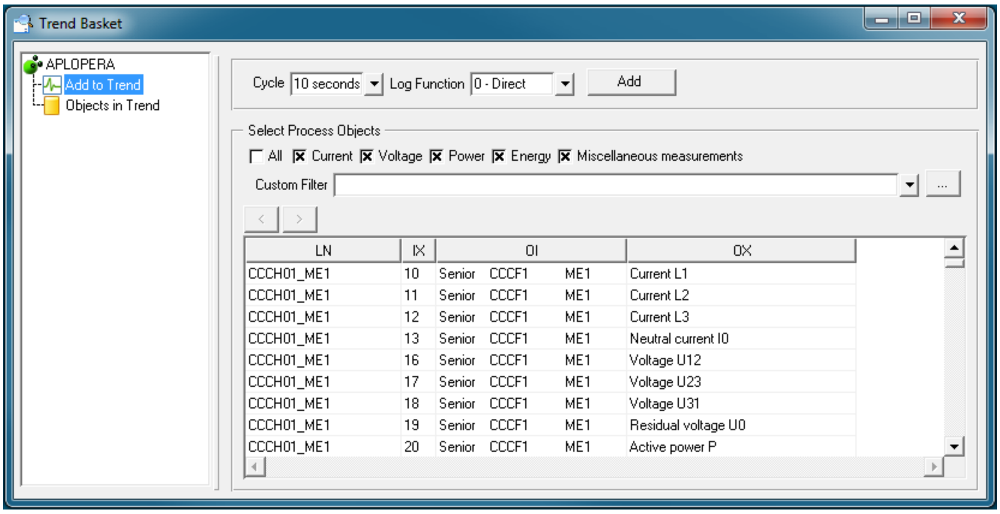


Figure 119: The Trend Basket

When the Trend Basket is opened for the first time, needed Data Objects and Time Channels are automatically created. The number of Data Objects, the maximum length of logs and the default logging cycle are defined in file \sc\sa_lib\base\bbone\use\bgu_ta_setup.txt. The name of trend related Data Objects and Time Channels starts with FTU_.

13.1 Add to Trend

In **Add to Trend** view you can browse the process database and select Process Objects to which you want to activate the data logging.

13.1.1 Making filters

By default all Process Objects are shown. If there are more than 10000 Process Objects in the process database, **<** and **>** buttons can be used for navigating back and forward.

There are several quick filters for known measurements. In addition to quick filters you can make a **Custom filter**.

Custom filter is defined either by writing the condition directly to the drop down list or by clicking **...** button, which opens a sub dialog for making the filter.

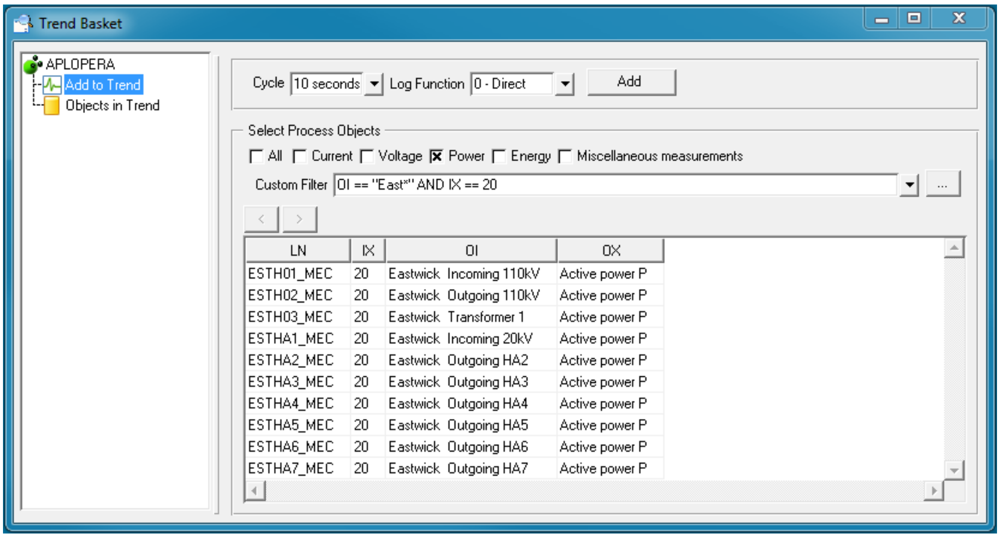


Figure 120: Example where both Quick filter and Custom filter is used

13.1.2 Adding process objects to trend

To add Process Objects to trend do the following:

1. Select Process Objects from the table
2. Define **Cycle** and **Log Function**
3. Click **Add** button
4. Confirm the operation

Process Object can be added only to one Cycle and Log Function. Objects that are already added to trend are shown as disabled.

If you have selected more Process Objects than there are available Data Objects, Process Objects cannot be added to trend. You can increase the number of Data Objects in file \sc\sa_lib\base\bbone\use\bgu_ta_setup.txt. Logged data will remain in already existing Data Objects.

13.2 Objects in Trend

In **Objects in Trend** view you can see the Process Objects that have active data logging. Also the logging cycle, Log Function and Datalog are shown for each process object.

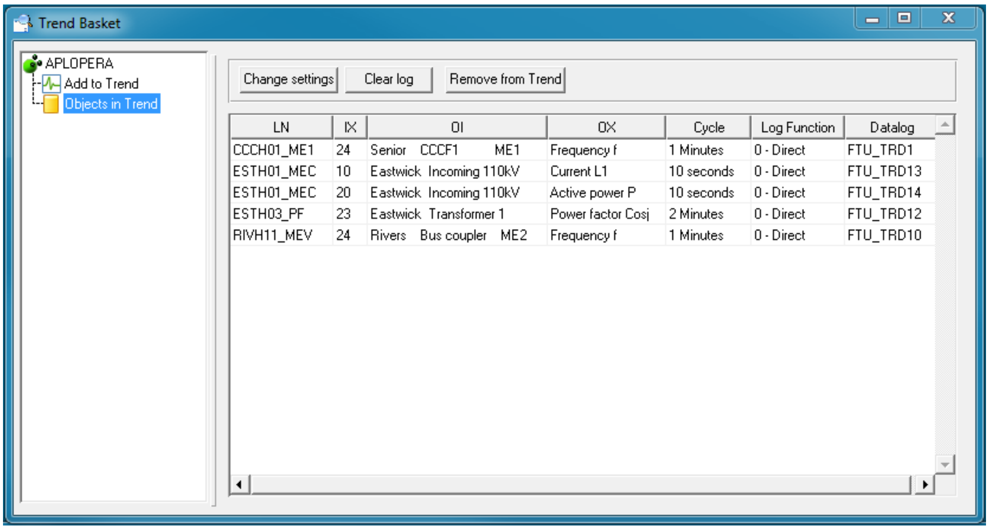


Figure 121: Objects in Trend

13.2.1 Changing settings

To change **Cycle** and/or **Log Function** of the object do the following:

1. Select objects from table
2. Click **Change Settings** button
3. Confirm the operation

Changing settings will clear all data from selected objects and data collection starts from scratch.

13.2.2 Clearing datalog

To clear datalog do the following:

1. Select objects from table
2. Click **Clear log** button
3. Confirm the operation

13.2.3 Removing objects from trend

To remove Process Objects from trend do the following:

1. Select object from table
2. Click **Remove from Trend** button
3. Confirm the operation

Section 14 Trends Display

Trends Display is used for trend analyses, as well as for showing measured values in curve or table form.

A trend is a time related follow-up of process data. All types of process objects, such as in and out data or binary, analogue and digital data can be illustrated as trends.

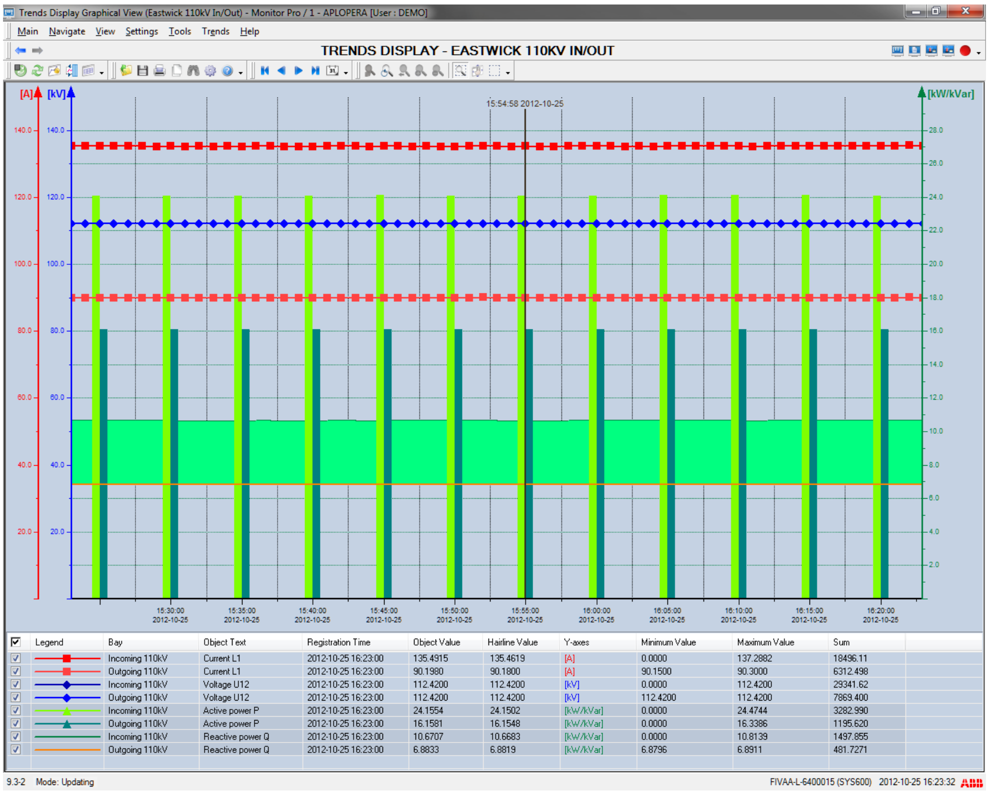


Figure 122: Trends Display

14.1 Introduction to trends

The Trends Display configuration includes a set of parameters (colors, fonts and so on), which are called the trend preconfigurations. The user can create, delete or apply an existing preconfiguration to the trend display.

14.2 Defining Trends coloring

The SYS600 Trends Display can be configured to use different colors for axes, curves, gridline and background. By using this functionality, the Trends Display can be adapted to display the events according to the convention required by the application.

14.2.1 Trends coloring example

The following is an example of trends coloring in an application, and the steps of how it has been defined in the **Color** dialog.

To modify the color settings in the Trends Display:

1. In the Trends Display, select **Settings/Display Settings/Graph Settings...** or select  **Graph Settings...** on the toolbar. The dialog opens, see Figure 123.
2. Select **Common settings/** and check the Use Color Scheme check box if the color from the current active Color Scheme should be used.
3. Click **Change**.
4. In the Color Setting Tool dialog, select **Graphical View** in the **Category** list.
5. Click the color field for any of the available colors.
6. Specify the color values in the Color dialog and click **OK**.
7. In the **Category** list, select **Tabular View** and change the colors in the same way.
8. Edit the settings on the **Axis properties** and **Curve Properties** tabs if necessary. Click **OK** to save the changes.

14.2.2 Setting the gridline color in Trends Display

The color used for the gridlines in the Trends Display can be configured in the following way:

1. In the Trends Display, select **Settings/Display Settings/Graph Settings...** or select  **Graph Settings...** on the toolbar. The Graph Settings... dialog opens, see Figure 123.

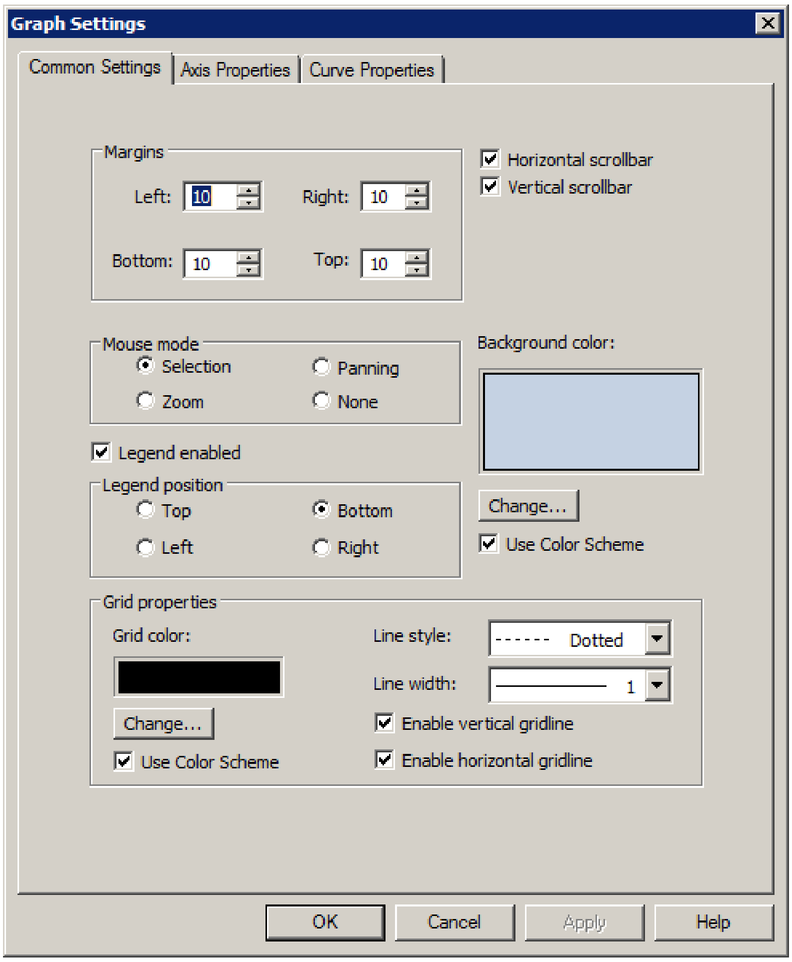


Figure 123: Graph Settings dialog

2. Click **Change** in the Grid properties to change the gridline color. The Color Settings Tool opens, see Figure 124.

Figure 124: The Color Settings Tool

3. Click the color box next to the **Gridline**. The Color dialog opens, see Figure 125.

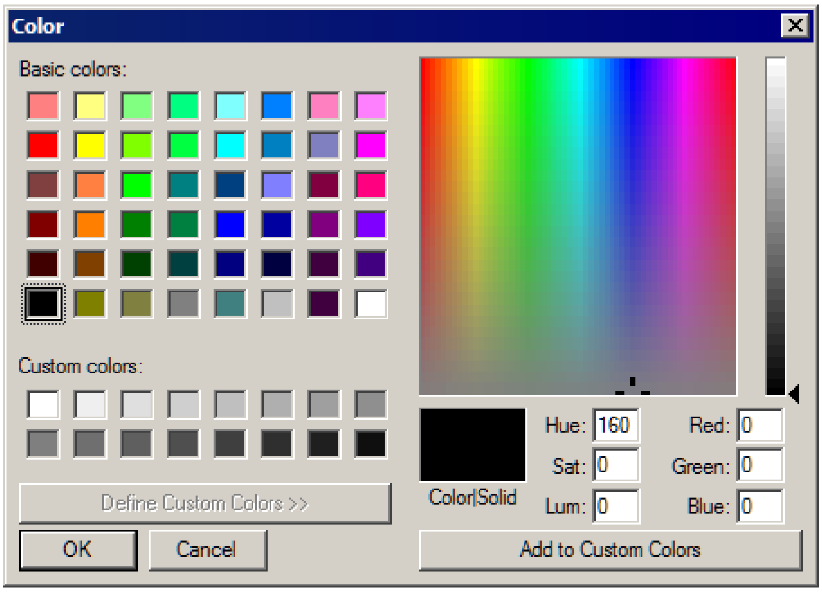


Figure 125: Color dialog

4. Select the color and click **OK**. The Color dialog closes and the selected color can be seen in the Gridline color box.

14.3 Trends Display parameters

The Trends Display has a set of parameters, such as the number of data logs and the maximum length of data logs that can be changed if required. These parameters are stored in the text file \sc\sa_lib\base\bbone\use\bgu_ta_setup.txt.

If some of the parameters must be changed, the file should be copied first to the sc \'Application name'\aplmod4\use directory.

The copied file can be edited using the SCIL Editor. The parameters are stored as SCIL variables. Therefore, ensure that the file syntax is correct when editing the file. The syntax can be checked using the Check Syntax option in the SCIL Editor.

The parameters are described in the following table:

Table 149: Parameter descriptions

Parameter	Description
Number of Datalog	The number of data logs created for the trends, that is, the maximum number of process objects that can be logged at the same time. The maximum value of this attribute depends on the size of the system and the hardware used, and it should be found out experimentally.
Max Length of Datalogs	The maximum number of registrations in one data log, that is, the maximum number of history values stored for one process object. The maximum value of this attribute is 65535.

The number and maximum length of the data logs used in the Trends Display are application specific parameters. Their effect on the system performance should always be measured by using the actual hardware. Too large values of these parameters may cause poor system performance.

After changing one of the parameters, restart the Trends Display and delete the existing data logs. After this, the changes take effect.

14.4 Application objects

Trends Display in the Monitor Pro application uses a set of application objects. These are Data Objects and Time Channels. All the needed application objects are created when the Trends Display is started for the first time. If the full set of needed objects exists before the first Trends Display start-up, the objects are not updated. To reinitialize the objects completely, delete the old objects and restart Trends Display.

The following two files are used in the Trends Display:

- \sc\sa_lib\base\bbone\use\bgu_ta_creat.txt
- \sc\sa_lib\base\bbone\lang<*>\sai_ta_objec.txt

Both files are used in the Trends Display to create and initialize the necessary application objects. The marking <*> refers to the current application language number.

14.4.1 Data objects

Trends Display uses data objects named FTU_TRD*, where * means 1 to the maximum number of trends. The default maximum trend number is 20 (NUMBER_OF_DATALOGS parameter of the bgu_ta_setup.txt file).

14.4.2 Time channels

Trends Display uses the following time channels:

- FTU_TRT10 (time channel for 10 seconds time interval)
- FTU_TRT30 (time channel for 30 seconds time interval)
- FTU_TRT60 (time channel for 1 minute time interval)
- FTU_TRT120 (time channel for 2 minutes time interval)
- FTU_TRT300 (time channel for 5 minutes time interval)
- FTU_TRT600 (time channel for 10 minutes time interval)

Section 15 Measurement Reports

Measurement Reports is used for various types of time related reports, such as hourly, daily, weekly, monthly and yearly reports.

Measurement Reports can be used for reporting:

- Energy (active, reactive)
- Current (for example bay level)
- Voltage (for example bay level)
- Frequency
- Temperature
- Other measured data

Generally, the reports are time related followups of process data, metered data, or calculated data. The data can be illustrated as reports in tabular or graphical form. The data for the reports is calculated and stored in real time. Report data is collected and calculated cyclically. The most common method is to collect raw data from the process, refine it and store it in the report database.

The collection and calculation of report data can be initiated in the following ways:

- At predefined time intervals
- As a result of a spontaneous event
- As a result of a calculation
- Based on a condition
- On the operator's request

15.1 Main components

The Measurement Reports function is based on a divided system structure, and it consists of the following main components:

- Report engineering: Configuration database (RCDB) and engineering tools integrated in the Object Navigator.
- Report data logger: Data sampling and calculation methods (BMU_* Application Objects).
- Report database (RDB): Data Objects.
- Report data viewer: Monitor Pro Measurement Report displays and data provider.

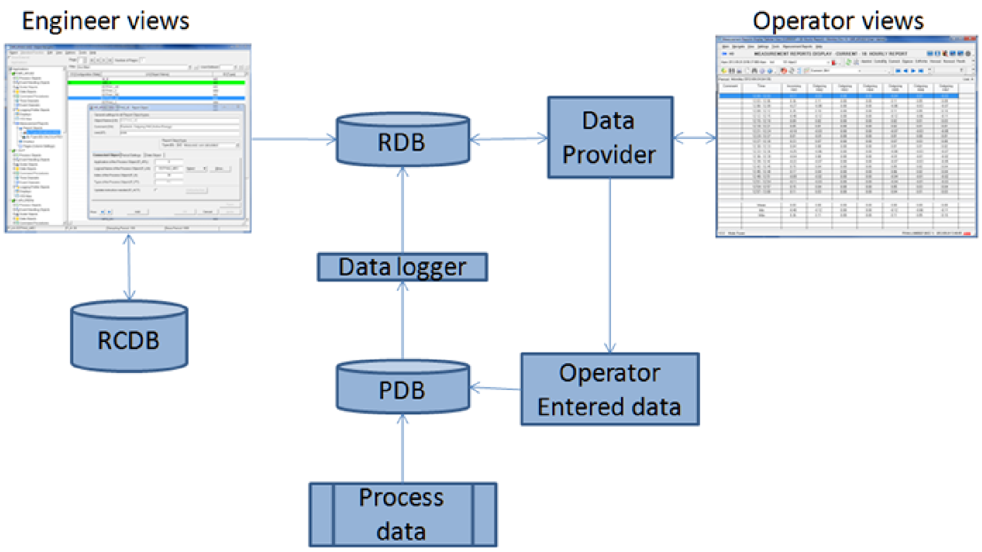


Figure 126: Measurement Reports component view

15.2 Report engineering steps

Engineering the Measurement Reports should be done in steps as listed underneath:

1. Specifying which Process Objects will be part of the Measurement Reports and are used as Report input data.
2. Setting the overall application specific Measurement Reports definitions.
3. Naming convention for all Report Object types.
4. Report Object type determination.
5. Monitor Pro navigation and menu construction design.
6. Report Objects creation.
   Using the Report Object Generator based on filters fulfilling the specifications from Step 1 is the most efficient way.
7. Report Page creation.
8. Report Display creation and connection to Report Page configuration from previous step.
9. Report Object type activation after detailed engineering has been done, e.g. specific Report Object settings.
10. Verification of all Measurement Reports functions in help of some input data simulation.

15.2.1 Step 1: Report input Data

Any Process Object of type Analogue Input, Digital Input or Pulse Counter can be used as input for the Report database.

Normally, Process Object values come from a real existing data collecting device, such as an IED or PLC, via the communication network.

If other than real process data needs to be collected, it can be done either on the user's request via a project specific data entering tool, or as event or time driven request using a Command Procedure. In both cases, the output must be written to a Process Object.

15.2.2 Step 2: Overall application specific definitions

After the Report Data input has been specified, the overall application specific Measurement Reports settings should be applied.

Changing some of these settings with a fully engineered configured and active Report database will cause lost of historical data.
For more detailed information about these application specific settings, see Section 15.3.1.

15.2.3 Step 3: Naming conventions

In order to easy recognize the relation between input data and Report Objects, the following naming convention is recommended:

"Prefix" + "Process Object LN" + "_" + "Process Object IX" + "Report Type ID".

15.2.4 Step 4: Report Object type determination

The used Report Object type should be determined based on the input data type. The two basic type groups for measured and calculated objects contain several types, each having a different purpose:

For Measured objects:

- Sum calculated (MS)
- Mean calculated (MM)
- Minimum calculated (MN)
- Maximum calculated (MX)
- Period value (MP)

For Calculated objects:

- Calculated based on sampled value (CS)
- Calculated based on period value (CP)
- Calculated based on sampled and period values (CB)

15.2.5 Step 5: Monitor Pro Measurement Reports navigation

Before starting with the detailed engineering, make sure that the Monitor Pro Measurement Reports navigation structure is fixed and accepted by the customer. For example:

Main level: Type of Measurement

CURRENT
ACTIVE ENERGY
VOLTAGE
....

Sub level: Location and Report Base type

Station A: Day Report
Station B: Week Report
....

Main level: Location

REGION A
REGION B
Substation X
....

Sub level: Measurement type and Report Base type

DAY Report CURRENT
ENERGY WEEK Report
....

15.2.6 Step 6: Report Object generation

It is recommended to utilize the Report Object Generator tool to create Report Objects. The tool is described in more detail in Section 15.4.

Detailed engineering for certain Report Objects can be done by using the Measurement Reports engineering tools integrated in the Object Navigator.

15.2.7 Step 7: Report Page generation

Every Report Display can contain a certain number of pages. Each page should not contain more than 20 columns.

15.2.8 Steps 9 and 10: Verification, activation and testing

After all the Measurement Reports configuration data has been created, the data can be activated and tested.

After a new or modified Report Display has been activated, all currently running Monitor Pro sessions must be closed and reopened. The navigation menu structure will only be reconstructed during Monitor Pro start-up.

15.3 Engineering aspects

All tasks related to Measurement Reports engineering can be performed with tools integrated in the Object Navigator.

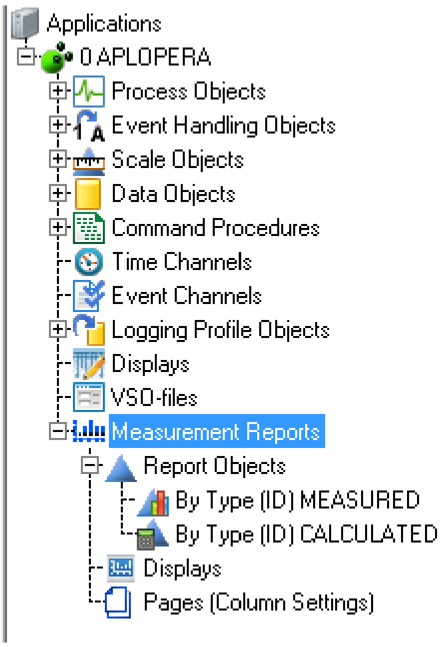


Figure 127: Measurement Reports objects in Object Navigator tree

The engineering tools are available only for local applications. The Measurement Reports branch will be hidden for external applications.

A valid license is needed for Measurement Reports for the user to be able to use/engineer the Measurement Reports. Otherwise, the Measurement Reports node will be hidden.

All Report Object type data will be shown in a table. No other view option is available.

The tabular view will have a similar Look-and-Feel as the one used for other Application Objects, and it is based on the view used for Process Objects.

15.3.1 Overall application specific configurations

If the Measurement Reports node is selected, some overall configurations can be made.

This configuration procedure has been moved here from Monitor Pro **Application Settings** dialog.

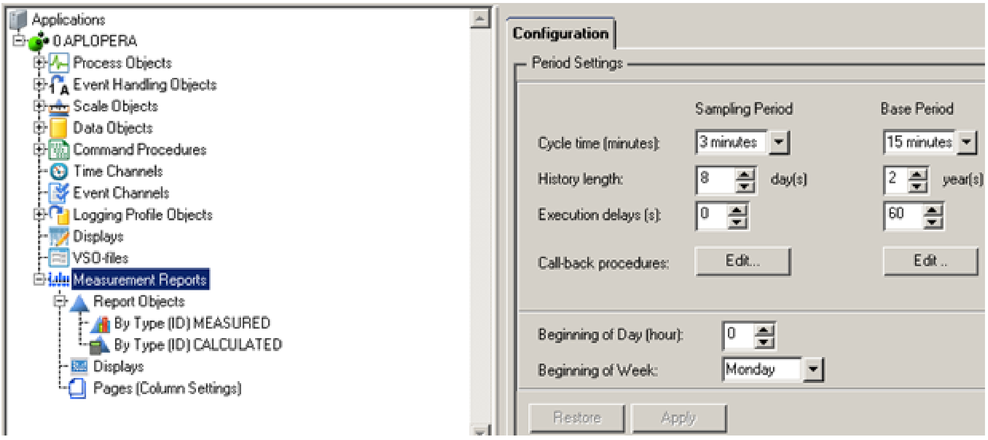


Figure 128: Application configurations

15.3.1.1 Period Settings

These period settings will be used as default settings for all new Report Objects.

If, for some Report Objects, different period settings are needed, they can be changed on Report Object level.

Changes will be applied to all Report Objects that use these default settings.

15.3.1.2 Sampling Period

Cycle time *

Range: 1, 2, 3, 5, 6, 10, 15, 20, 30 minutes

Default: 3 minutes

15.3.1.3 History Length for sampled values *

Range: 0-100 days

Default: 15 days

15.3.1.4 Base Period

Cycle time (BP)*

Range: 10, 15, 30 or 60 minutes

15.3.1.5 History length for base period values history *

Range: 0-5 years

Default: 2 years

15.3.1.6 Beginning of reporting day *

Range: 0–23 in hours after midnight

Default: 0

This setting defines at what time the reporting day starts. This time is also used in Day reports as start time for the data.

15.3.1.7 Beginning of reporting week

Range: Monday-Sunday

Default: Monday

This setting defines at what time the reporting week starts. Used for Week reports. Changing this setting will not cause any historical report data loss.

The definitions are valid for the entire application. Changing the definitions (marked with an asterisk (*)) causes loss of the existing history data because the Data Objects holding the report data need to be reconstructed.
The Report Object specific settings, which differ from these application settings, will not be overwritten. In this case, history data will also not be lost.

15.3.1.8 Execution delays

Ranges:

- 0–60 seconds for the sampling period, default value is 0 seconds
- 60–300 seconds for the base period, default value is 60 seconds

The execution delays can be defined both for base period cycle and for sampling period cycle. The execution delays must be used when process objects are not spontaneous, and the execution delays need an update command to be used before the current value is available in the process database. After time channels are triggered, an execution delay defines how long the program waits for before the evaluation of values starts.

When update commands are used, the commands are sent after the time channels are triggered. The data logger program reads the value from the process database after the execution delay.

The execution delay for sampled values must be less than both the delay for base period values and the smallest used sampling period.

The execution delay for base period values must be less than the smallest used sampling period.

15.3.1.9 Call-back procedure

15.3.1.10 For sampling cycle

A call-back procedure for the sampling period is executed when the other sampling routines have been executed. Therefore, it is possible to execute programs defined by the user every time the sampling routines have been executed.

15.3.1.11 Call-back procedure for period cycle

A call-back procedure for the base period is executed when the other base period routines have been executed. Therefore, it is possible to execute programs defined by the user every time the period routines have been executed.

15.3.2 Engineering tools common functionality

All configuration data will be shown in object tables.

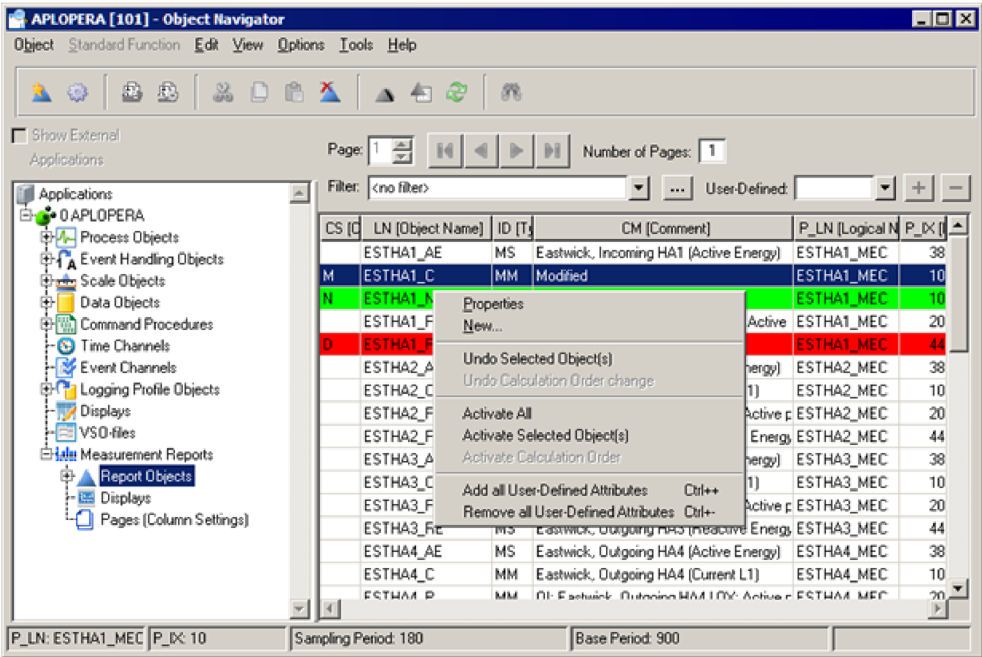


Figure 129: Measurement Reports object table

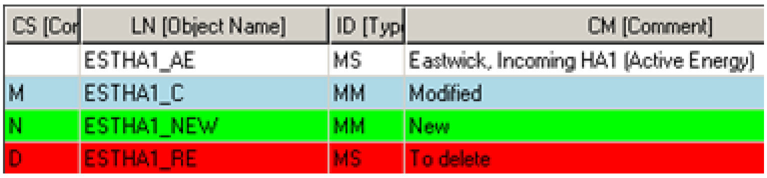
Three tables are available for Reports Objects: one for all Report Objects, one for measured objects and one for calculated objects.

The table length (number of lines per page) and the column size auto-adjustment option can be changed from the Object Navigator's **Options** menu.

15.3.2.1 Data presentation

Modifications, e.g. add new, remove (delete) or edit, to any report object type in existing object properties will not be immediately activated. All changes will be stored temporarily. This temporary stored data will be kept until the user has decided to store it permanently or explicitly undos the temporary changes. If, for any object, a temporary data set exists, the corresponding item in the object data table will be highlighted by using a different background color:

New	Green
to be Delete	Red
Modified	Blue
Active not modified	White



CS [Co	LN [Object Name]	ID [Typ	CM [Comment]
	ESTHA1_AE	MS	Eastwick, Incoming HA1 (Active Energy)
M	ESTHA1_C	MM	Modified
N	ESTHA1_NEW	MM	New
D	ESTHA1_RE	MS	To delete

Figure 130: Object configuration state coloring

Each report object type contains a set of attributes, which can be removed or added to the object table with the help of the **User Defined** attributes drop-down menu.

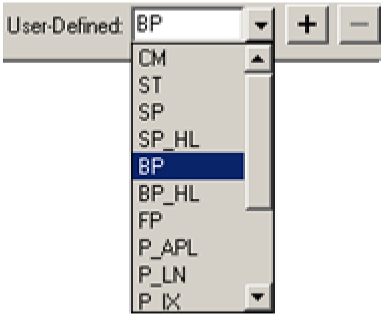


Figure 131: User-Defined attributes

15.3.2.2 Sorting

Double-clicking a table column title will sort the table based on the selected column data. The current sort order will be shown in the 5th field of the Status bar.

15.3.2.3 Page navigation

Page navigation works in a similar way as the other application objects in the Object Navigator.

15.3.2.4 Filter handling

With the filter dialog, the user can enter a condition to filter out certain objects.

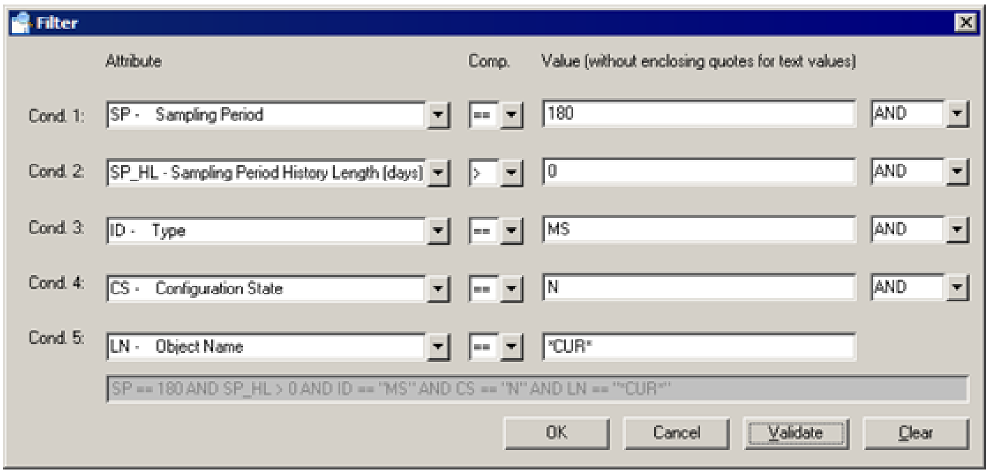


Figure 132: Filter dialog

All available object attributes can be used for the filter condition. Mixed logical filter operations (and/or in one condition) are not allowed. It is not possible to enter any condition freely.

For text value conditions, the leading and trailing quotes can be omitted.

Before activating the condition, the selected condition can be validated. In case the condition is invalid, an information dialog will be shown with details about the condition status.

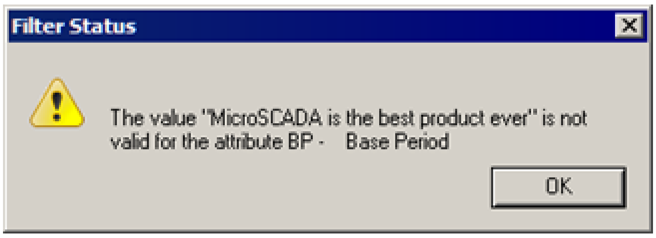


Figure 133: Filter condition validation dialog

15.3.2.5 Commands

The most common commands can be activated from the Toolbar, the Menu bar or with keyboard shortcuts.

15.3.2.6 Toolbar

Figure 134: Toolbar commands

1. **New:** Opens the Input dialog to enter a new object name.
2. **Properties:** Opens the Object Properties dialog.
3. **Delete:** Marks the selected object to be deleted if activated. New objects will be removed directly after confirmation is accepted.
4. **Refresh:** Refreshes the object table.

15.3.2.7 Menu bar commands

Figure 135: Menubar

The following additional menu bar commands are available:

- **Edit/Select All**: Selects all objects from the current page.
- **Edit/Deselect All**: Clears the selection.
- **Tools/Report Object Generator**: Opens the Report Object Generator tool.

15.3.2.8 Keyboard shortcuts

The following keyboard shortcuts can be used:

- Ctrl+N: New
- Ctrl+D: Delete
- Ctrl+A: Select All
- F5: Refresh
- Ctrl++: Add all user defined attributes
- Ctrl+-: Remove all user defined attributes

15.3.2.9 Context menu

The following functions are available from the object table context menu:

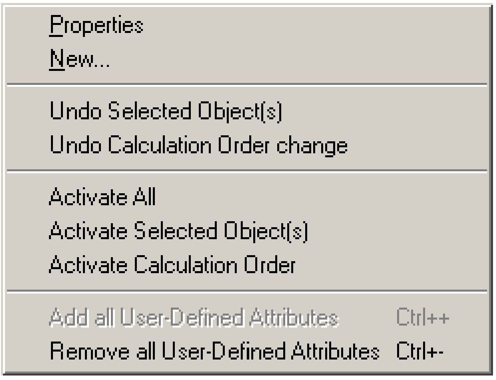


Figure 136: Object table context menu

- **Properties**: Open the Object Property dialog for the selected object.
- **New...**: Open the new object name input dialog.
- **Undo Selected Object(s)**: Undo all modifications for the selected object(s) that have not been activated already.
- **Undo Calculation Order change**: Undo possible calculation order changes.
- **Activate All**: Activate all modifications for all modified objects.
- **Activate Selected Object(s)**: Activate all modifications for selected modified objects.
- **Activate Calculation Order**: Activate possible calculation order changes.
- **Add all User-Defined Attributes**: Add all user-defined attributes to the object table.
- **Remove all User-Defined Attributes**: Remove all user-defined attributes from the object table.

15.3.2.10 Object Properties dialog, common functionality

For modified valid attribute values, the corresponding label will be colored blue.

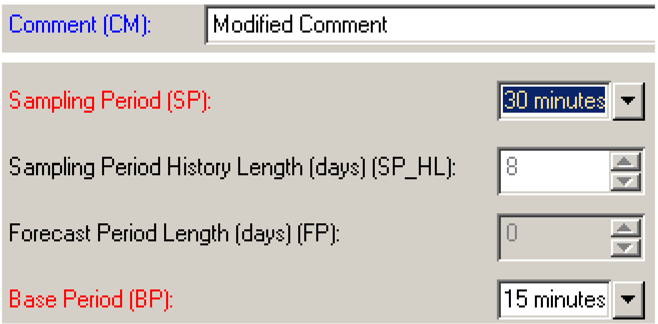


Figure 137: Property dialog coloring

In case the current attribute value is invalid, the corresponding label will be colored red. In the example above, the sampling period cycle has a larger value than the base period value, which causes the invalidation. For SCIL instruction type attributes, the code will be validated on update request. Invalid SCIL instructions cannot be updated to the corresponding attribute.

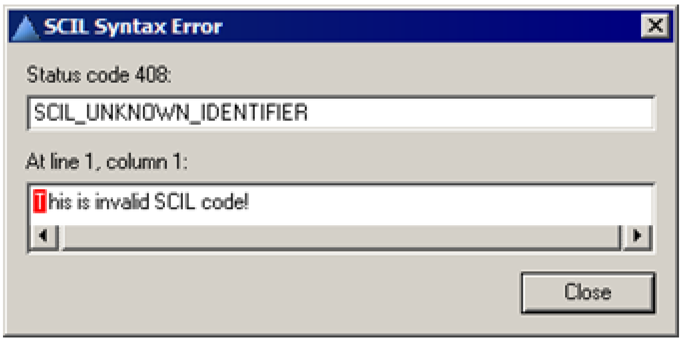


Figure 138: Invalid SCIL-instruction notification

Object modifications cannot be saved as long as any of the attributes has an invalid value. The Object Property dialog containing invalid attribute values will be closed without notification if the user clicks **Cancel** or presses the ESC key, navigates to another object or fetches the current valid configuration data. In case valid modifications are detected, a confirmation dialog will appear if the user attempts to navigate to another object or to click the **Cancel** or **Fetch** button.

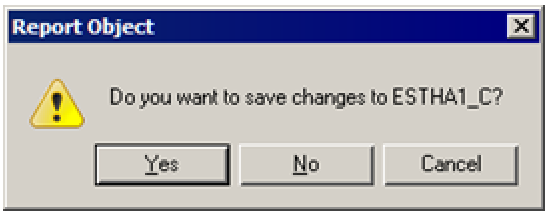


Figure 139: Modification save confirmation

15.3.2.11 Commands

Figure 140: Property dialog command buttons

Table 150: Property dialog commands

Button	Explanation
Row: Left and Right arrow	Navigate to the previous or next object in the object table.
Add	The **Add** button can be used to add multiple new objects. Clicking this button will append a single new character to the Logical Name and keep all the other settings. With this feature, an efficient way to add several objects having the same basic properties can be provided. It can be seen as a kind of copy/paste feature.
OK	When the **OK** button is pressed, the Report Object modifications are saved and the dialog is closed.
Cancel or the ESC key	Pressing the **Cancel** button discards all changes and closes the dialog.
Apply	When the **Apply** button is pressed, the Report Object modifications are saved and the dialog stays open.
Fetch	The last saved configuration will be restored to the property dialog.

15.3.3 General Object handling commands

15.3.3.1 Add new

Figure 141: New object name input dialog

The new object name will be validated before the detailed configuration can be made. If the new entered name is invalid, a notification dialog shows the name rule violation.

Figure 142: Object name invalid information dialog

15.3.3.2 Modify existing

To modify existing object configuration, either double-click the corresponding line in the object table or select the appropriate item in the menu or toolbar.

15.3.3.3 Delete new or active

Before an object will be deleted (new not active) or marked as to be deleted (active), the delete action has to be confirmed.

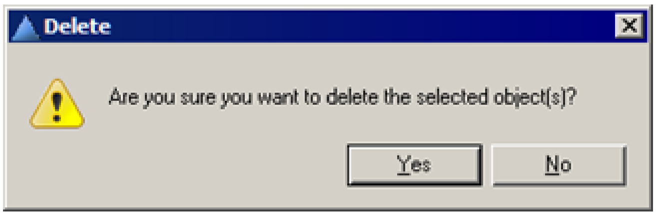


Figure 143: Delete confirmation dialog

Delete new:	New, not yet activated objects will be immediately and permanent removed from the configuration database. No undo function is available.
Delete active:	Object configuration will be marked as to be deleted in the configuration database. Final deletion will be done with the activation command.

15.3.3.4 Activation

Based on the selection in the object table, modified or new objects can be activated. The activation can be selected from the context menu.

15.3.3.5 Undo

Undo actions will be applied immediately, and no confirmation dialog will appear. The Undo function is only available for modified selected objects.

Table 151: UnDo actions

State (CS)	Action
New (N)	Same as Delete New Objects but without confirmation.
Deleted (D)	The tag to delete the object will be removed. If the object data has been modified before it has been tagged as to be deleted, the modified configuration will be restored.
Modified (M)	All modifications will be removed. The original active configuration data will be restored.

15.3.4 Report Objects

Each Report Object owns a set of attributes. These attributes describe the functionality for every object within the Reports Objects as there are, beside others:

- Which process data should be logged for the reports: Process Object connection
- How often should the process data be sampled: Sampling Period SP
- How often should the sampled data be calculated: Base Period BP
- How long should the sampled data and the calculated data be hold: History Length for sampling and base period SP_HL and BP_HL
- What type of calculation should be performed for the calculated (base period) values based on the sampled data: Type ID

The maximum number of Report Objects is limited to 10 000.

Report Object configuration tools are accessible from the Object Navigator object tree as shown in Figure 144.

Figure 144: Report Objects node in Object tree

The Report Objects node contains three navigation schemes:

- All Report Objects
- Navigation by Type ID for measured objects
- Navigation by Type ID for calculated objects

15.3.4.1 Attributes

Table 152: Attributes for measured Report Objects

Description	Attribute	m/o	Type	Range	Remarks
Configuration State	CS	m	Text	M, N, D	M=Modified, N=New, Deleted, empty = Active
Logical name	LN	m	Text	1..59 characters	
Report Object Type	ID	m	Text	MS, MN, MX, MM	Measured: MS = sum calculated MN = min calculated MX = max calculated MM = mean calculated MP = period value
Comment	CM	o	Text	0..240 characters	'P_LN':POX'P_IX' from connected Process Object will be used.
Unit	ST	o	Text	0..255 characters	'P_LN':PST'P_IX' from connected Process Object will be used.
Sampling Period	SP	o	Integer	1, 2, 3, 5, 6, 10, 15, 20, 30	Minutes, shown as seconds in the object table.
Sampling Period history length	SP_HL	o	Integer	0..365	Days
Base Period	BP	o	Integer	10, 15, 30, 60	Minutes, shown as seconds in the object table.
Base Period history length	BP_HL	o	Integer	0..5 years	The current year plus the number of selected BP_HL years.
Forecast period	FP	o	Integer	0..1 day	
Application of Process Object	P_APL	o	Integer	0..255	For performance reason, it is recommended to connect Process Objects from the local application only.
LN of Process Object	P_LN	o	Text	0..255	From connected Process Object.
IX of Process Object	P_IX	o	Integer	0..9999	From connected Process Object.
PT of Process Object	P_PT	o	Text	AI, DI, PC	'P_LN':PPT'P_IX' from connected Process Object.
Value type of Data Object	VT	o	Text	INTEGER, REAL	Default: Integer for P_PT=PC (Pulse Counter) else Real. For new objects, other combinations can also be selected.
Table continues on next page					

Description	Attribute	m/o	Type	Range	Remarks
Max value of Pulse counter	PS	o	Integer	0..MAX_INTEGER	Only available for VT = INTEGER. For PS > 0 the Data Object logging function "5 – Pulse Difference" will be used. For PS=0, the Data Object logging function "4 – Difference" will be used.
Linear scale of Process Object	P_SC	o	Real	-1,000,000.00 ... 1,000,000.00	The raw Process Object value will be multiplied with this value before it is stored in the Data Object.
Update instruction needed	P_ACT	o	Boolean	True, false	If true, the P_IN instruction will be executed in every sampling period.
Update Instruction	P_IN	-	Text vector	SCIL commands	The update instruction will be executed before the data will be logged in the sampling period. For Type ID - MP, this instruction will be executed before the data will be logged in the base period. This instruction can be used to retrieve the process data, calculate or simulate some Process Object value.

Table 153: Attributes for calculated Report Objects

Description	Attribute	m/o	Type	Range	Remarks
Report Object Type	ID	m	Text	CS, CP, CB	Calculated: CS (sampled value) CP (period values) CB (both, sampled and period values)
Input Report Objects	DATA_IN	-	Text vector		
Calculation operation	OPER	-	Text vector	SCIL commands	The calculation operation can be a single line calculation or a multi line SCIL program with a #RETURN command.

Attributes marked with m (mandatory) are always visible. The other attributes are optional and can be added and removed from the table with the selection box for user defined attribute columns. Attributes marked with a minus (–) sign cannot be added to the table at all, for example if the data type for that attribute is a vector of SCIL commands.

15.3.4.2 Status bar

If an object is selected, the details shown in the Status bar depend on the selected Report Object type. Details can be found in the table below.

Table 154: Statusbar for Report Objects

Selection / Object Type	Attribute				
	Field 1	Field 2	Field 3	Field 4	Field 5
No object selected			Number of objects in current view	Total number of objects	Sorting order
MS, MX, MN, MM	P_LN	P_IX	SP	BP	Sorting order
	P_LN	P_IX		BP	Sorting order
CS, CP, CB	LN	Calculation Order change notification			Sorting order

15.3.4.3 Properties Dialog

The Properties dialog is designed to add new Report Objects, as well as modify or copy existing (the **Add** button) Report Objects.

Report Objects marked as to be deleted cannot be modified. In that case, all attributes are disabled.

The upper part of the Properties dialog is the same for all Report Object types. The lower part contains the object type specific attributes separated in three tabs. This lower part is invisible for new Report Objects as long the Report Object type (ID) has not been selected.

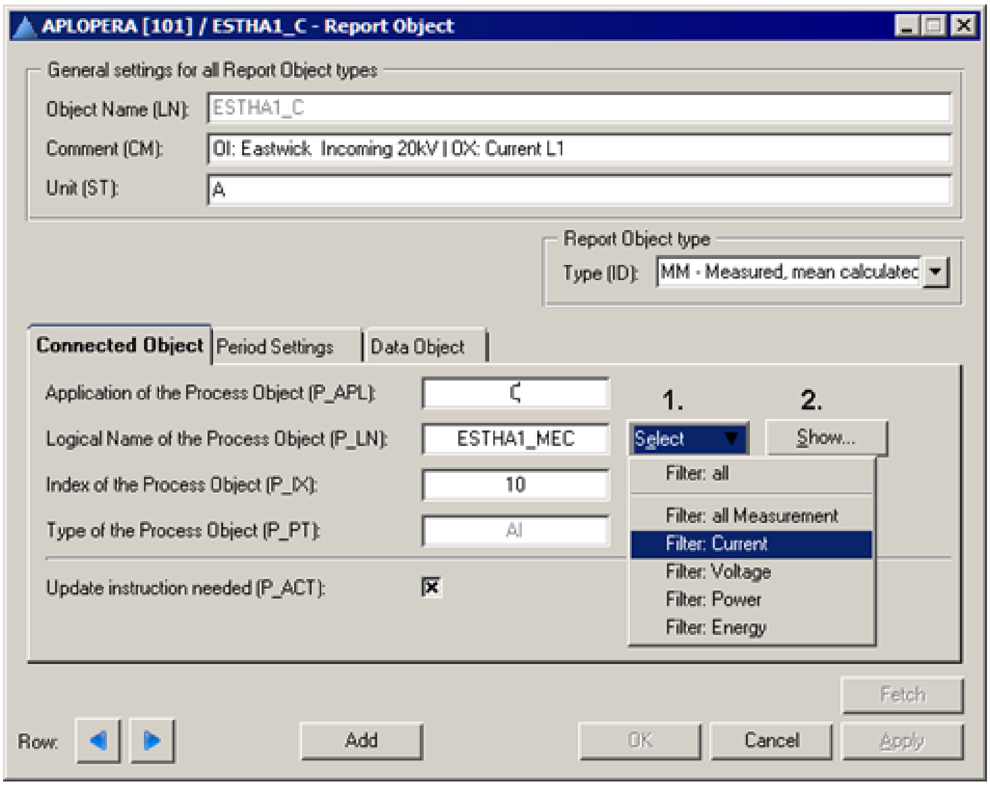


Figure 145: Properties dialog for measured Report Objects

15.3.4.4 Connected Object tab

In this tab, the Process Object connection can be selected. If needed, update instructions, which will get executed before the data sampling and calculation takes place, can be defined here.

The Process Object properties dialog can be loaded for the connected Process Object (the **Show...** button). A new connection can be established using the Application Object selector (the **Select...** button).

15.3.4.5 Period Settings tab

Figure 146: Period Settings tab in Properties dialog for Report Objects

In this tab, the value for the sampling and base (calculation) period can be selected. These values can either be retrieved from the default application settings, or some object specific setting can be applied.

15.3.4.6 Data Object tab

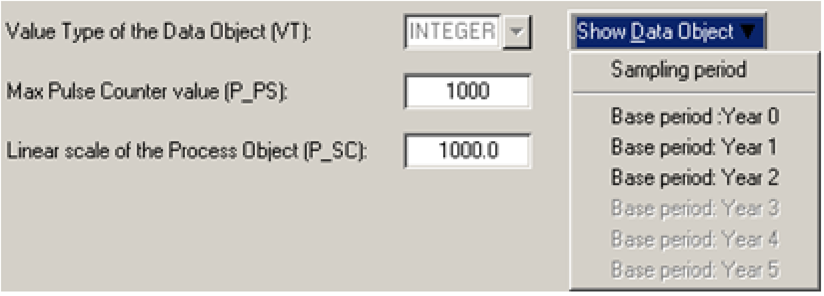


Figure 147: Data Object tab in Properties dialog for Report Objects

In this tab, the Data Objects which store the sampled (sampling period) and the calculated values (base period) can be selected, and the Data Object form can be loaded for all existing Data Objects (the **Show Data Object** button).

15.3.4.7 Calculation tab

The properties dialog for calculated Report Objects has no Connected Object tab, but it has a special ***Calculation*** tab for the calculation settings.

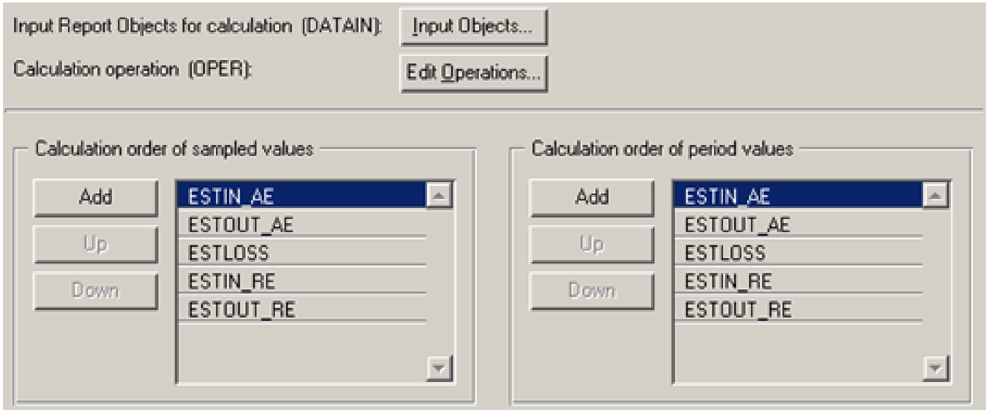


Figure 148: Calculation tab for calculated Report Objects

Within this tab, the input Report Object names and the calculation instruction can be added with the help of the Visual SCIL Program Editor.

The period settings for calculated objects cannot be changed. They will be automatically set to the period settings attribute values from the entered input Report Objects.

The input Report Objects must use the same period settings in order to be validated.

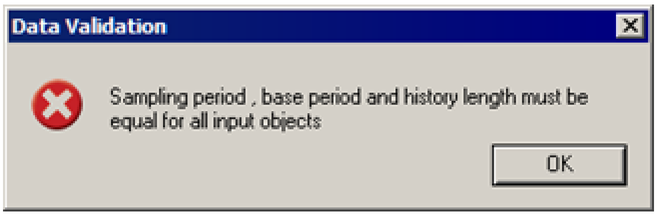


Figure 149: Period setting violation warning

15.3.4.8 Calculation operation (OPER)

The calculation operation can be an single line calculation, or a multi line SCIL program with a #RETURN command.

The input parameters can be referenced with variables %OBJ1..%OBJn.

The %OBJ1 is the first report object from the input parameter list and %OBJn is the last one.

If there is only one input parameter, the %OBJ variable without index number refers to the input parameter.

The type of a result calculation operation can be INTEGER or REAL, or a vector of the types.

Examples:

```
;Single line operation for one input Report Object
%OBJ + 100

;Single line operation for two input Report Objects
%OBJ1 DIV %OBJ2

;Multi line operation
@i1 = %OBJ1          ;Input Report Object 1 (first input Object in DATAIN
@i2 = %OBJ2          ;Input Report Object 2 (second ..)
@i3 = %OBJ3          ;Input Report Object 3 (third  ..)
#return %i1 * %i2 + %i3     ;return calculation result
```

15.3.4.9 Calculation order

The two Calculation order lists for sampled and base period values show the current order of all existing calculated objects. The lists can also be used to change the order of the current selected object.

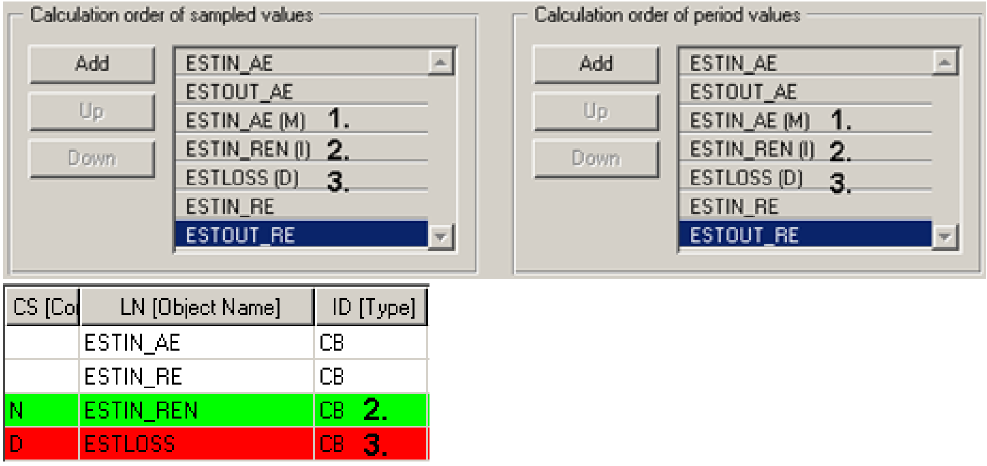


Figure 150: Calculation order visibility

1. Calculation order modified, shown in the Status bar if the calculated Report Object is selected in the object table.
2. New not activated calculated Report Object.
3. Calculated Report Object marked as to be deleted.

The changed calculation order can be activated or removed (**Undo**) with the help of the Object table context menu:

- Undo Calculation Order change
  Used to undo possible changes in the calculation order.
- Activate Calculation Order change
  used to activate possible changes in the calculation order.

15.3.4.10 Attribute validation rules

Before a new or modified Report Object configuration can be saved, a certain set of attributes must have valid values. The rules for these attributes are listed in the table below.

Table 155: Attribute validation rules for Report Objects

Attribute	Validation rules
LN	Valid Application_Object_Name (no dots), unique, should not end with numerical character 0 to 5. The last rule to avoid a possible name conflict with Data Objects used for base period, where the running year number will be appended to the name.
SP	Smaller than BP
BP	Bigger than SP
P_*	Connected Process Object must exist in the selected application.
P_IN	Valid SCIL instruction
For calculated Report Objects only	
DATA_IN	Entered Report Objects must exist and they must have the same period settings.
OPER	Valid SCIL instruction

15.3.5 Report Displays

Each Report Display object has a set of attributes.

These attributes describe the functionality for every object within the Reports as there are, beside others:

- Which time range should be shown: Report Base RB
- How should the Monitor Pro navigation menu look like: Report Type RT and Display Header DH
- Which report data should be shown: Page Configuration PC
- Which summary information and how many should be included: Summary Information SU1 to SU4

The following time related Report Display base types are supported:

- Hourly report (time resolution: 1,2,3,5,6,10,15,20, or 30 minutes), can be used only for sampling period values
- Daily report (time resolution: 10, 15, 30 or 60 minutes)
- Weekly report (time resolution: 1 day)
- Monthly report (time resolution: 1 day)
- Yearly report (time resolution: 1 month)

Report Display configuration tools are accessible from the Object Navigator object tree as shown in the figure below.

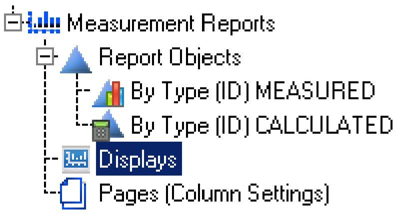


Figure 151: Report Displays node in Object tree

The maximum number of Report Displays is limited to 1 000.

The report data columns to be shown within the Report Display are configured in the Report Page, which must be connected to the Report Display configuration.

15.3.5.1 Attributes

Table 156: Attributes for Report Displays

Description	Attribute	m/o	Type	Range	Remarks
Configuration State	CS	m	Text	M, N, D	
Display Id	LN	m	Text	FIXED	
Report Base	RB	m	Text	HOUR, DAY_10, DAY_15, DAY_30, DAY_60, WEEK, MONTH, YEAR	
Report Type	RT	m	Text	Pre-defined types for all report base types: • Active Energy • Reactive Energy • Active Power • Reactive Power • Current • Voltage • Temperature • User defined	Used for Monitor Pro/Navigation/ Measurement Reports/ menu
Display Header	DH	m	Text	0..100 character	Used for Monitor Pro navigation/ Measurement Reports/Report type/sub menu
Table continues on next page					

Description	Attribute	m/o	Type	Range	Remarks
Comment	CM	o	Text	0..255 characters	
Unit	ST	o	Text	0..10	
Scale	SC	o	Real	-1,000 .. 1,000	
Interval	TI	o	Integer	1, 2, 3, 5, 6, 10, 15, 30	Minutes, only available for Report Base = HOUR
Summary Information	SU1..4	o	Text	SUM, MIN, MAX, MEAN, NONE, In-Day periods if defined in Calendar Tool	SU 1..3: RB = HOUR, WEEK SU 1..4: RB = DAY_xx SU 1: RB = MONTH and YEAR
Page Configuration	PC	o	Text		Link to Page configuration
Authorization Group	AG	o	Text	REPORTS, GENERAL, other	
Forecast area enabled	FC	o	Boolean	True or false	

Attributes marked with m (mandatory) are always visible. The others are optional and can be added and removed from the table with the selection box for user-defined attribute columns.

15.3.5.2 Statusbar

Table 157: Statusbar for Report Displays

Selection / Base Type	Attribute				
	Field 1	Field 2	Field 3	Field 4	Field 5
No object selected			Number of objects in current view	Total number of objects	Sorting order
All	ST	SC	SP	BP	Sorting order

15.3.5.3 Properties Dialog

The properties dialog is designed to add new, modify or copy (Add button) existing Report Display configurations.

The properties dialog replaces the Standard Function installation and configuration tool for Measurement Reports.

Report Displays marked as to delete cannot be modified. In this case all attributes are disabled.

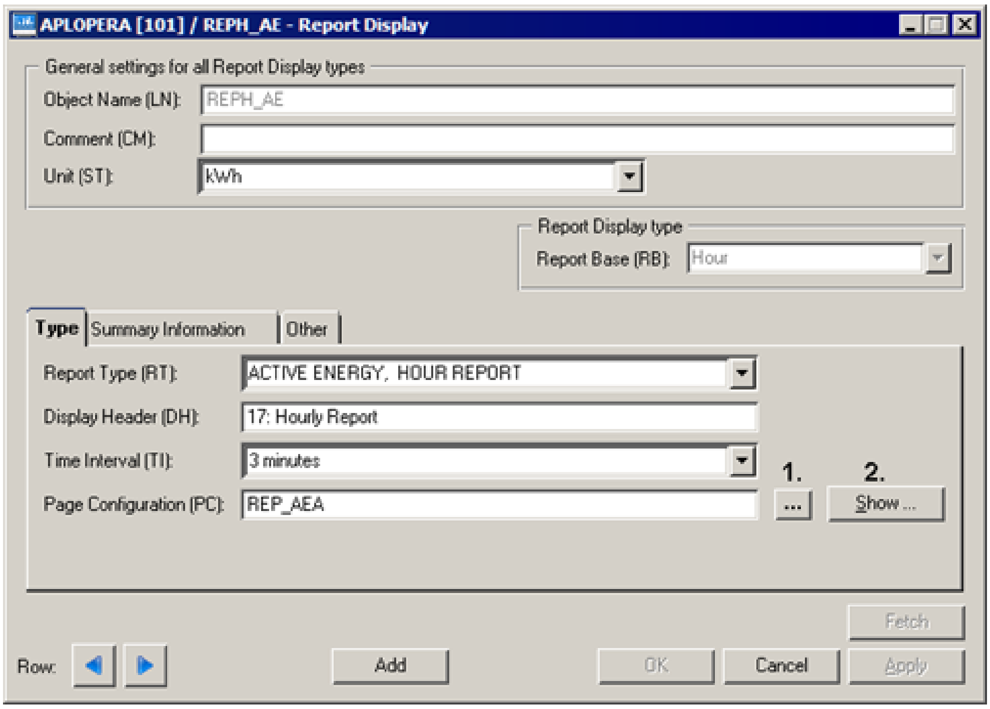


Figure 152: Properties dialog for Report Displays

15.3.5.4 Type tab

1. Connect to a existing Page Configuration using the Application Object selector tool
2. Open the property dialog for the current selected Page Configuration

The values for Report Type and the Display Header attribute are used to construct the Monitor Pro Measurement Report navigation menu. Text succeeding a comma in the Report Type attribute will not be shown in the menu and can be used as additional information.

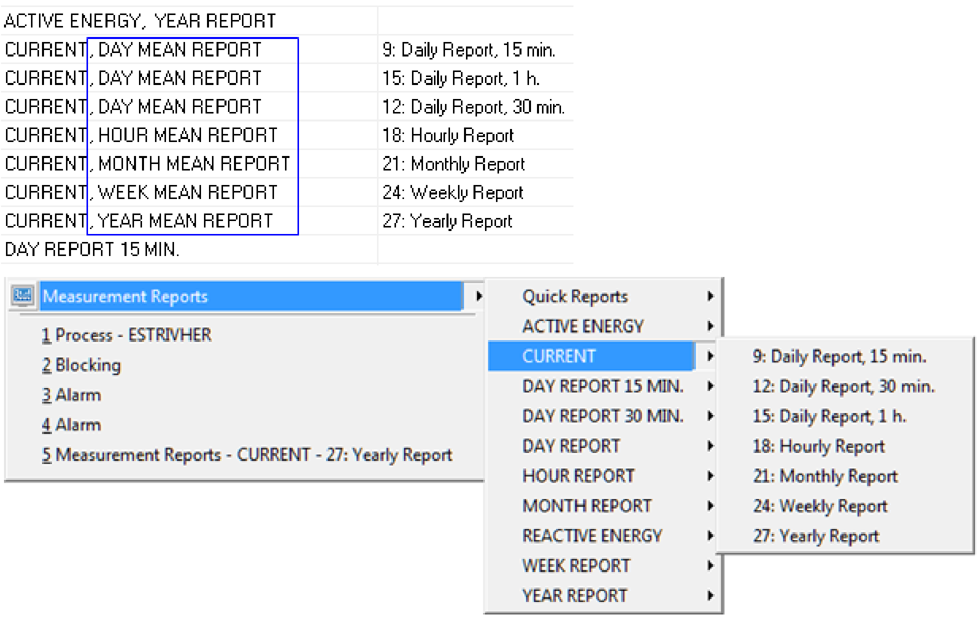


Figure 153: Relation Report Display attributes and Monitor Pro Navigation menu

Using numerical character's as pre-fix for both menu parts helps to get the menu constructed in a desired order.

15.3.5.5 Summary Information tab

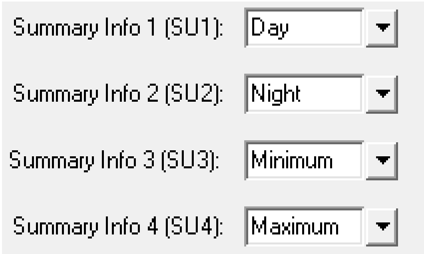


Figure 154: Summary Information tab in Properties dialog for Report Displays

The time range used for the Day, Night or other In-Day periods can be defined in help of the Calendar Tool.

15.3.5.6 Other tab

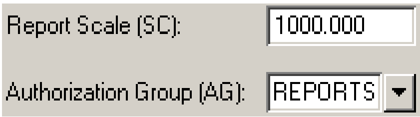


Figure 155: Other tab in Properties dialog for Report Displays

It is recommended to use the pre-defined authorization group REPORTS.

The Report Scale attribute can be used to multiply all values shown in the selected Report Display with a factor to e.g. toggle the sign (SC = -1.0) or to change the value range from kV toV (SC = 1000) or vice versa (SC = 0,001).

15.3.5.7 Attribute validation rules

Before a new or modified Report Display configuration can be saved a certain set of attributes must have valid values. The rules for this attributes are listed in below table.

Table 158: Attribute validation rules for Report Displays

Attribute	Validation rules
LN	Valid Application_Object_Name (no dots), unique
RT	Not empty
DH	Not empty
PC	Selected Report Page configuration must exist

15.3.6 Report Page

The maximum number of Report Pages is limited to 1 000.

Each Report Page object has a set of attributes. These attributes describe the functionality for every object within the Reports as there are, beside others:

- Which report data should be shown: Column Data
- How many pages within one Report Display should be shown: Number of pages NP
- How should the report data be presented: Decimals
- How should report data editing be treated: Editable and manual entry tagging

The Report Page configuration tools are accessible from the Object Navigator object tree as shown in the figure below.

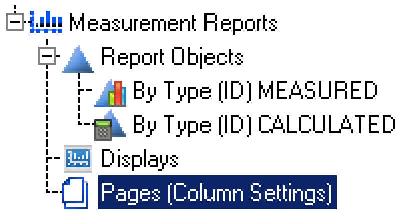


Figure 156: Page node in Object tree

Each Report Page configuration can be used for all Report Display base types.

It is not necessary to create different Page configurations for, e.g. HOUR and DAY based Report Displays, if the same Report data should be presented.

15.3.6.1 Attributes

Table 159: Attributes for Report Pages

Description	Attribute	m/o	Type	Range	Remarks
Configuration State	CS	m	Text	M, N, D	M=Modified, N=New, Deleted, empty = Active
Page Id	LN	m	Text		
Comment	CM	m	Text		
Number of pages	NP	m	Integer		
Page data	List				

Table 160: Page data

Description	Type	Range	Remarks
Page Title	Text	0..255 characters	
Number of columns	Integer	1..50	It is recommended not to use more than 20 columns per page.
Column data	List		

Table 161: Column data

Description	Type	Range	Remarks
Title	Text vector	1 or 2 lines	
Report Object name(s)	Vector	1 .. Max. 10 Report Objects	
Decimals	Integer	0 .. 9	
Table continues on next page			

Description	Type	Range	Remarks
Editable	Boolean	True / False	
Manual entry tagging	Boolean	True / False	True 206.0 m False 207.0
Operation	Vector	SCIL commands	

15.3.6.2 Status bar

If a configuration is selected, the shown details depend on the selected Report Page type. Details can be found in the table below.

Table 162: Status bar for Report Pages

Selection	Attribute				
	Field 1	Field 2	Field 3	Field 4	Field 5
No object selected			Number of objects in current view	Total number of objects	Sorting order
All	LN	CM			Sorting order

15.3.6.3 Properties Dialog

The properties dialog is designed to create new Report Page configurations, or modify or copy (Add) the existing Report Page configurations.

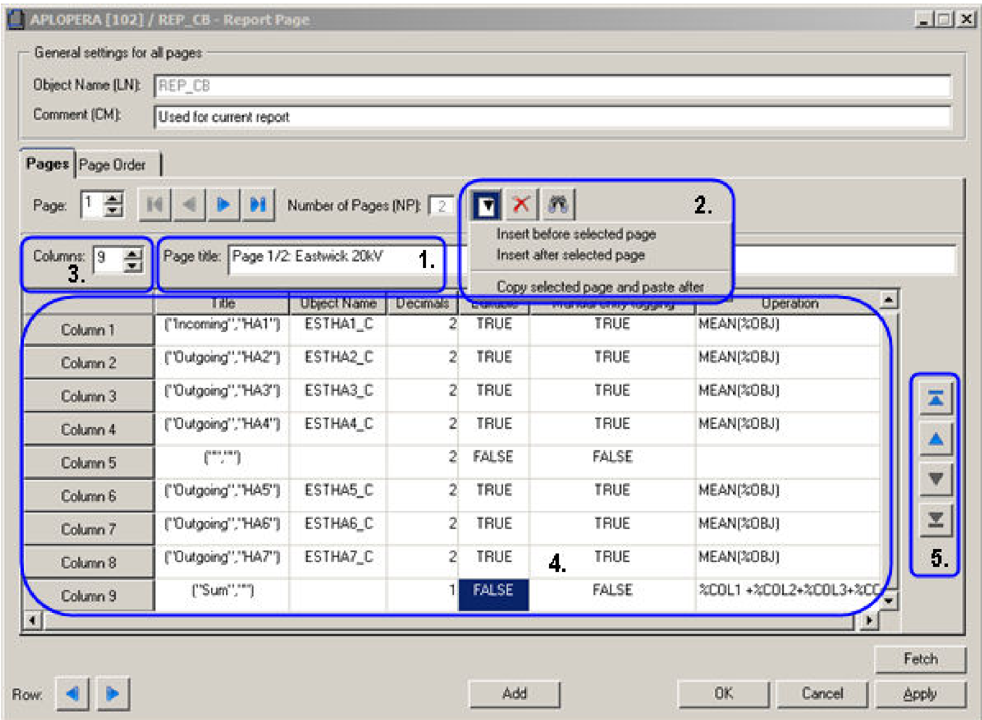


Figure 157: Properties dialog for Report Pages

1. Text field to enter a title for the selected page.
2. Insert new page before or after current selected page, copy selected page and paste after, remove the selected page, open the find/replace dialog. The find/replace dialog can be

used to replace text patterns within the title, Object Name or operation column for the selected page. This can be usefull for a copied page to replace e.g. the bay designation.

3. Using the numeric spinner up button will add a new column. The down button removes the last column.
4. Edit the column properties for each report object. Double-click with the left mouse button will activate the cell data edit handler.
5. Buttons to change the position of the selected column.

15.3.6.4 Column Data

Below the corresponding column data input functions are described more detailed.

15.3.6.5 Column Title

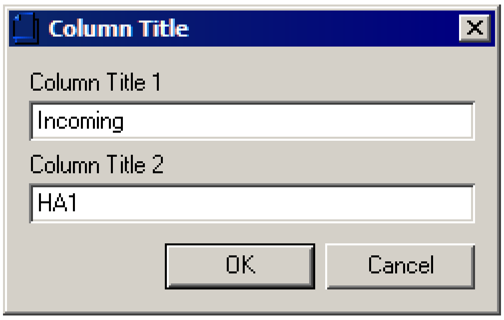


Figure 158: Report Page column title input dialog

At least one column title must be configured.

15.3.6.6 Object name(s)

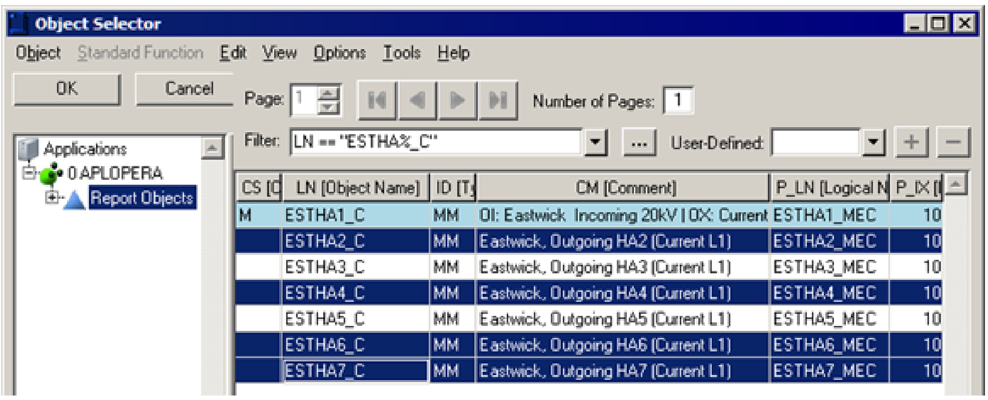
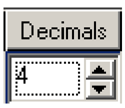
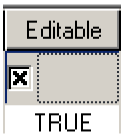


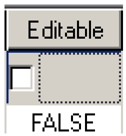
Figure 159: Report Object selection for Report Page

The Report Object names which should be added to the selected column can be freely selected using the standard selection methods. If more than one Report Object has been added some operation must be specified. If no operation is defined the summary of all selected objects will be used as default calculation.

15.3.6.7 Cell Data edit handler

Number of **decimals** to be shown 

Edit Report Object data value in Report Display enabled 

disabled 

Manual entry tagging enabled/disabled. If this option is enabled manual entered data will be shown with different color and a value suffix.

To apply the cell data changes another table cell must be selected.

15.3.6.8 Operation

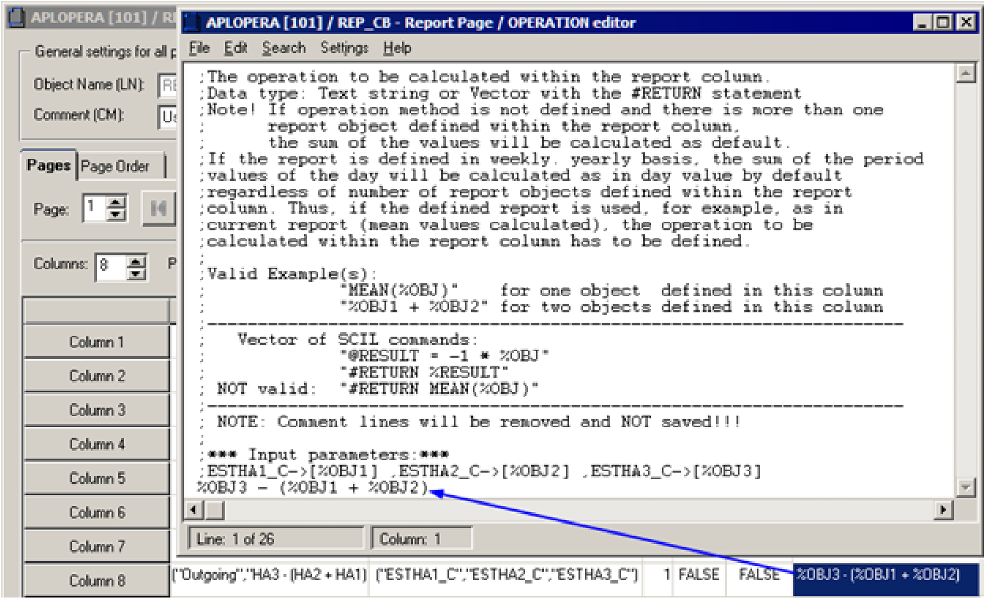


Figure 160: Report Page Operation editor

Valid SCIL examples can be seen in commented operation help part.

For the operation to be calculated within the report column all the arithmetical functions supported by SCIL can be used.

- If only one report object is defined to be shown within the report column, it can be referenced with the variable %OBJ.
  The variable includes the measured values of the row in question.
  If one hour is shown on the row of the report picture and base period cycle within the report application is 15 min, the variable holds 4 values. If a day is shown in the row of the report picture, the variable holds 96 values, and so on.
  The data type of the variable: Vector.
- If several report objects are defined to be shown within the report column, a single object can be referenced with the variable %OBJ'vector element index'.
  The data type of the variable: Vector.
- The column object can be referenced with the variable %COL'column number'. The column object has to be calculated before making a quotation to it. That means that e.g. %COL5 cannot be used in the operation of COL1 but %COL5 can be used in COL6 operation.
  The data type of the argument: Vector.

When the calculation procedure is defined by SCIL programming, the operation must be a text vector containing a #RETURN statement.

The result type of the calculation operation may be integer, real or a vector of the types.

Example:

```
@RES = SUM(%OBJ)
#IF %RES(1) > 10 #THEN #RETURN 1
#ELSE #RETURN 0
```

15.3.6.9 Page Order tab

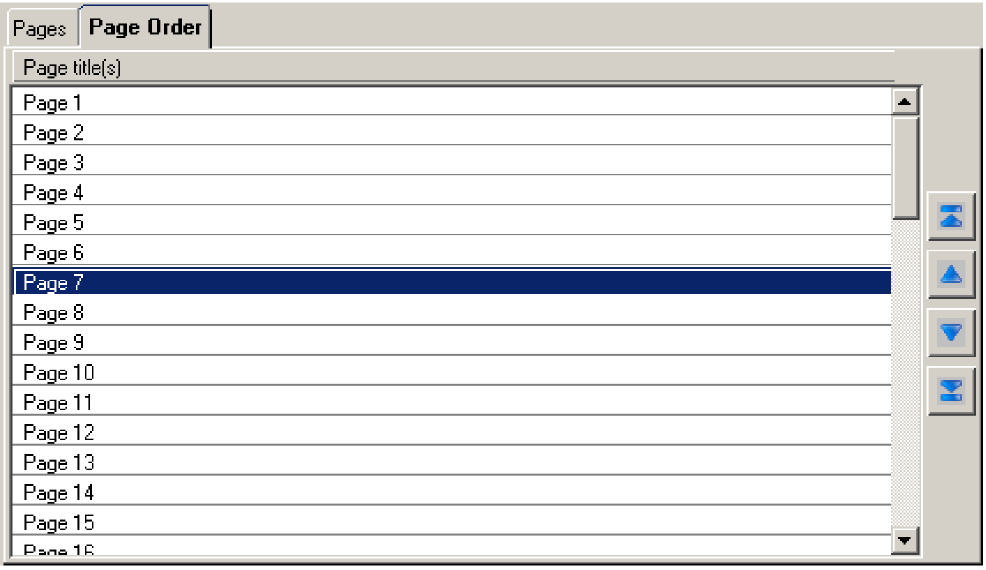


Figure 161: Report Page order tab

In help of this tab the position for the selected page can be changed.

15.3.6.10 Attribute validation rules

Before a new or modified Report Page configuration can be saved, a certain set of attributes must have valid values. The rules for this attributes are listed in the table below.

Table 163: Data attributes for Report Displays

Attribute	Validation rules
LN	Valid Application_Object_Name (no dots), unique.
Page data	At least one page with valid column data.
Column data	All columns must have a title defined and at least one valid Report Object assigned to it. All column data combinations are allowed. If the title is empty but not the "Object Name" or "Operation" column the title cell will be colored yellow as a kind of warning. Changes are allowed to get saved.

15.4 Report Object generator

The purpose of this tool is to provide an easy to use and fast way to create or modify the Report Object configuration database.

Based on user defined rules, the Report Objects will be created or can be modified if they already exist.

This tool will not activate new or modified Report Objects. Activation needs to be done the same way as for single Report Objects from the Report Object table, see Section 15.3.3.

15.4.1 Input Process Objects tab

First, the Process Objects used as input for the Report Object generation must be defined. The mandatory condition cannot be overruled, and will always be used to make sure that only supported measurements will be included. In addition to the mandatory conditions, some user conditions can be added to filter out unnecessary measurements.

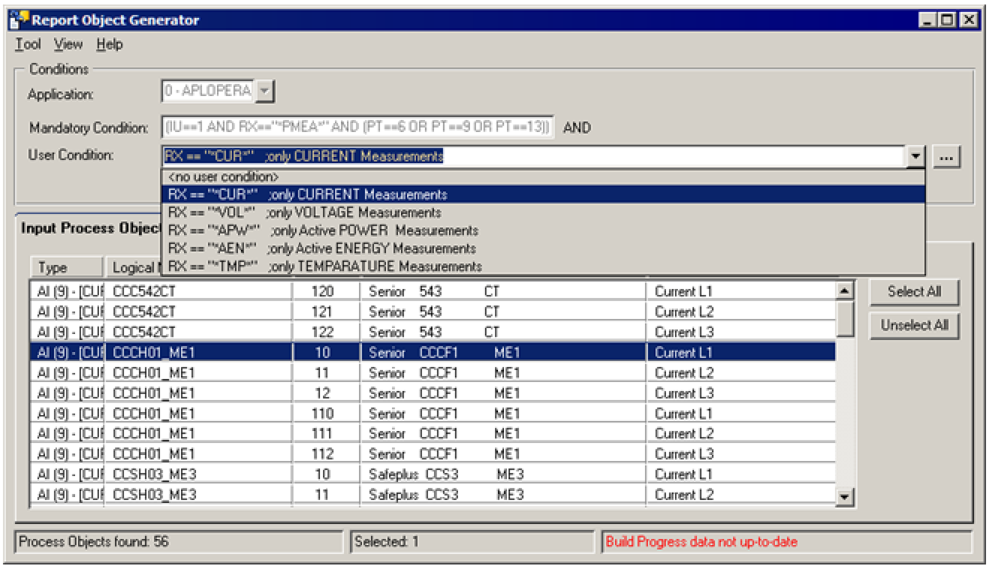


Figure 162: Report Object Generator Input Process Objects tab

Five pre-defined conditions are available for the following measurement types:

1. Current
2. Voltage
3. Active Power
4. Energy
5. Temperature

It is recommended to handle different types of measurements in separate steps, for example to create all Report Objects for current measurement first, then energy measurements etc.

The input list will be refreshed automatically with every user condition change.

To include only some of the found Process Objects from the input list for Report Object generation, the Process Objects must be selected. If no Process Object from the input list is selected, all will be included in the Report Object generation.

15.4.2 Build Rules tab

A rule can be defined for all known measurements to specify the properties to be used for the corresponding Report Object.

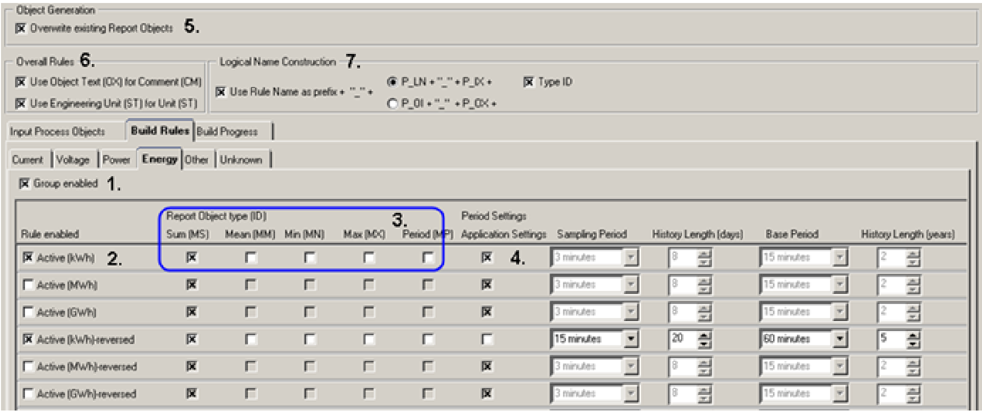


Figure 163: Report Object Generator Rules tab

Following rule settings can be made here:

1. Measurement Group can be enabled/disabled
2. Single measurement can be enabled/disabled
3. Which type of Report Objects should be created
4. Use the overall application report settings or specific ones
5. Specify if existing Report Objects will be overwritten or excluded
6. Overall rules:
    - Comment (CM) rule: If this option is enabled, the Report Object attribute CM will be constructed from the connected Process Object OX attribute. Otherwise, it will be left empty.
    - Unit (ST) rule: If this option is enabled, the Report Object attribute ST will be fetched from the connected Process Object ST attribute. Otherwise, it will be left empty.
7. Report Object name LN construction:
    - Use the rule name as prefix. If this option is enabled, the applied rule name will be used as a prefix. See Table 164 for details about rule names.
    - Use the LN_IX attribute or the validated OX_OI attribute combination from the connected Process Objects. Validated OI_OX means that white space and special characters that are not allowed will be replaced with the underscore character (_).
    - Use the Report Object Type ID as appendix. This option is recommended to be used if more than one Type ID is enabled for a measurement Process Object.

The rules will be saved when the tool is closed and restored when the tool is reopened.

Table 164: Process Object Rule name relation

Group name	Process Object			Rule name
	RX (28..30)	Index IX	Description (OX)	
Current	CUR	10	Current L1	CUR_10
		11	Current L2	CUR_11
		12	Current L3	CUR_12
		131	Current	CUR_131
	NCU	13	Neutral current I0	NCU_13
	DNC	14	Dir. Neutral current Ij	DNC_14
	MDC	15	Max.demand I15min	MDC_15
	RCR	54	Residual current I0	RCR_54
	CSQ	139	Positive, Negative, Zero Sequence current	CSQ_139
		140		CSQ_140
		141		CSQ_141
Voltage	VOL	16	Voltage U12	VOL_16
		17	Voltage U23	VOL_17
		18	Voltage U23	VOL_18
		50	Voltage U1	VOL_50
		51	Voltage U2	VOL_51
		52	Voltage U3	VOL_52
		53	Voltage U0	VOL_53
		132	Voltage	VOL_132
	RVL	19	Residual voltage U0	RVL_19
	VSQ	142	Positive, Negative, Zero Sequence voltage	VSQ_142
		143		VSQ_143
		144		VSQ_144
Power	APW	20	Active power P	APW_20
		133	Active Power	APW_133
	RPW	21	Reactive power Q	RPW_21
		134	Reactive Power	RPW_134
	APP	22	Apparent power S	APP_22
		135	Apparent Power	APP_135
	COS	23	Power factor cosj	COS_23
		136	Power factor	COS_136
		91	Power factor L1	COS_91
		92	Power factor L2	COS_92
		93	Power factor L3	COS_93
	PAP	73	Active power L1	PAP_73
		74	Active power L2	PAP_74
		75	Active power L3	PAP_75
	PRP	79	Reactive power L1	PRP_79
		80	Reactive power L2	PRP_80
		81	Reactive power L3	PRP_81
Table continues on next page				

Group name	Process Object			Rule name
	RX (28..30)	Index IX	Description (OX)	
Energy	AEN	38	Active Energy (kWh)	AEN_38
		39	Active Energy (MWh)	AEN_38
		40	Active Energy (GWh)	AEN_39
		41	Active Energy (kWh)-reversed	AEN_41
		42	Active Energy (MWh)-reversed	AEN_42
		43	Active Energy (GWh)-reversed	AEN_43
	REN	44	Reactive Energy (kvarh)	REN_44
		45	Reactive Energy (Mvarh)	REN_45
		46	Reactive Energy (Gvarh)	REN_46
		47	Reactive Energy (kvarh)-reversed	REN_47
		48	Reactive Energy (Mvarh)-reversed	REN_48
		49	Reactive Energy (Gvarh)-reversed	REN_49
Other	FRQ	24	Frequency	FRQ_24
		138		FRQ_138
	HDS	25	Harmonic distortion	DS_25
	TMP	26	Temperature	TMP_26
	IMP	137	Impedance	IMP_137
Unknown	PT=6		Digital Input	PT_6
	PT=9		Analog input	PT_9
	PT=13		Pulse Counter	PT_13

15.4.3 Build Progress tab

From this tab, the Report Object generation progress can be started by using the Function key F2 or by selecting **Start** from the **Tool** menu.

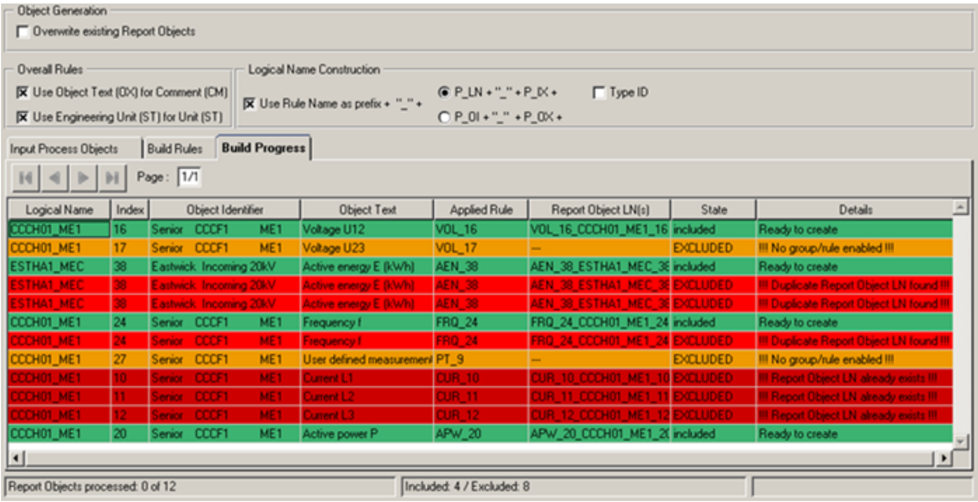


Figure 164: Report Object Generator Build progress tab

The table shows the main attributes from input Process Objects, the applied rule, the constructed Report Object LN and the build progress state for every object.

Table 165: Build progress state details

State	Details	Remarks
Included	Ready to create	
	Ready to modify	Report Object already exist and the overwrite option is enabled.
Excluded	No group/rule enabled	This measurement Process Object is excluded because either the corresponding group or the measurement item rule has been disabled.
	The applied rule has no Report Object type (ID) defined	Select a Report Object in the **Build rules** tab for the measurement.
	Duplicate Report Object LN found	Use the Type ID as appendix for the Report Object LN construction to avoid this duplication detection.
	Report Object LN already exists	Report Object already exist and the overwrite option is disabled.
	Report Object LN to long	The limit of 59 character length for the Report Object LN has been reached. Most likely, the P_OI_P_OX attribute combination is selected for the Report Object LN construction.
	Max Report Object count reached	The limit of 10 000 Report Objects has been reached.

15.5 Reports configuration database

All the Report configuration data will be stored in one SCIL database:

- General definitions and settings
- Report Object configuration data
- Report Display configuration data
- Report Page settings
- Runtime data
- Report Object Generator rules

Each of the items listed above have one section within the SCIL database. The database is stored under the application reports directory. With the first Measurement Reports configuration tool usage, the database will be generated from the default template located in <MicroSCADA installation drive>:\sc\sa_lib\defaults\reports. The data from this database will be copied to application specific system variables for fast access during runtime.

Table 166: Overview for Report configuration data storage

Data	Runtime storage	RCDB
Execution delays	APL:BSV32._SP(BP)Delay	Settings.Runtime
Execution Cycles	APL:BSV32.Sampling(Base)Periods	Settings.Runtime
Data storage definitions	APL:BSV32	Settings.Global
Report Object Data	APL:BSV36.OBJ_*	ReportObjectConf.Objects*
Report Object Groups	APL:BSV33.SP(BP)_xx (xx = sampling period respectively base period)	RuntimeData.ReportObject.Groups
Calculation Order	APL:BSV35.Order_SP(BP)	ReportObjectConf.Calculation
Report Display Configuration	APL:BSV37.Objects.OBJ_*	ReportDisplayConf.Objects
Report Page Configuration	APL:BSV38.Objects.OBJ_*	ReportPageConf.Objects
Internal	APL:BSV30, 31, 34 and 39	--

15.6 Report data logger

The Report Data logger is responsible for the following three main tasks:

1. Startup initialization. Depending on the length of time the system was down, the Data Objects for sampling and base period has to be filled for the down time range with not sampled values.
2. Collect and store the sampled Process Object values in Data Objects for the sampling period.
3. Collect and store the calculated Report Data values in Data Objects for the base period.

Below is a description of the sequential execution of the two main tasks for the sampling and base period.

15.6.1 Time channel BMU_CYCLE with 1 minute execution time

15.6.2 Sampling Period

1. Collect all Report Object names that belong to this sampling period cycle.
2. Execute the update instructions, ReportObject.P_ACT = 1.
3. Execute sampling period after sampling period execution delay.
4. Calculate Data Object Index for the actual time.
5. Handle the measured Report Objects (Type ID = MS, MN, MM, MX and NOT MP):
    - 5.1. Use the instruction attribute (IN) from the Data Object for the sampling period to evaluate the sampled value.
    - 5.2. Store the sampled value to the Data Object.
6. Handle the calculated Report Objects in defined sampling period calculation order:
    - 6.1. Collect the stored sampled values for all input Report Objects from DATAIN attribute.
    - 6.2. Execute the calculation operation from OPER attribute.
    - 6.3. Store the calculated value to the Data Object.
7. Execute the callback procedure for the sampling period.

15.6.3 Base period

1. Collect all Report Object names that belong to this base period cycle.
2. Execute the update instructions, ReportObject.P_ACT = 1 and Type ID = MP.
3. Base period execution after base period execution delay.
4. Calculate Data Object Index for the actual time.
5. Handle the measured Report Objects (Type ID = MS, MN, MM, MX and MP):
    - 5.1. If Type ID = MP, store the actual input Process Object value to the Data Object for the sampling period (one value only).
    - 5.2. Type ID <> MP: collect all sampled values from the last expired base period from the sampling period Data Object, e.g. for SP = 1 minute and BP = 15 minutes 15 sampled values to collect.
    - 5.3. Based on Type ID, calculate the base period value from the collected sampled values, Type ID =
      MS: calculate the sum value from the collected sampled values.
      MX: calculate the maximum value from the collected sampled values.
      MN: calculate the minimum value from the collected sampled values.
      MM: calculate the mean value from the collected sampled values.
      MP: collect the value from the Data Object for sampling period.
    - 5.4. Store the calculated value to the Data Object for the base period.
6. Handle the calculated Report Objects in defined base period calculation order:

   6.1. Collect the stored base period values for all input Report Objects from DATAIN attribute.
   6.2. Execute the calculation operation from OPER attribute.
   6.3. Store the calculated value to the Data Object.
7. Execute the callback procedure for the base period.

15.7 Report data viewer

The Report Data viewer consists of two main parts:

1. Report data provider
2. Monitor Pro displays

15.7.1 Data provider

The Report data to be shown in Monitor Pro Report displays will be provided by the SAGR_REPORTS_* command procedures.

User actions, such as edit values or add a comment, are also handled by the data provider.

All edit actions will be logged to the following file:

```
Reports/bmu_edit.log
```

All comments will be stored in separate files for every Report Display:

```
Reports/'Report Display LN'.mno
```

15.7.2 Monitor Pro Report Display configurations

The SYS600 Measurement Reports Display can be configured to use different colors for axes, curves, gridline and background. By using this functionality, the Measurement Reports Display can be adapted to display the Report data according to the convention required by the application.

All this configuration data can be stored either in an application specific or in user specific preconfiguration file. Unlike other display types, the application specific and the user specific preconfigurations are not accessible separately from the navigation menu. If a user specific preconfiguration exists, it will be used. If no such preconfiguration exsist but an application specific one does, that one will be used. In any other case, the default values will be used.

For more detailed information on settings for the graphical area, curve properties and legend control, see Section 14.

The time column format in the tabular view is defined by a parameter in section [REPORT_TIMECOLUMN] of the FrameWindow.ini configuration file.

Parameter name: ShowRangeInTimecol

Parameter value:

- 0: classic format, only period start time will be shown, e.g. 08:15 -
- 1: new format, period start time and end time will be shown, e.g. 08:15 - 08:30

The time format can also be changed in help of the "Time Range" parameter in the **General Settings** dialog.

15.8 Migration and Upgrade strategy

15.8.1 Upgrade from LIB510 based Measurement Reports

If the classic LIB510 Measurement Reports still needs to be used, no upgrade action should be performed.

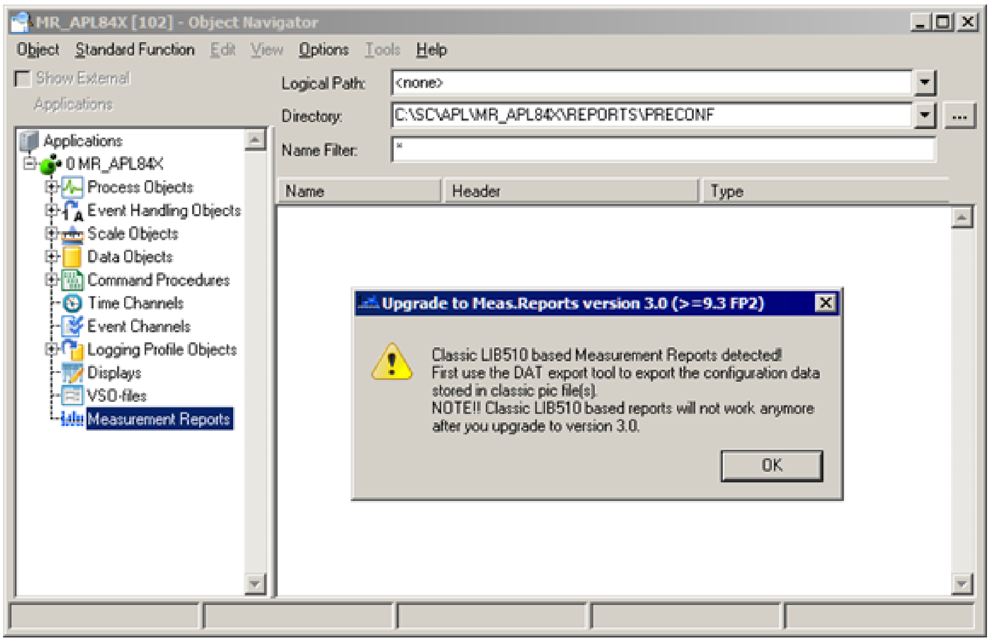


Figure 165: LIB510 Measurement Report detection

The upgrade for LIB510 based Measurement Reports needs to be done in two steps.

15.8.1.1 Step 1

DAT export tool selection from the Tool Manager.

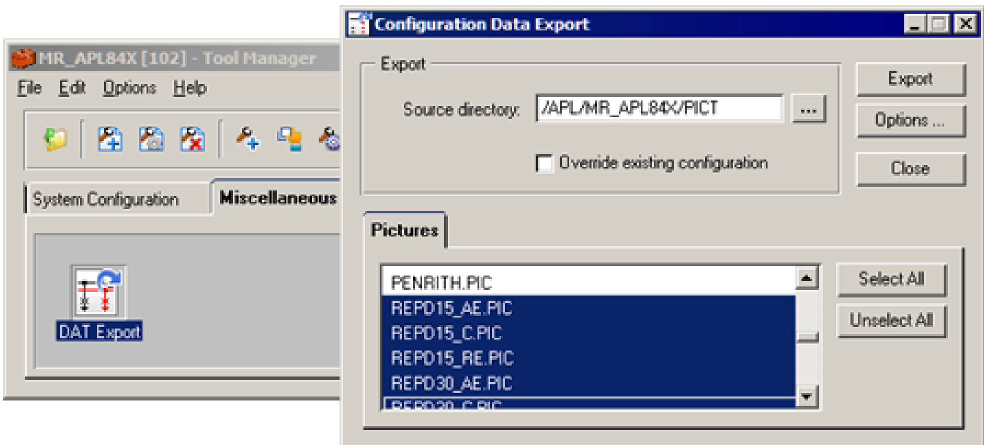


Figure 166: Dat Export tool for LIB510 Measurement Report upgrade

The Configuration Data Export tool locates the report configuration of the picture functions from the SCIL picture. The tool transfers the separated information to Measurement Reports related configuration files.

15.8.1.2 Step 2

See Section 15.8.2.

15.8.2 Upgrade from MicroSCADA Pro 9.x based Measurement Reports

The upgrade forthe Measurement Report build with SYS600 MicroSCADA X version will not be performed automatically.

The Report Object types EN (Entered data), GA (Gauge meter) are not supported anymore. They will be automatically converted to MP (Measured Period) type. Also user defined groups are not supported anymore.

The upgrade is as the other Reports engineering tool integrated in the Object Navigator.

An upgrade information message appears on the Measurement Reports node selection of the Object tree.

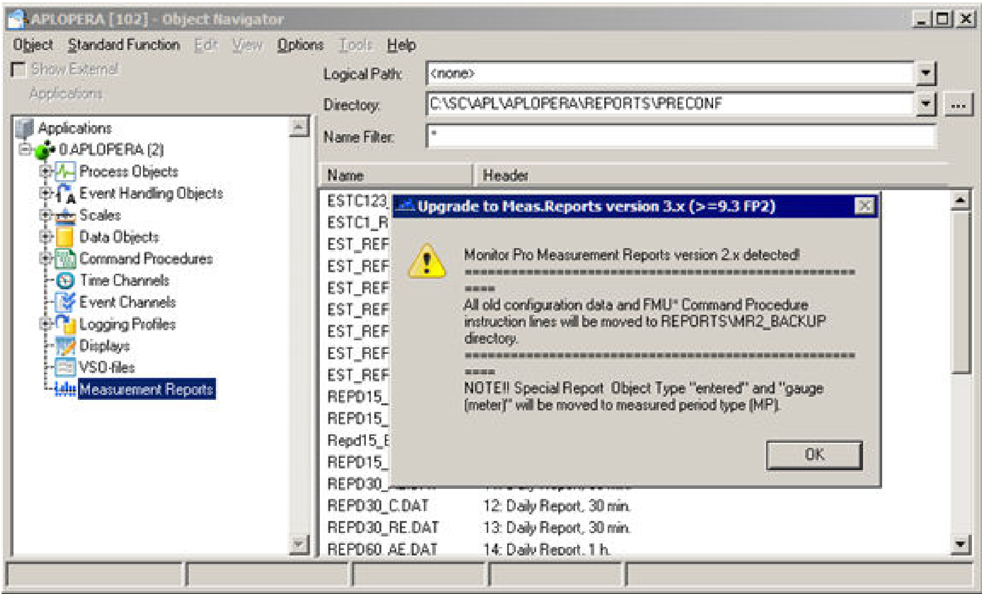


Figure 167: SYS600 Measurement Report upgrade

During the upgrade process, the following actions will be performed:

- All the old configuration data and settings from different locations will be collected, converted and stored in the Reports configuration database.
- New Data Logger BMU_* Application Objects will be installed.
- New Display Data Provider SAG_REPORTS_* Application Objects will be installed.
- The old runtime data structure will be cleaned, and the new one will be constructed, initialized and activated.
- All files, FMU_* Application Objects and configuration data used for old the Measurement Reports version will be moved to the application reports/MR2_backup directory.
- The old initialization procedure will be removed from APL_INIT_1(2,H):C if found.

If one of the following command procedures from LIB500 bbone package are used in some operation attribute they need to get replaced:

Table 167: LIB500 bbone Reports command procedure upgrade

LIB500 command procedure	Upgrade strategy
BGU_INDSF	Needs to be manually replaced with BMU_SP_IND command procedure call. Input parameter: 1. T1: Start time 2. T2: End Time 3. l_ReportObject: Report object configuration data. If running in the Report data sampling and calculation context (BMU_*:C) the Report Object data can be retrieved from the global variable %l_ReportObjects. Assume the Report Object name is "BAY01_CURL1" the SCIL command to get the Report Object configuration data is @l_ReportObject = %l_ReportObjects.OBJ_BAY01_CURL1
BGU_INDPF	Same as for BGU_INDSF
BGU_READSF	Will get automatically upgraded if found in application
BGU_READPF	Will get automatically upgraded if found in application
BGU_WRITSF	Same as for BGU_INDSF but in addition the Report Object data to be stored needs to be supplied in %v_OV variable. Make sure that the data vector length fits with the the selected T1-T2 time range.
BGU_WRITPF	Same as for BGU_WRITSF

15.9 Localization

All tool and dialog items, notification messages and object attribute texts can be translated using the Text Translation Tool.

1. Measurement Reports Visual SCIL tools integrated in Object Navigator can be found in the <MicroSCADA installation drive>:\sc\sTool\apl_build directory. The file name for all this tools start with OBJMR.
2. The Language text database <MicroSCADA installation drive>:\sc\Stool\AplBuild \OBJMR_LANG.SDB contains the Report Object, Report Display and Report Page attribute titles, help text and descriptions.

Section 16 Group alarm

The group alarm function can be used to summarize the alarm state of alarm points that can be freely selected. Groups can be defined, for example, for alarms from a specific location or alarms with the same functional background.

Group alarms can be further summarized and included as inputs to other groups of alarms in a tree structure. The group alarm state can be used to create an alarm overview display.

16.1 Group alarm database

All group alarm properties are stored in SYS600 SCIL Database (SDB). The group alarm database is located in sc/'application name'/apl_ and the database file name is APL_GRPAL.SDB.

Up to five sub levels can be defined for each group alarm. Each level can contain up to 999 members.

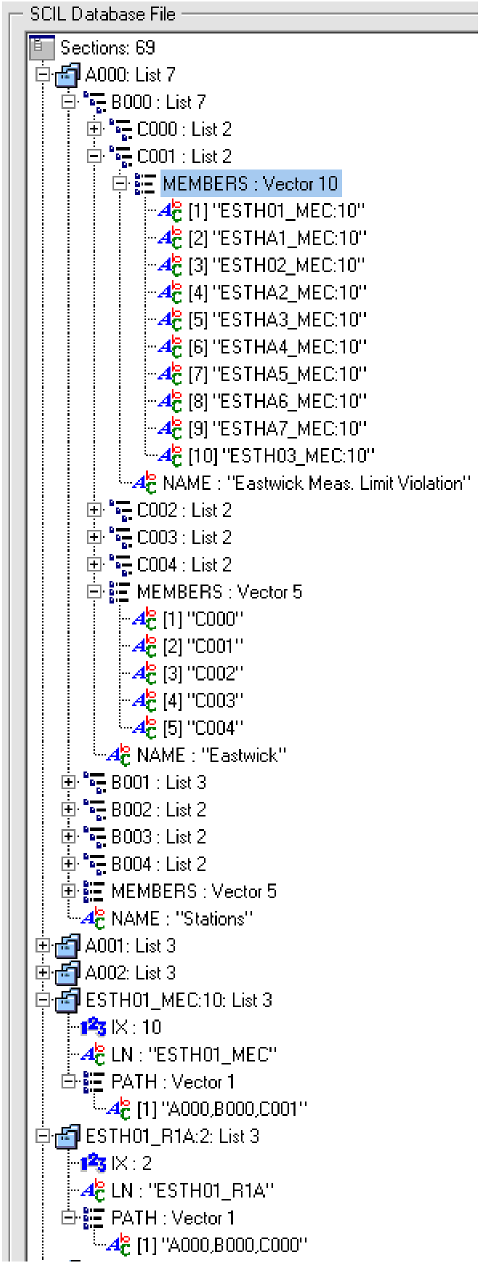


Figure 168: The group alarm database

The group alarm database is a copy of a template. The database is copied when the Group Alarm Editor is used for the first time.

The template database is located in sc\sa_lib\defaults\Group_Alarm, and the template database file name is bgu_grpal.sdb.

The group alarm database is located in sc\'application name'/apl_ and the database file name is apl_grpal.sdb.

The group alarm database contains three types of sections:

1. The group alarm structure.
2. The group member attributes.
3. The group alarm processing settings.

Do not manipulate the databases directly with SCIL. Use the dedicated tools for group alarms.

16.1.1 Group alarm structure

The section names in the group alarm structure always start with A000-A999. Each section contains the following attributes:

1. Name
   The name is used as an object text for the group state Process Object. The name can be changed in the Group Alarm Editor.
2. Array of member names
   The attributes contain the names of all group members.

16.1.2 Group member attributes

Attribute sections are used for the group members. Every process object acting as a member of an existing group gets its own section. The section name uses the group member logical name and the index.

Each section contains the following attributes:

1. LN: The logical name for the process object.
2. IX: The index for the process object.
3. PATH: Describes the location of the process object within the group alarm structure. The PATH variable is used to quickly locate the alarming object within the group structure and to calculate the new group alarm state.

16.1.3 Settings for group alarm processing

The Settings section has the following attributes:

1. AckChildren: Automatically acknowledges all child objects when group object is acknowledged.
2. AckGroup: Automatically acknowledges a group object when all child objects are acknowledged.
3. ResetWithAB: Resets the child object state within the group when the Alarm Blocking attribute is activated.

The settings can be changed in the Group Alarm Editor.

16.2 Process Object database

The group alarm status information is stored in a process object. The process object is created when a new group is created. The process object attributes can be modified with the Process Object Tool inside the Object Navigator.

16.3 Installation and initialization

The Group Alarm Editor is located in the **Application Objects** tab of the Tool Manager. The group alarm is initialized when the Group Alarm Editor is launched for the first time.

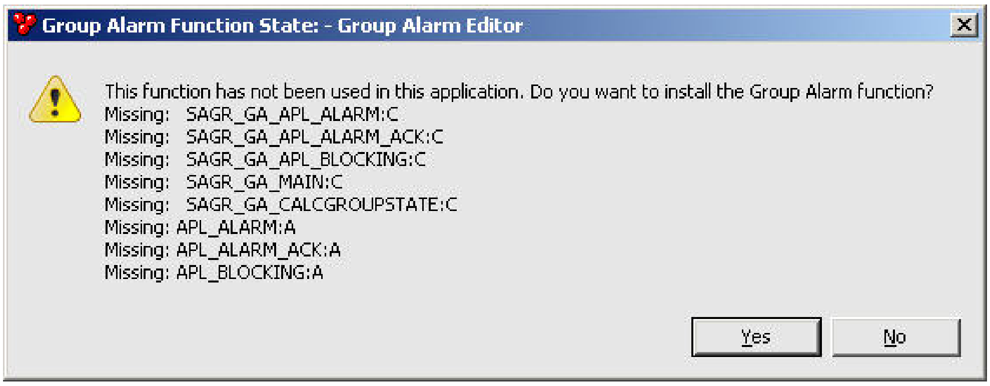


Figure 169: Group alarm initialization

16.4 Group Alarm Editor

The group alarm database and the grouping structure are created with the Group Alarm Editor.

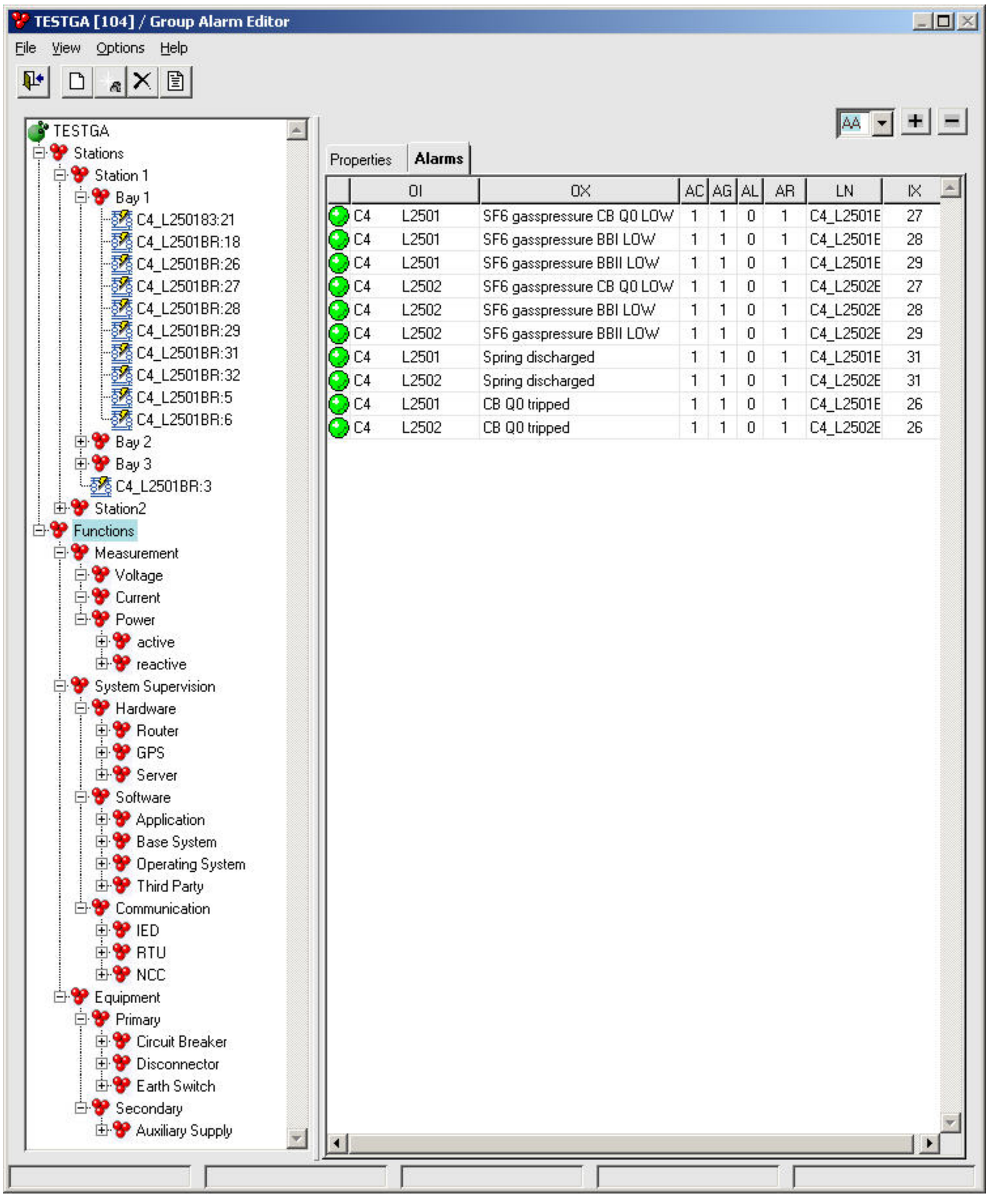


Figure 170: Group Alarm Editor

The Group Alarm Editor has three parts:

1. The group alarm structure as a tree view.
2. The **Properties** page for the selected object.
3. A list of members for the selected group.

16.4.1 Group alarm structure in tree view

The following commands are available in the toolbar to manage the group alarm database:

1.  : New Alarm
2.  : New Group
3.  : Delete Group
4.  : Edit Group name

The commands can also be selected by right-clicking the structure and selecting them from the context menu:

Figure 171: The group alarm context menu

16.4.1.1 Adding new group

New groups can be added anywhere in the existing structure. Click  or select **New Group** from the context menu. Enter the group name into the Input Dialog.

Figure 172: Input Dialog for adding a new group

There are no restrictions for the group name, except that the group names must be unique within a group branch.

16.4.1.2 Editing group name

Existing group names can be edited at any time. Renaming the groups does not affect the group alarm processing routine.

Figure 173: Input Dialog for editing a group name

16.4.1.3 Deleting groups

Alarm signals or groups can be deleted at any time. Deleting group objects also removes the linked group state process object.

Before deleting an alarm group, a confirmation dialog is displayed.

Figure 174: Confirmation dialog for removing a group

16.4.1.4 Adding new alarm

To add alarm signals to an existing group:

1. Click  or right-click the structure and select **New Alarm**.
2. The Select Process Object dialog opens. Select alarm objects from the Process Object structure and click **Add**.

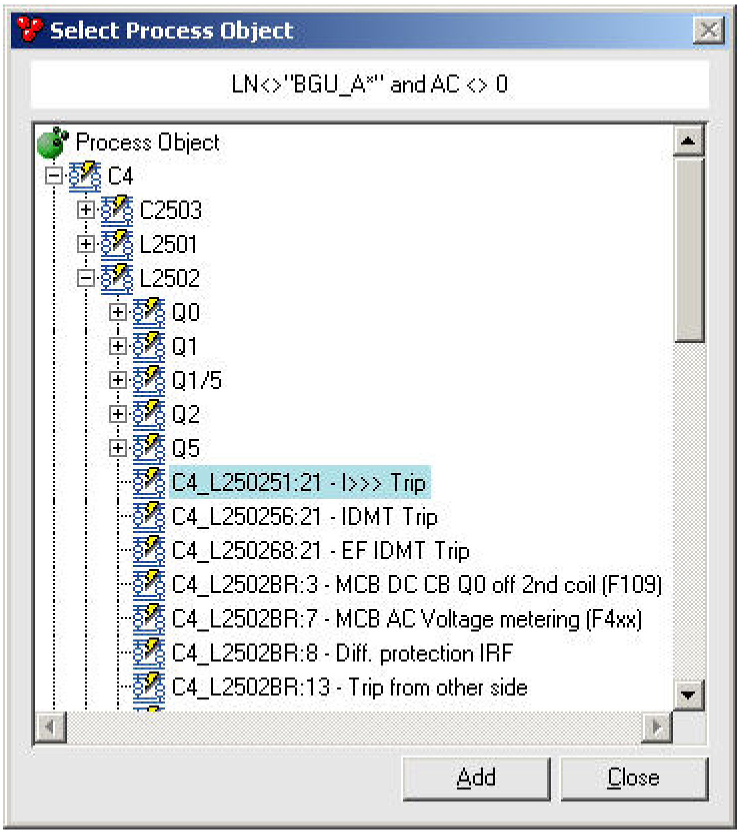


Figure 175: Process object browser for adding alarm signals

It is possible to hide process objects by adding a filter through the **Filter** dialog.

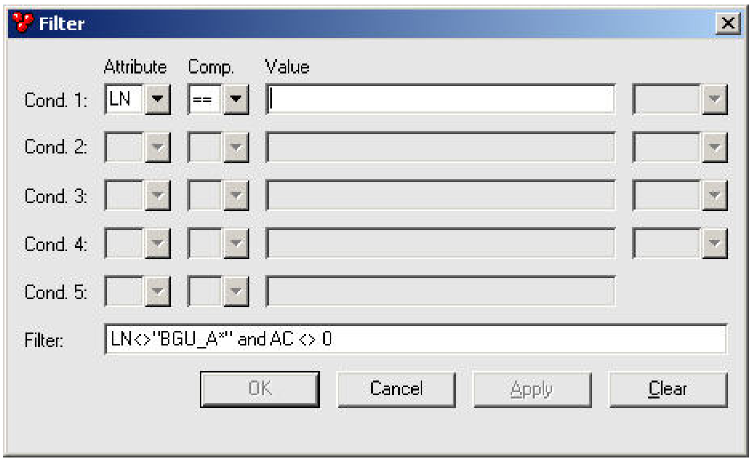


Figure 176: The object browser Filter dialog

Process object groups starting with BGU_A should be removed from the process object browser. These names are reserved for group alarm state objects. Also, Process Objects with Alarm Class equal to 0 cannot be assigned to a group.

When new alarm signals are added, the object browser can stay open. It is possible to select different groups in the group structure. New alarms are always added to the selected group.

16.4.2 Properties page for selected object

The appearance of the properties page depends on the selected object type.

16.4.2.1 Group alarm member object

The attributes for the process object can be modified in the Object Navigator or from this view by launching the **Process Object** dialog.

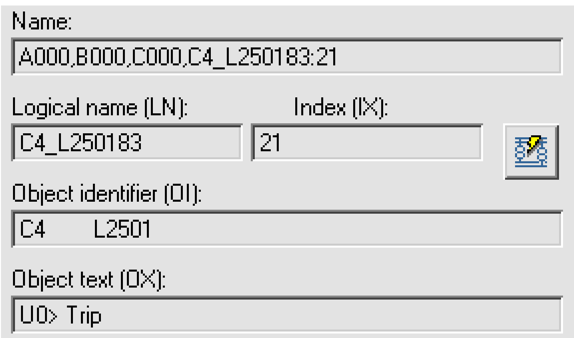


Figure 177: The Properties view for the group alarm member object

16.4.2.2 Group alarm state object

The attributes for the Process Object presenting the state of this group can be modified in the Object Navigator or by launching the Process Object form from the Properties view of the Group Alarm Editor.

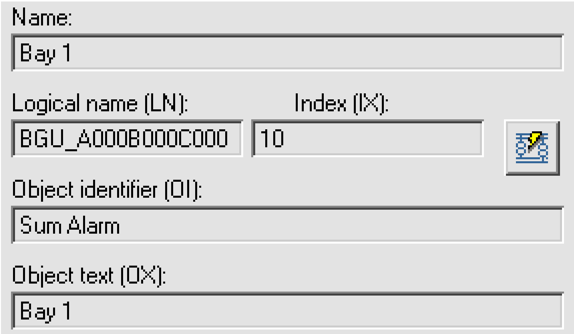


Figure 178: The Properties view for the group alarm state object

16.4.3 Member list for selected group

The **Alarms** page presents the state of all connected alarm signals for the selected group.

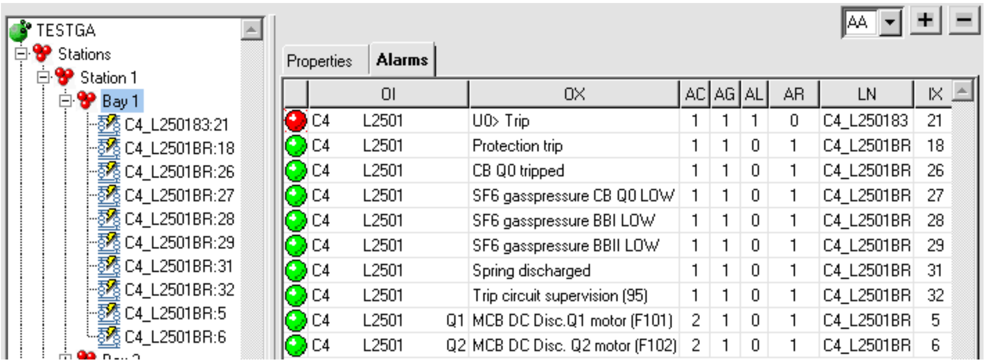


Figure 179: List of members for the selected group

As a default, the attributes OI, OX and the alarm state are shown in the Alarms view. The symbol for the alarm state for each signal can be seen in the first column. Additional attributes can be added by using the **+** and **-** buttons and by selecting the attribute from the list.

The **-** button removes the last column. Befault columns cannot be removed.

16.4.4 Optional group alarm settings

The following options are available for group alarm processing:

1. **Automatic Acknowledgement**
   If this option is active:
   1.1. All group members are automatically acknowledged with the group state object acknowledgement.
   1.2. The group state object is automatically acknowledged when all group members are acknowledged.
2. **Recalculate group alarm with Alarm Blocking**:
   2.1. If this option is active, the Alarm Blocking activation and deactivation on a group alarm member forces a new group state calculation, whereas the member state is excluded.

The options can be activated/deactivated in the **Options** menu of the Group Alarm Editor.

Figure 180: Group alarm settings

16.5 Group Alarm Viewer

The Group Alarm Editor can also be started in the Viewer mode, where controls for adding and removing the group members are hidden, and only members belonging to a dedicated group are shown.

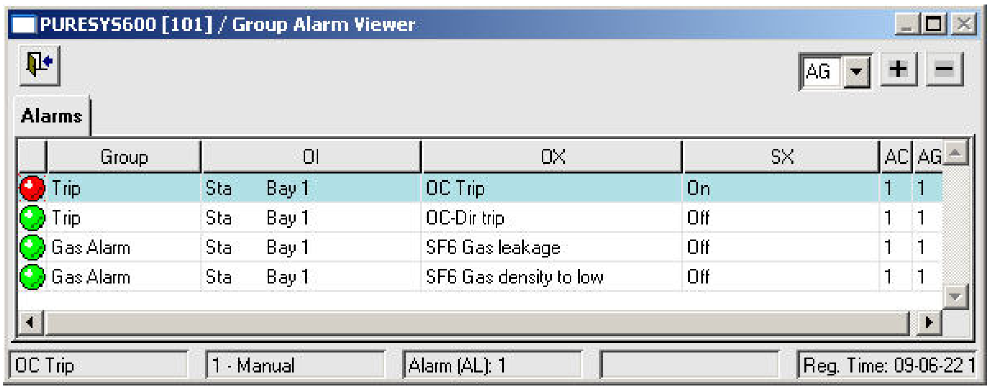


Figure 181: Group Alarm Viewer

The Viewer mode is started by supplying a list as a custom argument in Monitor Pro with an attribute called LN. This must be the Logical Name for the group state Process Object, e.g.:

LIST(LN="BGU_A000")

The Viewer mode can be opened from Monitor Pro in two ways:

1. Via a customized Menu item.
2. Connected to a display element.

The following dialog shows the settings that have to be made in order to open the Group Alarm Viewer via a new menu item.

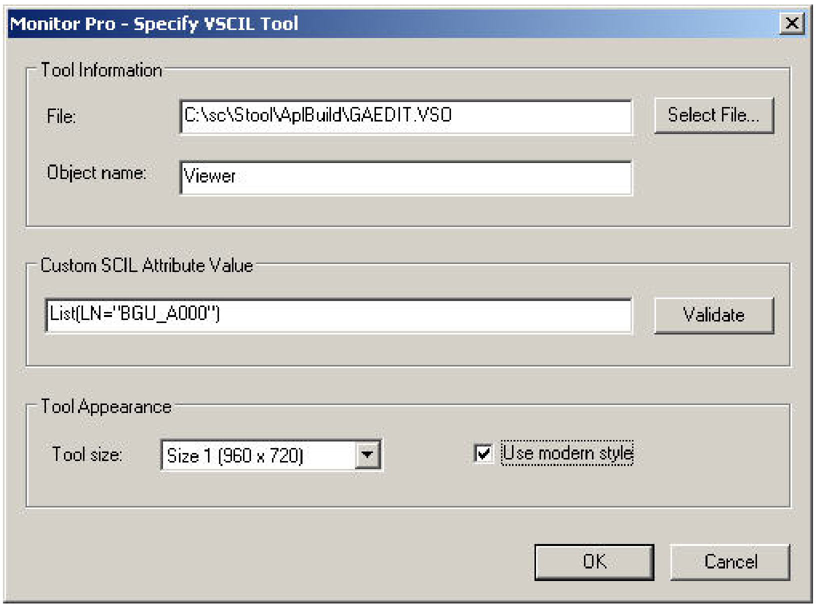


Figure 182: Group Alarm Viewer menu settings

The Group Alarm Viewer can also be launched by using the Display Builder Tool Launcher feature for any element in the display.

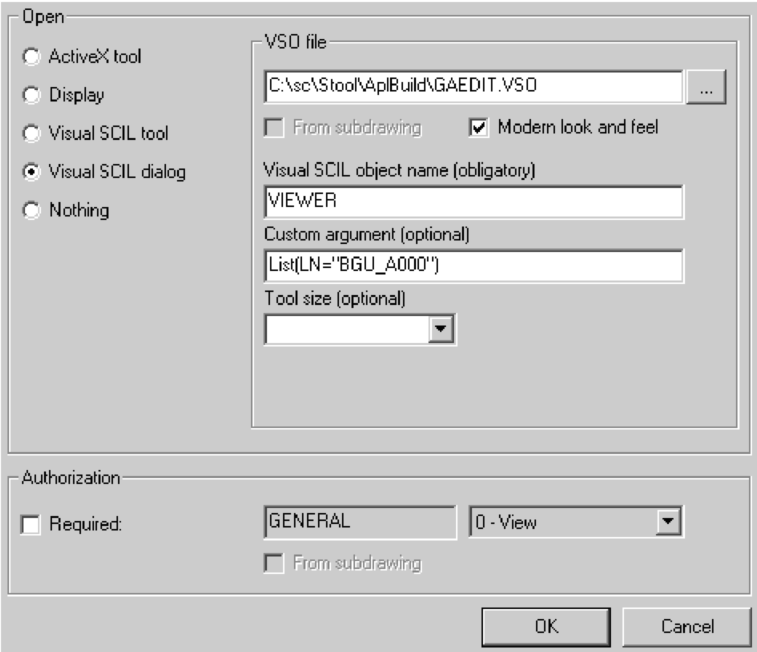


Figure 183: The Group Alarm Viewer Tool Launcher settings

16.6 Group alarm symbol

The following symbol can be used to show the group alarm state:

sc\prog\graphicsEngine\Palette\01 - SA_Common\Group Alarm Led.sd

Figure 184: Group alarm state symbol

To install the symbol, the group state process object must be selected with the Display Builder Object Browser.

The data variable Name is mapped to the object text (OX) attribute during symbol installation and used as a label in the symbol.

16.7 Group alarm processing

The pre-defined event channel's APL_ALARM, APL_ALARM_ACK and APL_BLOCKING must exist. If they do not exist, they will be created during the group alarm function initialization. The command procedures SAGR_GA_APL_ALARM, SAGR_GA_APL_ALARM_ACK and SAGR_GA_APL_BLOCKING are connected to the corresponding event channel.

16.8 Localization

The **Group Alarm Editor** dialog items and notification messages can be translated with the Translation Tool. To access the Translation Tool, select **Text Tool** in the **User Interface** tab of the Tool Manager.

16.9 Application engineering information

16.9.1 Group alarm files

The following table shows all group alarm function related files and their functionality.

Table 168: Group alarm function related files

File	Functionality	Path
BGU_GRP_AL.SDB	Group alarm database template	/SA_LIB/DEFAULTS/GROUP_ALARM
GaEdit.vso	Group Alarm Editor	/STOOL/APLBUILD
Group Alarm LED.sd	Group alarm symbol	/PROG/GRAPHICSENGINE/PALETTE/01 - SA_Common

16.9.2 Command procedures

When the Group Alarm Editor is used for the first time, the following command procedures are created.

Table 169: Group alarm function related command procedures

Name	Functionality
SAGR_GA_MAIN	Main Routine
SAGR_GA_APL_ALARM	Connected to pre-defined event channel APL_ALARM
SAGR_GA_APL_ALARM_ACK	Connected to pre-defined event channel APL_ALARM_ACK
SAGR_GA_APL_BLOCKING	Connected to pre-defined event channel APL_BLOCKING
SAGR_GA_CALCGROUPSTATE	Calculation for the group alarm state

16.9.3 Process objects

The following table shows the group alarm function related process object attributes.

Table 170: Process object attributes

Logical Name	Index	Obj. type	Process Object	Remarks	Group identifier
BGU_A*	10	BI	Group alarm name	Internal	BAEALAXALA

* LN Convention:

LN[1..4] always BGU_

LN[5..8] A000..A999 for first level groups

LN[9..12] B000..B999 for second level groups

LN[13..16] C000..C999 for third level groups

LN[17..20] D000..D999 for fourth level groups

LN[21..24] E000..D999 for fifth level groups

Section 17 Configuring Audible Alarm for Workplace X

Audible Alarms for Workplace X are based on severity that is calculated from the alarm state and the alarm class of the process object and the limit value warning/alarm information of measurement process object. When process object is in normal state the severity is 0. When non-measurmeent process object is in alarming state the severity is set to equal as alarm class. For measurement processe object, the the limit value warning state will have severity 1 and alarming state severity 2.

Alarm sounds for different severities can be configured with the Audible Alarms Settings tool that can be opened from the SYS600 Tool Launcher. With the tool you can select which if any alarm sound should be played in the Workplace X when alarm with certain severity is activated. There are number of pre-defined alarm sounds that can be selected and pre-listened in the tool. When changes are done, the opened Workplace X monitors must be re-opened or refreshed to activate the new settings.

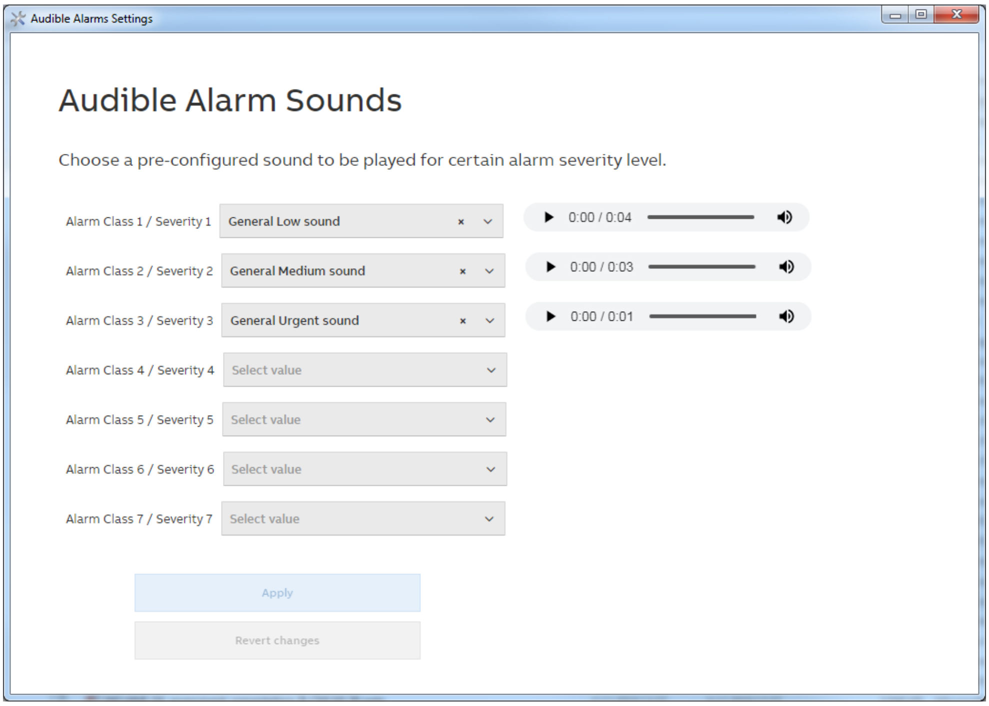


Figure 185: Audible Alarm Sounds

Section 18 Customizing application window

With Monitor Pro, the user can:

- Add custom toolbars
- Rename custom toolbars
- Delete custom toolbars
- Rename menu commands
- Delete menu commands
- Copy or move menu commands from one place to another
- Rename toolbar buttons
- Delete toolbar buttons
- Copy or move toolbar buttons from one place to another
- Configure a shortcut key for the login dialog

However, some menu commands and toolbars are generated programmatically and, therefore, they cannot be customized. For example, menu commands in the Process menu and the buttons in the application specific toolbars or application specific menus cannot be customized.

In addition, the user cannot add new menus, menu commands or toolbar buttons, but the existing ones can be copied and moved. Furthermore, pre-defined toolbars cannot be deleted, but the user can hide or show them.

Monitor Pro layouts are user specific. The default layout is loaded when the user logs on the application for the first time. Monitor Pro customization can be started in three different ways:

1. By double-clicking any empty area of the toolbar docking area.
2. By right-clicking any empty area of the toolbar docking area. From this menu, select the **Customize**.
3. By selecting **Settings/Customize**.

18.1 Layout adaptations

During logging, Monitor Pro asks if the user wants to take the new layout for menus, toolbars and icons into use. If old layout settings do not already exist, the new layout is taken directly into use. Icons are also updated to menus and toolbars.

If the user chooses to keep the old layout in use, the new icons can still be taken into use by using the Reset Icons functionality. To update the icons, select **Settings/Reset Icons...**.

18.2 Application window title

Monitor Pro application window title includes the display name and the possible preconfiguration name (for Process Display files, their name is shown without the file type identifier). If preconfiguration is not loaded, a No Preconfiguration text is shown.

The title format is:

{Display} – (preconfiguration) – Monitor Pro / {SYS600 application number} – {SYS600 application name} [User: {user name}].

The display can be one of the following values:

- Process Display
- Alarm Display Template 1
- Alarm Display Template 2
- Event Display
- Blocking Display
- Trends Display Graphical View
- Trends Display Tabular View
- Measurement Reports Display Graphical View
- Measurement Reports Display Tabular View

18.3 Customize dialog

In the **Customize** dialog, the user can:

- Add, rename, and delete custom toolbars (user defined toolbars). [1]
- Delete and rename menu commands [1] .
- Change the caption and style of the menu item or toolbar button.
- Get an action tool categorized listing.
- Drag and drop to any menu or toolbar.
- Create new customized menus, toolbars and menu commands.
- Access full keyboard shortcut customize dialog.

When the **Customize** dialog is open, the main menu and context menus can be displayed differently than at runtime. It is not possible to select any menus during customization.

The tool context menu can be used for customization, see Figure 186. The tool context menu can be displayed while the **Customize** dialog is open. It can be opened by right-clicking the toolbar button. The styles for this tool are the following:

- **Default Style**: includes both the text and icon, if available.
- **Text Only**: only the text (caption of the tool) is shown.
- **Image Only**: only the icon is displayed, if available.
- **Image and Text**: includes both the text and icon, if available.

[1] Some menu commands (for example, the menu commands in the **Process** menu, or items in application specific menus) and some toolbars (for example, the buttons in application specific toolbars) are not customizable because they are generated programmatically.

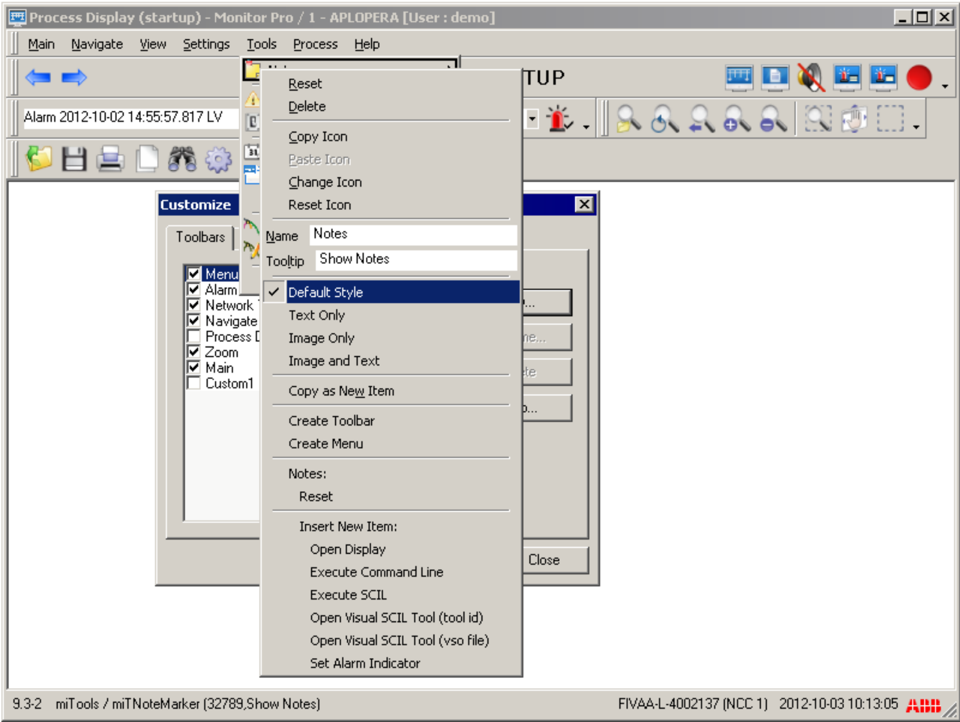


Figure 186: Tool context menu, selecting style

When the **Customize** dialog is open, toolbar buttons and menu commands can be moved around. Create a copy of the tool by pressing down the CTRL key while moving the tool.

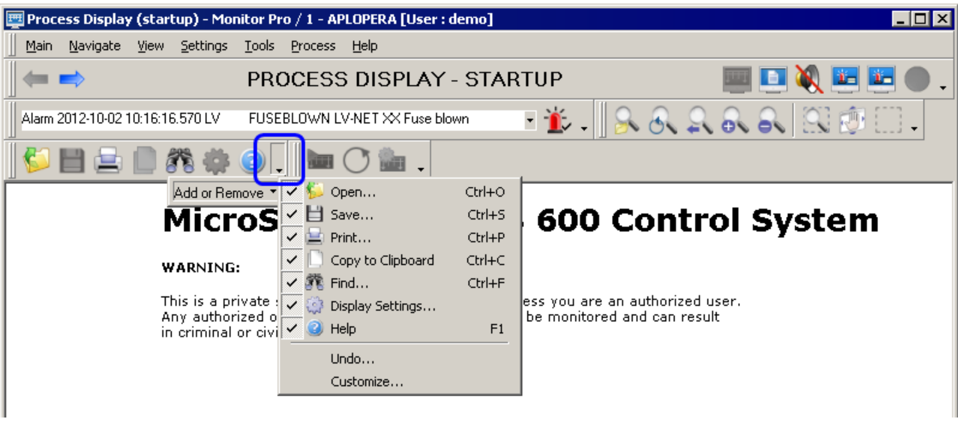


Figure 187: More Tools button

If the user is allowed to customize the toolbar, a **More tools** button is displayed, see Figure 187. This quick customization context menu allows the user to show or hide tools from the toolbar without opening the Customize dialog. However, menu commands must be customized by using the Customize dialog.

18.3.1 Toolbars tab

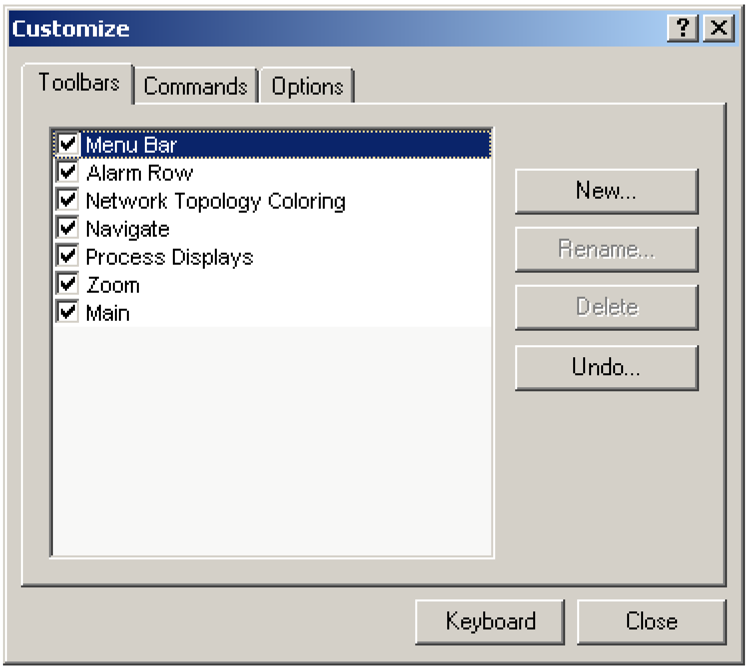


Figure 188: Toolbars tab in the Customize dialog

When the created toolbar is selected, the **Rename** and **Delete** buttons become active and the user can rename or delete the toolbar. By clicking **Revert** in the Toolbars tab, the last saved layout can be loaded. By clicking **Keyboard** in the Toolbars tab, a new dialog is displayed, see Figure 189.

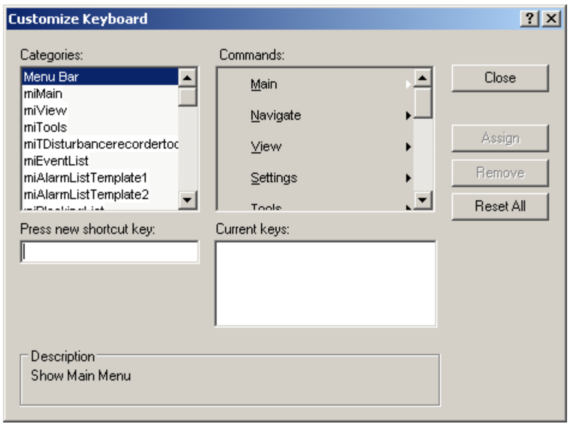


Figure 189: Keyboard shortcuts customization

In the **Categories** section, a new keyboard shortcut can be added to the menu commands and toolbars. The **Categories** list contains:

- Built-in Menus: all top-level menus
- Start with characters mi: all menus that have sub menu commands
- Start with characters tb: all toolbars (except user defined toolbars, i.e. menus created by using the **Customize** dialog)

The **Commands** list contains:

- Sub menu commands and buttons of a selected category

To give a new value for a tool (menu command or button in the toolbar), do the following:

1. In the **Categories** list, select the category in which the menu command or toolbar buttons are located.
2. In the **Commands** list, select the tool (menu command or toolbar buttons).
3. Type the new shortcut key combination in the text box Press new shortcut key.
4. Click **Assign** and the shortcut key appears in the Current keys field.

With **Remove**, the selected shortcut key in Current keys section can be removed. Reset the shortcut keys by clicking **Reset All**.

The user can configure the visibility of the toolbars and menus. Toolbar visibility and position are display specific: if, for example, the alarm row is moved to a certain position in Process Display, it will not affect the alarm row position in Event Display. For toolbar customization, use either the **Customize** dialog's **Toolbars** tab, or right-click the toolbar area and select or deselect toolbars. The user has access to different toolbars according to the display used. Table 171 lists the toolbars visible by default in each display.

Table 171: Toolbars according to different displays

Display	Toolbars
Process	Alarm Row Network Topology Coloring Navigate Process Displays Zoom Main Menu Bar
Alarm - Template 1	Alarm Row Alarm Display Template 1 Navigate Process Displays Main Menu Bar
- Template 2	Alarm Row Alarm Display Template 2 Navigate Process Displays Main Menu Bar
Event	Alarm Row Event Display Navigate Process Displays Main Menu Bar
Blocking	Alarm Row Blocking Display Navigate Process Displays Main Menu Bar
Trends - Graphical View	Alarm Row Trends Display Trends Graphical View Navigate Process Displays Zoom Main Menu Bar

Table continues on next page

Display	Toolbars
- Tabular View	Alarm Row Trends Display Trends Tabular View Navigate Process Displays Main Menu Bar
Measurement Reports - Graphical View	Alarm Row Measurement Reports Display Measurement Reports Graphical View Navigate Process Displays Zoom Main Menu Bar
- Tabular View	Alarm Row Measurement Reports Display Measurement Reports Tabular View Navigate Process Displays Main Menu Bar

Toolbar visibility is Display specific. For Process Display, Alarm Display, Event Display, Blocking Display, Trends Display and full screen mode, there are different configurations on which toolbars are shown. Change the visibility of the toolbar in a view by selecting the view and changing the visibility of toolbars. The configuration is automatically saved when some other view is selected.

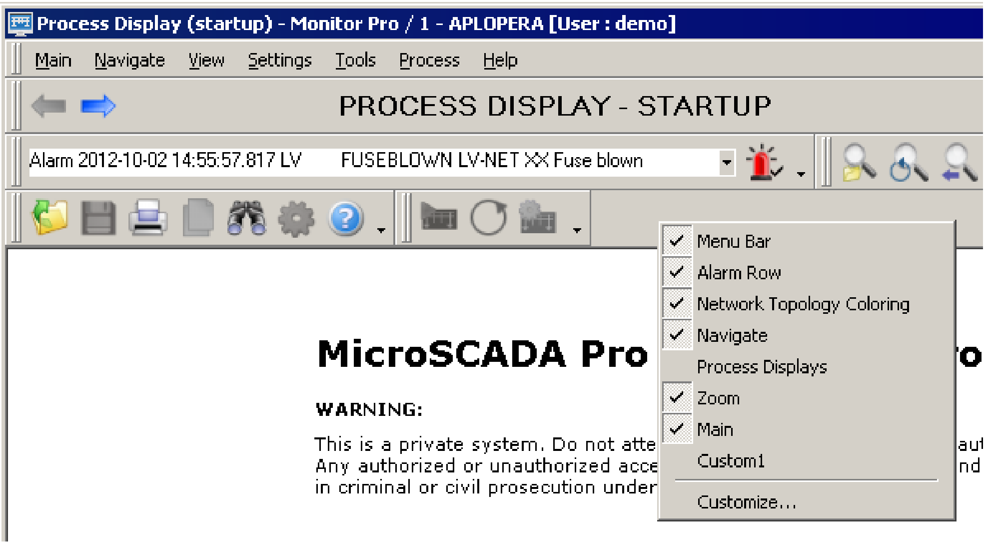


Figure 190: Selecting/deselecting toolbars by right-clicking on the toolbar area

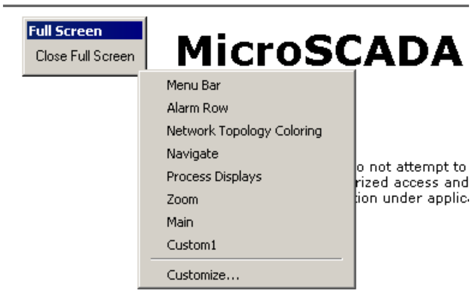


Figure 191: Toolbar menu in full screen mode

18.3.2 View Info field

The Navigate Toolbar display name is shown in the following format:

{Display} - preconfiguration.

For Process Display, the preconfiguration is a file name without the file type identifier. If the preconfiguration is not loaded, the No Preconfiguration text is shown.

The display can be one of the following values:

- Process Display
- Alarm Display Template 1
- Alarm Display Template 2
- Event Display
- Blocking Display
- Trends Display
- Measurement Reports Display

18.3.3 Commands tab

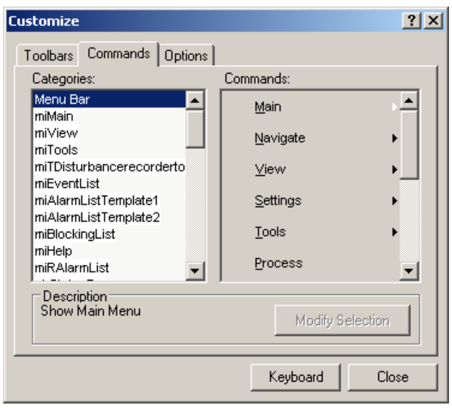


Figure 192: Commands tab in the Customize dialog

The Categories and Commands functions are the same as in assigning new keyboard shortcut keys, as described in Section 18.3.1.The tools can be moved around as described in the beginning of Section 18.3. When the **Modify Selection** button is clicked, a same kind of context menu appears as when a toolbar button or menu command is right-clicked while the Customize dialog is open. The **Modify Selection** button becomes active when a tool is selected either from the menu or from the toolbar.

18.3.4 Options tab

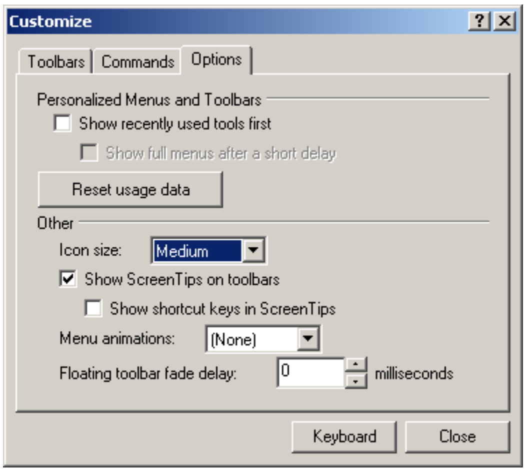


Figure 193: Options tab in the Customize dialog

The user can take the personalized menus into use via the **Options** tab. Only the most recently used menus are visible.

The data of menu usage is saved when the application is exited, and loaded again when the application is restarted (or the user logs into some application and user specific layout loading actions are done). Also, logging into the application loads the menu usage data in addition to the user specific layout file. Correspondingly, when logging out, the menu usage data is saved in addition to the user specific layout file.

In the Options tab, the user can select to use menu animations and the size of icons (Small, Medium or Large). The menu usage data can be deleted by clicking the **Reset usage data** button.

18.4 Adding audible alarm acknowledgement to toolbar

The **Audible alarm acknowledgement** button is enabled on the Monitor Pro toolbar when a related process object (ACK_SOUND:P1) is taken into use by setting the IU attribute value to 1. When this toolbar button is clicked, the command procedure ACK_SOUND:C is executed. This command procedure can be used to launch appropriate actions that are required in order to acknowledge an audible alarm. After changing the value of the IU attribute, logging out/in is necessary before the button becomes visible in the toolbar.

18.5 Adding new menus and toolbars

It is possible to create new Monitor Pro menus and menu commands in Customization Mode. Add menus by creating a new menu or by copying an existing menu.

18.5.1 Creating a new menu

To create a new menu:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a menu, for example **Help**.
3. Select **Create menu**, see Figure 194.

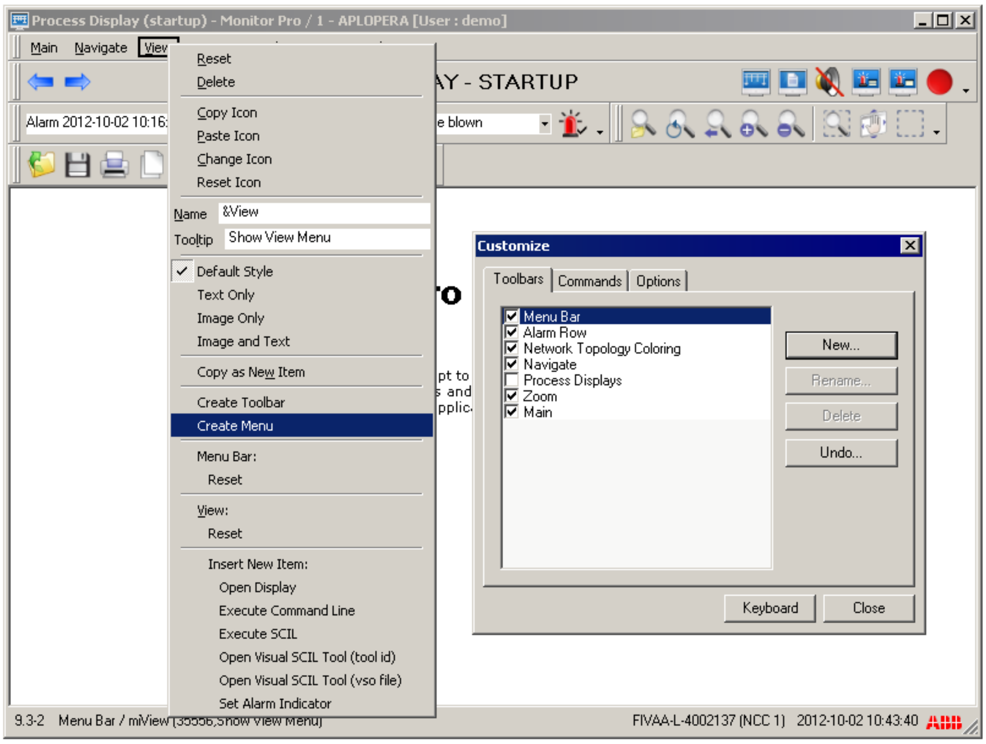


Figure 194: Creating a new menu

4. A new menu is created to the end of the menu bar.
5. Right-click a new menu and select **Name** to rename the menu with a descriptive name, see Figure 195.

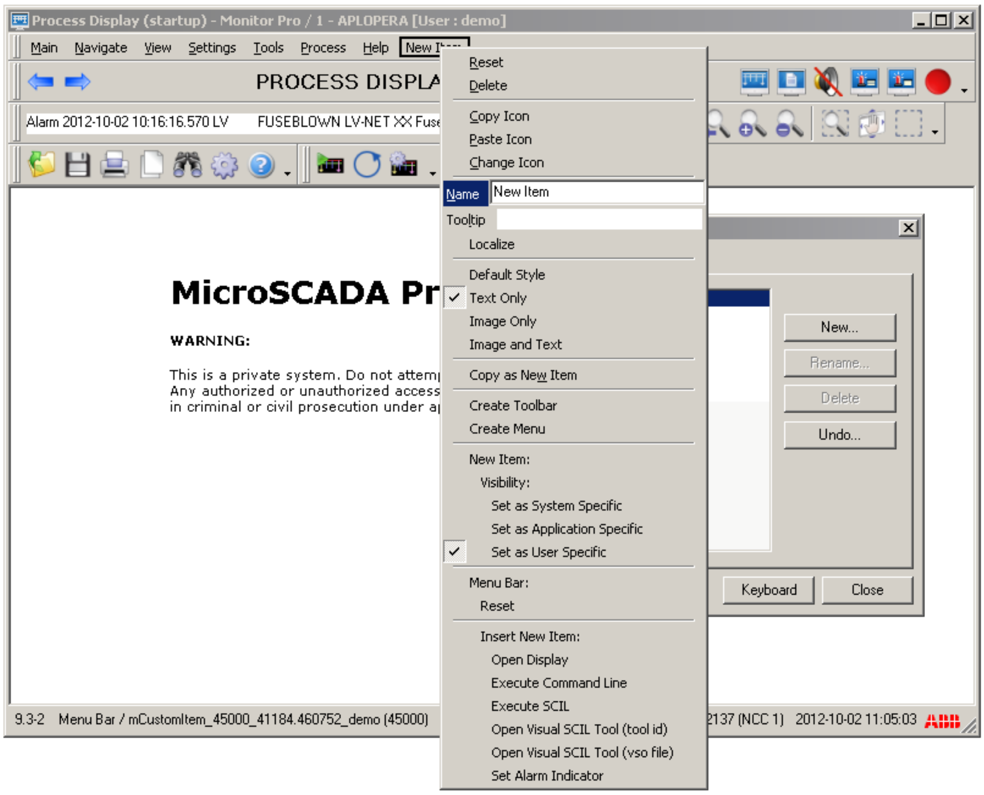


Figure 195: Renaming a new menu

6. Define a function for the menu. For more information about menu functions, see Section 18.14.

18.5.2 Creating a new menu by copying an existing menu

To create a new menu by copying an existing menu:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a menu, for example **Help**.
3. Select **Copy as New Item**, see Figure 196.

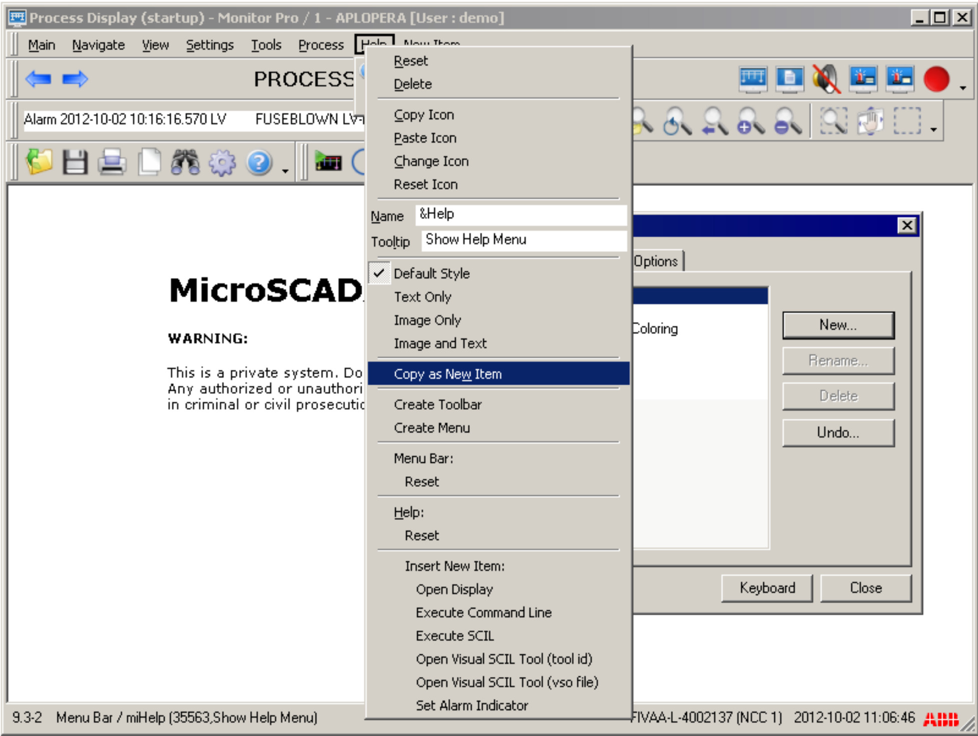


Figure 196: Creating a new menu by copying an existing menu

4. A new menu is created to the end of the menu bar.
5. Right-click the new menu and select **Name** to rename the menu with a descriptive name, see Figure 195.
6. Define a function for the menu. For more information about menu functions, see Section 18.14.

The **Copy as New Item** menu command is displayed on the customization context menu if the user has the authorization level 2 (Engineering) in the authorization group PRO_CONFIG_COMMANDS. If the current authorization group is not defined, the authorization group GENERAL is used. For more information about authorization, see Section 24.

18.5.3 Creating a submenu

Submenus can be created in the same way as a new menu, see Section 18.5.1. When the new menu has been created, select the menu item with the mouse and drag it to the menu where it is required, see Figure 194.

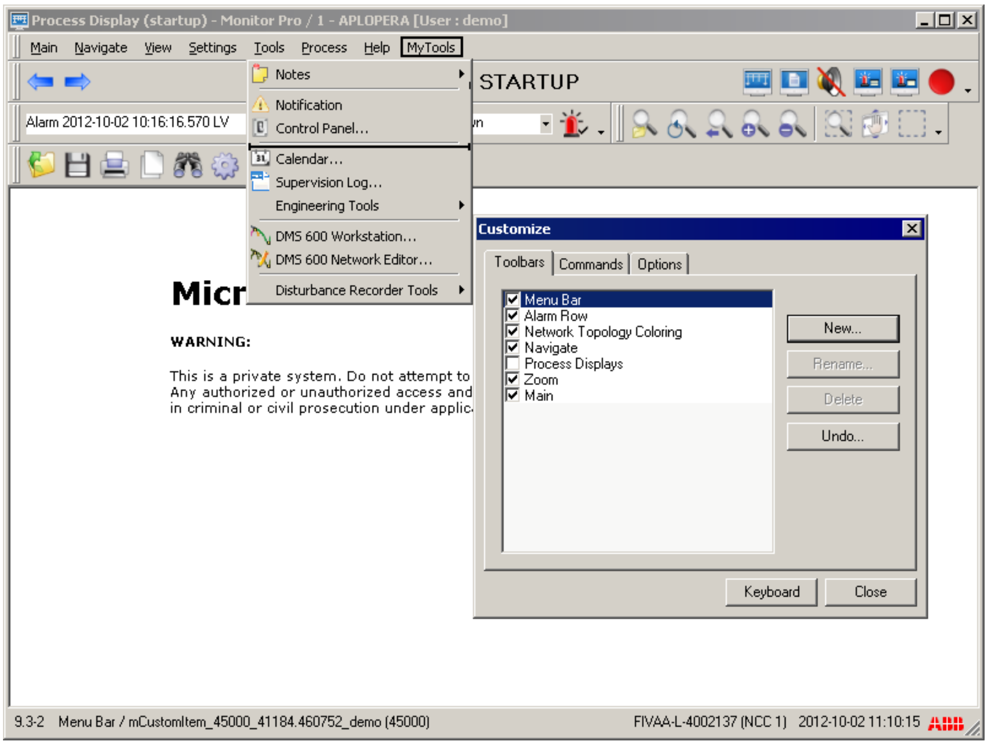


Figure 197: Creating a new submenu

18.5.4 Creating a new toolbar

To create a new toolbar:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a toolbar button or a menu, for example **Help**.
3. Select **Create toolbar**, see Figure 198.

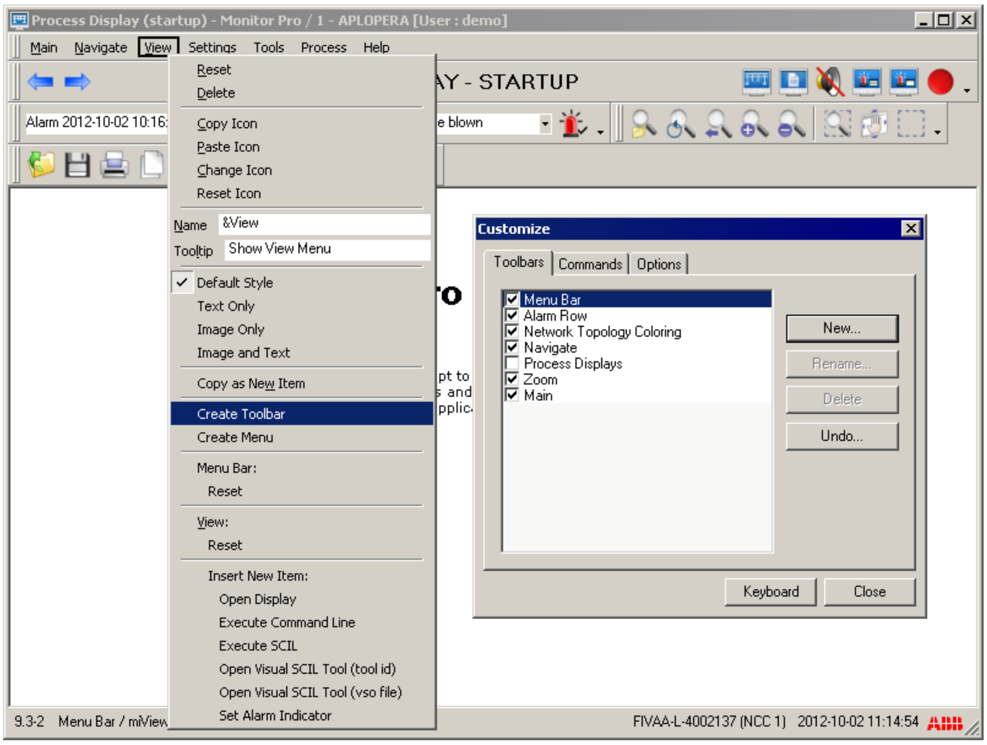


Figure 198: Creating a new toolbar

4. A new toolbar is displayed as a floating toolbar, Figure 199.

Figure 199: A floating toolbar

5. Right-click a new toolbar and select **Name** to rename the toolbar with a descriptive name.
6. Define a function for the toolbar. For more information about menu functions, see Section 18.18.

18.5.5 Creating a new toolbar in the Customize dialog

To create a new toolbar:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Click **New** in the Toolbars tab.
3. Name the new toolbar with a descriptive name in the **New Toolbar** dialog, see Figure 200.

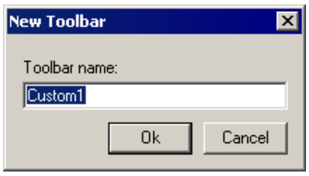


Figure 200: New Toolbar dialog

4. Save the toolbar name by clicking **OK**.

18.6 Customizing menu commands

Menus can be customized as follows:

- Add menu commands to the new menu by dragging a command from the **Commands** tab.
- Move the existing menu command from a menu to another.
- Copy the existing menu command to another menu.

18.6.1 Adding menu commands from Commands tab

Add a menu command by dragging it from the **Commands** tab to the new menu, see Figure 201. The new menu command acts as the original menu command.

The drag-and-drop operation using **Commands** tab creates a copy of the original menu command. The original menu command is located in its original place.

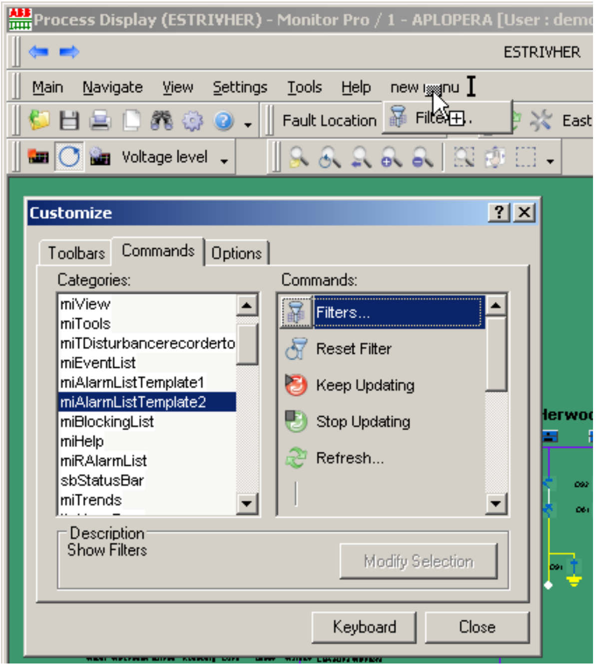


Figure 201: Copying menu commands to a new menu

18.6.2 Moving menu command between menus

Move a menu command by dragging it from the existing menu to the new menu, see Figure 202. A new menu command acts as the original menu command.

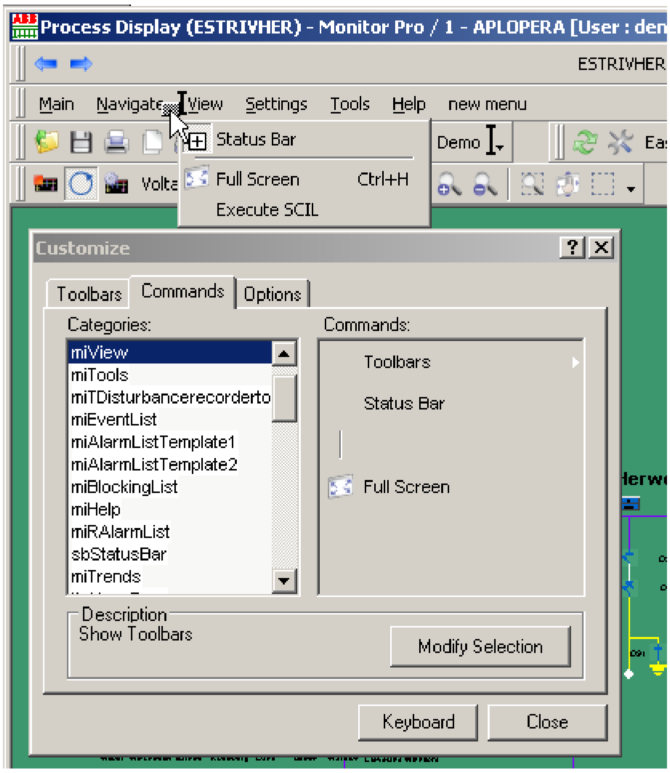


Figure 202: Moving menu command between menus

18.6.3 Copying existing menu command

To create a copy from an existing menu command:

1. Press CTRL.
2. Drag a menu command from the existing menu to the new menu, see Figure 202.

The new menu command acts as the original menu command.

18.7 Configuring process display context menus

The user can create context menus for the symbols of the process displays.

To open a context sensitive context menu in the process display and to run the selected menu command:

1. Right-click a symbol in the Process Display, see Figure 203.
2. Select a corresponding menu command to run the menu command.

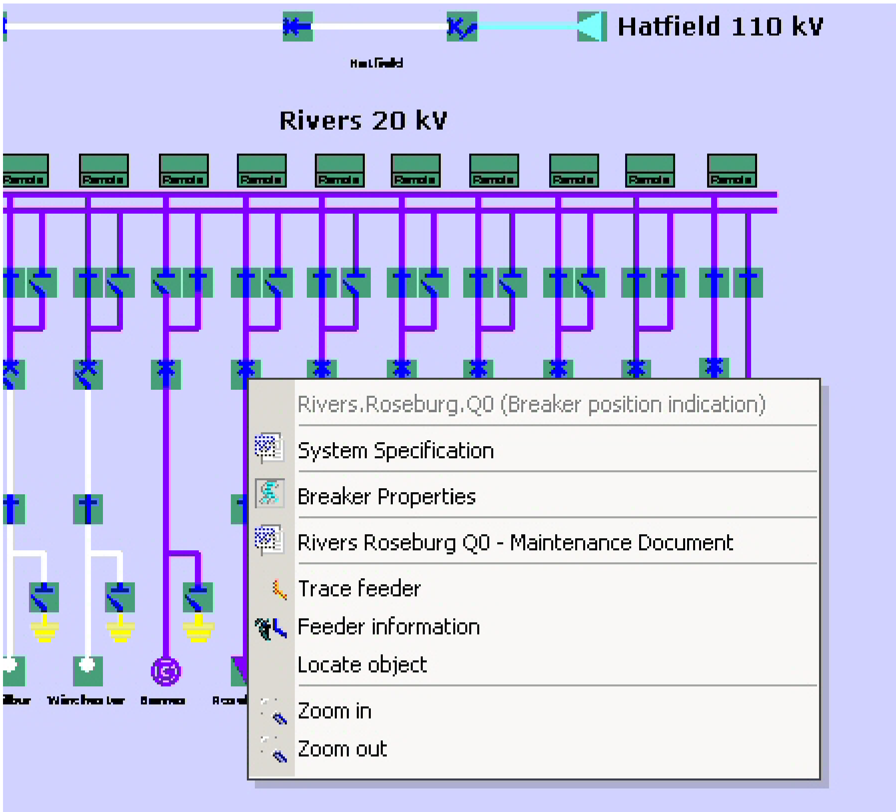


Figure 203: Context sensitive context menu

The menu structure is similar to the Windows **Start** menu. The menu commands are organized as folders and files in the file system. Therefore, no special tool is needed to configure the menu. The configuration is done by organizing the files, such as programs, documents or shortcuts and the directories in a file system.

A menu command can be, for example:

- a file
- a folder containing submenu commands
- a link

- a shortcut to a file
- a shortcut to a directory
- an Internet hyperlink

Object types and instances have separate directories, where files, such as technical documentation, Internet shortcuts or control commands (SCIL or VB scripts) exist. There is also a folder for the common menu commands. Table 177 shows the parameters that can be defined with a BAT, COM, SCIL, VBS or VSO file.

18.7.1 Language support

The menu commands have a national language support (NLS). Changing the language of the application window in run time affects the menu commands. However, this is not the case with the files, because the file name determines the name of the menu command. Thus, for each culture (country or language), there should be a folder where language specific files are located. Language specific menus can be built in these directories.

Use two-letter language codes when generating a menu structure. For more information about language codes, see http://www.w3.org/WAI/ER/IG/ert/iso639.htm.

18.7.2 Title, icon, and command

A title in the context menu item is used to give the operator additional information about the process object whose context menu was opened. For example, the title may be "Erador 20kV Station, Weathertop, Breaker 1". An icon and a caption of a menu command is, in fact, an icon that is associated to a file type. A caption is a file name without the file extension (.doc for example).

Disabled menu commands are grayed. The menu does not support check boxes, radio buttons or shortcut keys.

A file or a folder is an input for a menu command. Changing the name or the icon of the file changes the appearance of the menu command. If a file or a folder is hidden, it is not shown or the menu command is disabled. If a target of a link does not exist, the menu command is disabled.

The icon, name, and the file or folder attributes are read and a menu command is shown accordingly. If the file or the folder is a link or a shortcut, its' icon and name are used. Thus, the target of the link does not affect the appearance of then menu command.

The operator receives additional information about a process object by opening a context sensitive context menu. It contains the menu commands common to all process objects, i.e. menu commands that are process object type specific and instance specific. There can be menu commands that are shown only for a certain process object instance, such as a log, a figure, a video clip or a maintenance record.

When a menu command is clicked, a file is opened using the application that is associated with it in the operating system. A menu command may also be executable or a script. It is possible to give arguments to the executables or scripts.

Each application has its own directory in the file system, where the menu structure is constructed. The directory contains subdirectories, which determine the menu commands for:

- All process objects
- Process object types, for example breaker and transformer
- Process object instances

For more information about the menu structure, see Section 18.7.3.1.

18.7.3 Configuring context menu in Display Builder

When an object is right-clicked in Display Builder, three menu items are available for configuring context menus for objects. These items are:

- **Edit runtime menu for object**
- **Edit runtime menu for object type**
- **Edit runtime menu for all objects**

To edit the runtime menu for the current display, select **Actions/Edit runtime menu for current display...**.

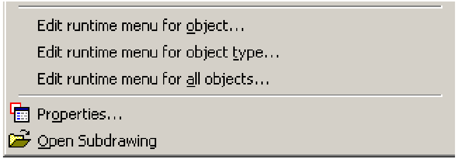


Figure 204: Display Builder context menus

When one of these menu items is selected, a dialog opens:

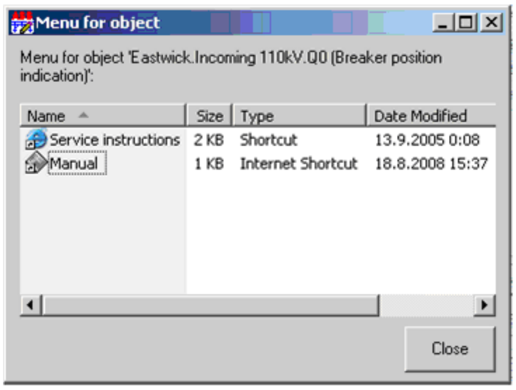


Figure 205: Dialog for configuring Display Builder context menus

In this dialog, the user can configure context menus. There are three different dialogs:

- In the **Menu for object** dialog, a context menu for this individual object can be defined.
- In the **Menu for object type** dialog, a context menu for this object type can be defined. Object type can be e.g. Breaker, Disconnector etc.
- In the **Menu for all objects** dialog, a context menu can be defined that is applied to all objects.

The user can define the context menus only for SA objects, i.e. objects that are displayed on sheets SA_* in Palette.

Save the display before the menu items are available.

The object must have a name before the user can define a runtime menu for it. The object name must contain only characters that are valid in directory name. If the user gives a name that contains characters that are invalid for directory name, the following error message is displayed:

Figure 206: Invalid object name error message

18.7.3.1 MENUS directory

The MENUS directory consists of directories such as all, objecttypes and instances.

- Menu commands, which are to be shown on each context menu, should be stored under the directory all.
- The directory objecttypes consists of all object types that a station overview can have. Menu commands that each object type can have are also located here.
- It is also possible that an instance of a breaker might have some menu commands that are instance specific, such as a figure of a breaker, online video stream, maintenance log or a web link to the manufacturer's home page. In that case, the menu commands are located in \<unique object id>.
- The menu for all objects are located in <apl>\MENUS\all.

18.7.4 Troubleshooting

Table 172: Possible problems with context-sensitive context menus

Description of the problem	Possible cause	Solution
Only a description of a context menu is displayed. A message "No items" or the icon indicates that the context menu is empty.	There are no commands in the menu directory. The menu is empty.	Check that the application has a menu structure, see Section 18.7.3.1.
The folder is displayed as a menu command and can be browsed. The folder does not contain submenu commands.		
The **Open With** dialog is displayed when running a menu command.	The file type of the menu command is unknown. The file has not been associated with any program.	Use the **Open With** dialog to select the program in which to open the file.
A menu command is disabled.	The user is not authorized to run the menu command or a link target does not exist.	Check the authorization level or the link target.
The custom object type does not have menu commands	A menu structure was not generated for the custom object type or the type does not have an SAObjectType attribute.	Add a menu structure manually for the custom object type and ensure that the object has an SAObjectType attribute.
An instance specific menu structure is not created for a symbol.	No view was defined when the menu structure was generated or the symbol's **Object Name** field is empty.	Use Display Builder to set the **Object Name** field for the symbol. Run the menu generator and use a viewpath argument to generate menus for instances.

18.8 Resetting layout, toolbars or localizations

These functions can be accessed from the **Settings** menu in the application window.

- **Reset Layout** resets the Monitor Pro layout. The old layout files are deleted and the new layout files are localized and taken into use.
- **Reset Toolbars** resets the toolbar positions. The user specific customizations are not deleted.
- **Reset Localizations** re-localizes the current Monitor Pro layout with the language used in the current session.

18.9 Process Display menu

A Process Display menu displays both the common parts for all process pictures and the specific parts for the currently active process picture.

The menu structure is similar to the Windows **Start** menu. The menu commands are organized as folders and files in the file system.Therefore, no special tool is needed to configure the menu. The configuration is done by organizing the files, such as programs, documents or shortcuts and the directories in a file system. For example, a menu command can be:

- a file
- a folder containing submenu commands
- a link
- a shortcut to a file
- a shortcut to a directory
- an Internet hyperlink

For more information about the national language support (NLS), see Section 18.7.1. For more information about the icons, see Section 18.7.2.

18.9.1 Process Display menu files

Common files for the process are located in [Appl path]\PICT\COMMON_MENU\. The files are displayed before the command separator in the **Process** menu.

Process Display specific files are separate files. For example, Process Display specific files for Example.v are in [Example]\MENU\, where the [Example] folder is the name of the Process Display file without the suffix v. The picture specific files are displayed after the command separator in **Process** Menu.

18.10 Configuring the Process Display search

The **Monitor Pro - Search** dialog is used for locating objects in the Process Display.

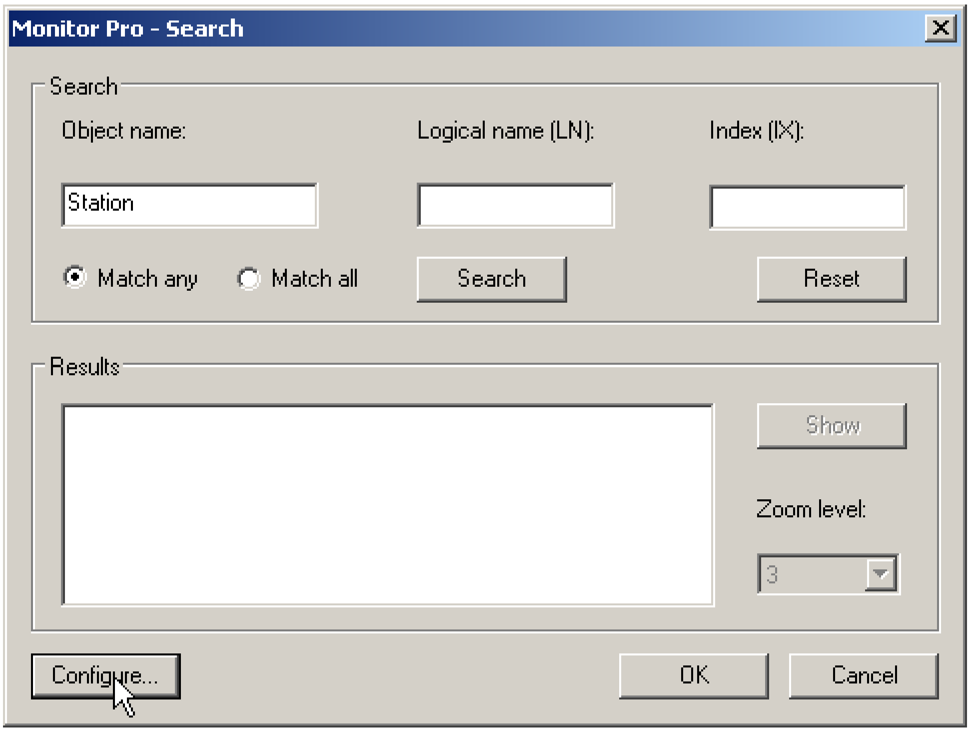


Figure 207: Process Display Search Dialog

It is possible to make more specific configurations to the dialog in the Monitor Pro - Customize Search dialog. The configurations are global, i.e. they affect all users. Configuring requires system administrator user rights. With the correct authorization, the **Configure...** button in the Search dialog is enabled.

To configure the Process Display search:

1. Open the **Monitor Pro - Search** dialog by clicking the  icon on the Process Display toolbar.
   The **Search** dialog opens.
2. Click the **Configure...** button, see Figure 207.
   The **Monitor Pro - Customize Search** dialog opens.

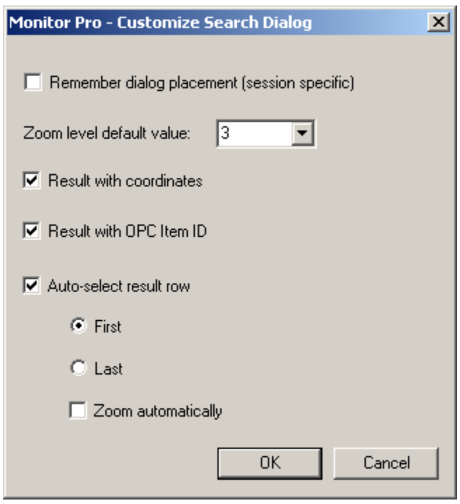


Figure 208: Customize Search Dialog

3. Set the configuration in the **Customize Search** dialog, see Figure 207. The following options are available:
   - **Remember dialog placement (session specific)**: When this box is checked, Monitor Pro opens the dialog in the same location during the whole session until logout.
   - **Zoom level default value**: Defines the zoom level used when the located object is selected from the result row and viewed in the Process Display.
   - **Result with coordinates** Checking this box makes the coordinates of the object visible on the Search dialog result row.
   - **Result with OPC Item ID**: Checking this box makes the OPC Item ID visible on the result row.
   - **Auto-select result row**: When this box checked and one of the **First**/**Last** radio buttons is selected, either the first or last result row is automatically selected in the Search dialog results. The **Zoom automatically** box can also be checked to make Monitor Pro automatically zoom on the selected result.
4. Click **OK** to take the configurations into use.

18.11 Opening Visual SCIL tools

The Visual SCIL tools can be opened by using a display element.

Add the following information to ExecuteItem[...].ini file in the menu folder, for example C:\sc \apl\AplOpera\menus\all\EN):

[OPENVSO]

VsoFileName=

VsoObjectName=

CustomData=

ModernLF=

ToolSize=

MenuItemCaption=

AuthorizationGroup=

AuthorizationLevel=

The caption shown in the context menu can be configured to the MenuItemCaption key. The icon used for the item comes from the file defined in the VsoFileName key.

The user can define several tools to be run from the context menu, one ini file corresponds one tool.

18.11.1 Using modal VSO dialogs with Monitor Pro

VSO dialog can be made modal by selecting the **Modal** option in VSO Dialog Editor. Modal dialog means that it is not possible to make any actions in Monitor Pro main window before the modal dialog is closed. Note that there are some limitations in support of modal VSO dialogs in Monitor Pro:

- If non-modal VSO dialog opens modal sub-dialog, other non-modal VSO dialogs that are open are not blocked. For example, when Tool Manager is open and another VSO dialog opens a modal sub-dialog, Tool Manager is not blocked.
- Setting a dialog to modal programmatically with SCIL is not supported.

18.11.2 Opening Visual SCIL dialog from Monitor Pro

1. Create a VSO file using Dialog Editor.
2. Define the launching of the dialog in Display Builder.

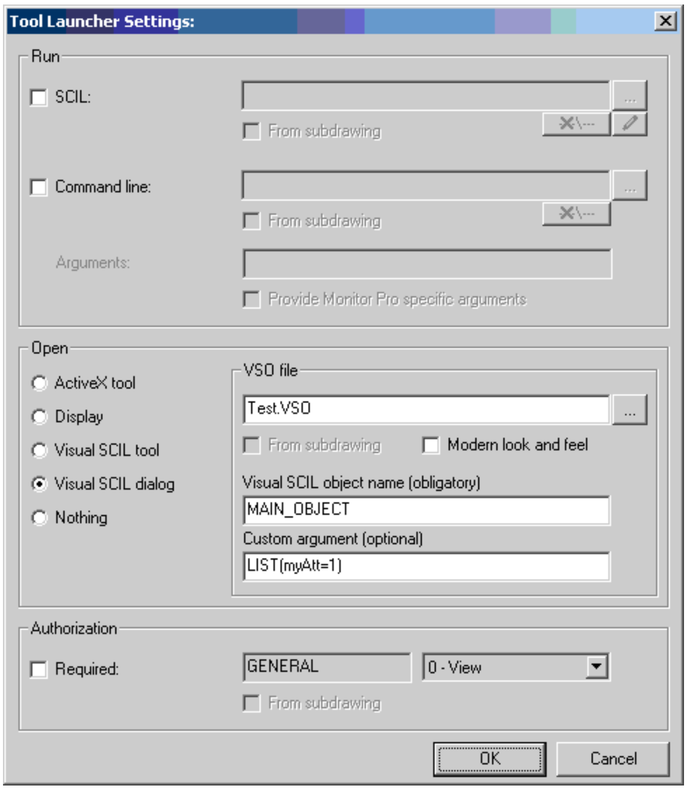


Figure 209: Tool Launcher dialog

18.11.2.1 Methods automatically executed when opening Visual SCIL dialog

The following list describes optional functions that can be defined in the **Visual SCIL** dialog for getting data from Monitor Pro to the custom **Visual SCIL** dialog. Define these methods as public methods in the **Visual SCIL** dialog. See Figure 210.

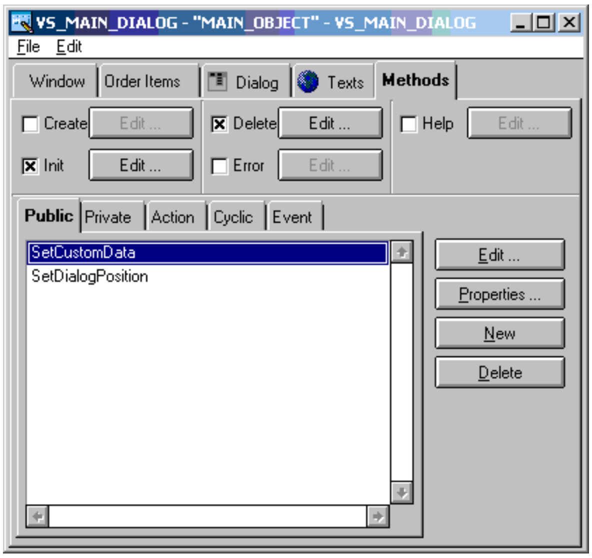


Figure 210: Methods for getting data from Monitor Pro

When opening the **Visual SCIL** dialog, methods are called in the following order:

1. INIT method of the dialog
2. GetAuthorizationGroup
3. SetAuthorizationLevel
4. SetDialogPosition
5. SetCustomData
6. SetDisplayInfo
7. SetIconInfo
8. SetFontInfo
9. SetColorsInfo

GetAuthorizationGroup

The GetAuthorizationGroup method is called by the underlying root object to get the authorization group from the VSO dialog. It will be applied when finding out the authorization level for the dialog. The method must return SCIL data type TEXT that defines the authorization group. For example, if the dialog should follow the MY_TOOLS authorization group, the implementation of the function would be:

#RETURN "MY_TOOLS"

SetAuthorizationLevel

The SetAuthorizationLevel method is called by the underlying root object to indicate the authorization level for the dialog. The function gets the authorization level as an INTEGER argument. For example, if the authorization group returned by function GetAuthorizationGroup is MY_TOOLS and the authorization level for this group in the User Management tool of MicroSCADA is 1, this function gets INTEGER value 1 as an argument. If the authorization group MY_TOOLS does not exist in the User Management tool of MicroSCADA, the authorization level of the GENERAL authorization group is passed as an argument into this function.

SetDialogPosition

SetDialogPosition method gets the following arguments that can be used when positioning the **Visual SCIL** dialog:

Argument(1) = The upper left corner of Frame window, LIST(X=value, Y=value)

Argument(2) = Screen coordinates of a mouse-click, LIST(X=value, Y=value)

Argument(3) = Frame window dimensions, LIST(TOP=value, LEFT=value, BOTTOM=value, RIGHT=value)

Argument(4) = Symbol dimensions, LIST(TOP=value, LEFT=value, BOTTOM=value, RIGHT=value)

where, value=screen coordinate

Below is an example of the method SetDialogPosition, which places the dialog on the right hand side of the symbol. If there is no room on the right hand side, the dialog is shown on the left hand side.

```
#local l_CursorPos = LIST(x=0,y=0) ;Coordinates of mouse click
#local l_SymbolDimensions=LIST(Left=0,Right=0,Top=0,Bottom=0); Coordinates
                of symbol
#local l_FrameWindowDimensions=LIST(Left=0,Right=0,Top=0,Bottom=0);
                Coordinates on Monitor
#local l_MyGeometry = this._geometry, x = 0, y = 50, i_Xoffset = 20,
                i_Yoffset = 75
#local i_SymbolMiddleY, i_SymbolMiddleX, i_MonitorMiddleX, v_MonBSZ =
                MON:BSZ, i_ScreenNumber = 1 ;MON:BSZ = Screen resolution
#if argument_count > 1 #then l_CursorPos = argument(2)
#if argument_count > 2 #then l_SymbolDimensions =
                argument(3)
#if argument_count > 3 #then l_FrameWindowDimensions =
                argument(4)
i_ScreenNumber = trunc(l_CursorPos.x / v_MonBSZ(1)) + 1
i_MonitorMiddleX = v_MonBSZ(1) / 2 + (i_ScreenNumber - 1) *
                v_MonBSZ(1)
i_SymbolMiddleX = l_SymbolDimensions.Left + ((l_SymbolDimensions.Right -
                l_SymbolDimensions.Left) / 2)
i_SymbolMiddleY = l_SymbolDimensions.Top + ((l_SymbolDimensions.Bottom -
                l_SymbolDimensions.Top) / 2)
;x coordinate
#if l_SymbolDimensions.Right + l_MyGeometry.w <
                l_FrameWindowDimensions.Right #then #block
x = l_SymbolDimensions.Right + i_Xoffset
#if x + l_MyGeometry.w > v_MonBSZ(1) * i_ScreenNumber #then x =
                l_SymbolDimensions.Left - l_MyGeometry.w - i_Xoffset
#block_end
#else #block
x = l_SymbolDimensions.Left - l_MyGeometry.w - i_Xoffset
#if x < 0 #then #block
#if l_CursorPos.x < i_MonitorMiddleX #then x = i_MonitorMiddleX +
                i_Xoffset
#else x = i_MonitorMiddleX - l_MyGeometry.w - i_Xoffset
#block_end
#block_end
;y coordinate
y = v_MonBSZ(2) - i_SymbolMiddleY - l_MyGeometry.h / 2
#if y < i_Yoffset #then y = i_Yoffset
#else_if y + l_MyGeometry.h > v_MonBSZ(2) - i_Yoffset #then y =
                v_MonBSZ(2) - l_MyGeometry.h - i_Yoffset
;set dialog position
.set this._geometry = list(x = x, y = y, w = l_MyGeometry.w, h =
                l_MyGeometry.h)
```

SetCustomData

The argument for SetCustomData function can be defined in the **Tool Launcher** dialog of the Display Builder, see Figure 211.

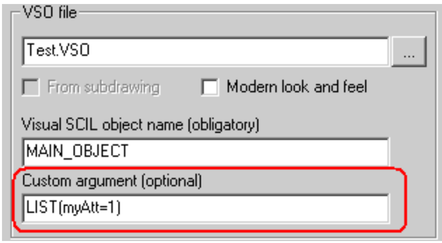


Figure 211: Custom data argument in Tool Launcher dialog of Display Builder

SetDisplayInfo

SetDisplayInfo method is called with argument type of list. This list contains various kinds of data that can be useful in dialog design, for example OPCITEMID. See Figure 212.

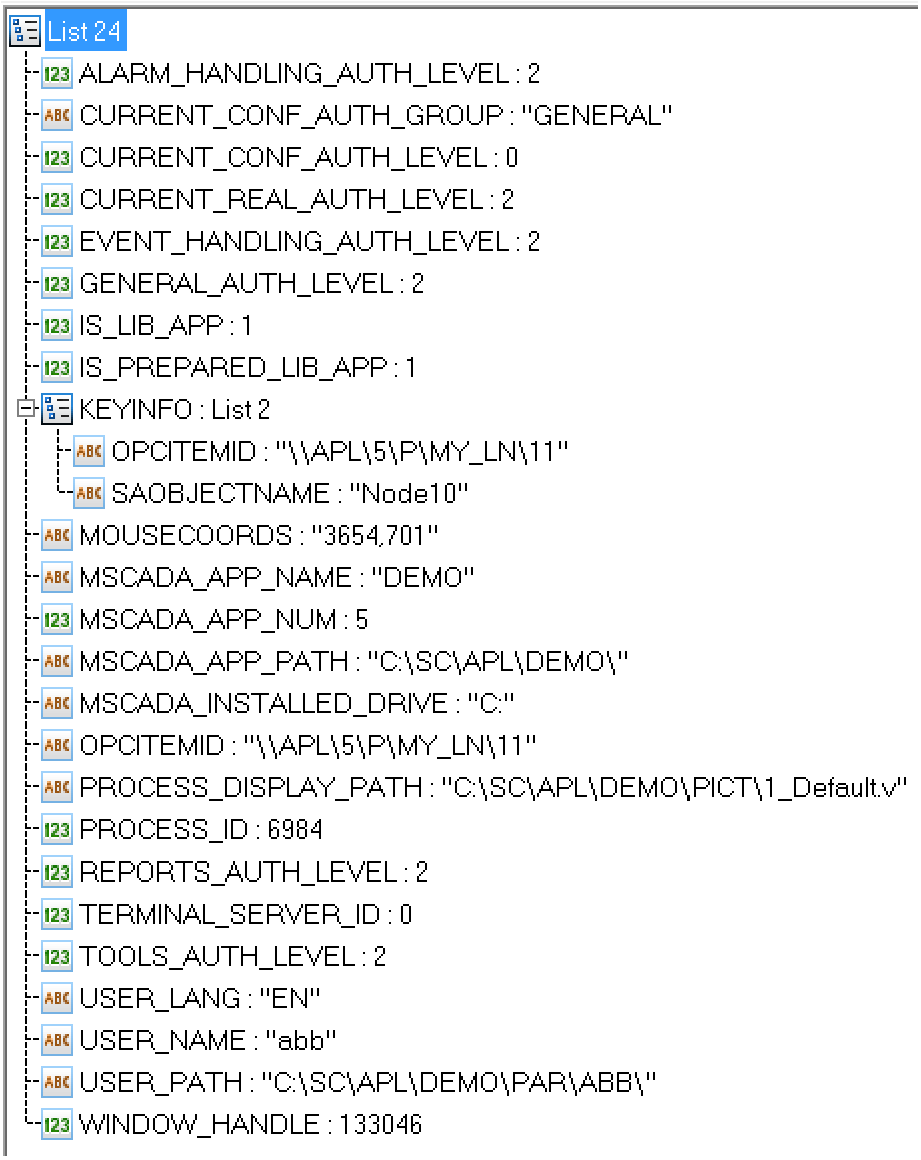


Figure 212: Data passed to method SetDisplayInfo

SendDisplayCommand

Use the SendDisplayCommand method for sending data to Monitor Pro. The method is available in an underlying, invisible **Visual SCIL** dialog that handles communication between Visual SCIL and Monitor Pro. This function is called in the parent object of the user's dialog.

There is a certain set of commands that can be passed to Monitor Pro. These commands are similar to those that can be passed to Monitor Pro as command line arguments.

The root.SendDisplayCommand(...) command does not work if the dialog is of VS_MAIN_DIALOG type. This is because the predefined object name "root" refers to the dialog itself. Use the name "parent" instead.

Examples

parent.SendDisplayCommand("test.v -ll:-1000,-1000 -ur:1000,1000")

parent.SendDisplayCommand("alarmlist_temp1 myPreconfiguration")

parent.SendDisplayCommand("LOCATE_OBJECT:OI=Rivers Barnes Q1;LN=RIVH04_Q1;IX=10")

The available commands are listed in Table 173.

Table 173: SendDisplayCommand

Command	Meaning
ALARMLIST_TEMP1 ALARMLIST_TEMP2	Alarm Display to be opened in an appropriate template. Preconfigured name can be given as an additional argument.
BLOCKINGLIST	Blocking Display to be opened.
EVENTLIST	Event Display to be opened (event list). Preconfiguration can be given as an additional argument.
REPORTS_GRAPHICAL REPORTS_TABULAR	Reports Display, (graphical or tabular) to be opened with an appropriate type. Preconfiguration can be given as an additional argument.
TRENDS_GRAPHICAL TRENDS_TABULAR	Trends Display to be opened with an appropriate type (graphical or tabular). Preconfiguration can be given as an additional argument.
<display name>.v	Process Display to be opened. Display argument may contain a path. Otherwise, the file is searched from the PICT directory. Process display can have the following additional arguments: -ll:x1,y1 -ur:x2,y2 This means lower left (x1,y1) and upper right coordinates (x2,y2) of Process Display area to be zoomed to. An -ll and -ur argument can also be given without a process display name. In this case, these arguments refer to the display itself. Examples: "test.v -ll:-1000,-1000 -ur:1000,1000" "-ll:-1000,-1000 -ur:1000,1000"
LOCATE_OBJECT:OI = <OI>; LN = <LN>; IX = <IX>	Locate object from opened Process Display. OI, LN and IX should be specified accordingly.

DisplayInfo

The DisplayInfo method is called by the underlying root object to notify the dialog about activity that happens in the Monitor Pro window. The function gets data describing the activity as a LIST argument. The argument LIST always contains three attributes: MESSAGE (name of message), MESSAGE_DATA (message data contents) and MSG_ID (identifier of the message). The function is called when:

Display changes (e.g. when user navigates to another process display)

Message name: DISPLAY_CHANGED

MESSAGE_DATA contents:

Attribute	Meaning
DISPLAY_ARG	Name of the display that was opened
DISPLAY_TYPE	Type of display that was opened. The type can be one of following values: PROCESS_GRAPHICS ALARMLIST_TEMP1 ALARMLIST_TEMP2 EVENTLIST BLOCKINGLIST TRENDS_GRAPHICAL TRENDS_TABULAR REPORTS_GRAPHICAL REPORTS_TABULAR

Example:

LIST(MESSAGE="DISPLAY_CHANGED",-

MESSAGE_DATA=LIST(DISPLAY_ARG="C:\SC\APL\MYAPP\PICT\S1.v",-

DISPLAY_TYPE="PROCESS_GRAPHICS"),MSG_ID=1)

Monitor Pro window is moved or resized on the screen

Message name: WINDOW_CHANGED

MESSAGE_DATA contents:

Attribute	Meaning
BOTTOMRIGHT_X	X-coordinate of bottom right corner of window client area in screen coordinates
BOTTOMRIGHT_Y	Y-coordinate of bottom right corner of window client area in screen coordinates
TOPLEFT_X	X-coordinate of top left corner of window client area in screen coordinates
TOPLEFT_Y	Y-coordinate of top left corner of window client area in screen coordinates

Example:

LIST(MESSAGE="WINDOW_CHANGED",MESSAGE_DATA=LIST(BOTTOMRIGHT_X=1302,-

BOTTOMRIGHT_Y=946,TOPLEFT_X=446,TOPLEFT_Y=258),MSG_ID=2)

Process display object is clicked

Message name: PD_OBJECT_CLICKED

MESSAGE_DATA contents:

Attribute	Meaning
MOUSE_X	X-coordinate where the mouse was clicked (in screen coordinates)
MOUSE_Y	Y-coordinate where the mouse was clicked (in screen coordinates)
OBJECT_NAME	Name of the object that was clicked
OBJECT_OPCITEM_ID	OPC item ID of the object that was clicked
OBJECT_TYPE	Object type

Example:

LIST(MESSAGE="PD_OBJECT_CLICKED",MESSAGE_DATA=LIST(MOUSE_X=717,MOUSE_Y=546,-

OBJECT_NAME="Node1",OBJECT_OPCITEM_ID="",OBJECT_TYPE="IED Generic"),MSG_ID=5)

Process display dialog is opened

Message name: PD_DIALOG_OPENED

MESSAGE_DATA contents:

Attribute	Meaning
MOUSE_X	X-coordinate where the mouse was clicked (in screen coordinates)
MOUSE_Y	Y-coordinate where the mouse was clicked (in screen coordinates)
CUSTATTR_KEY_NAME	Custom attribute name defining ProgID of the dialog to be opened
CUSTATTR_KEY_VALUE	Custom attribute value defining ProgID of the dialog to be opened.

Example:

LIST(MESSAGE="PD_DIALOG_OPENED",MESSAGE_DATA=LIST(MOUSE_X=200,-

MOUSE_Y=200,CUSTATTR_KEY_NAME="ToolLauncherProgId",CUSTATTR_KEY_VALUE=-

"ControlDlgs.Launcher"),MSG_ID=6)

Visual SCIL dialog is opened

Message name: VSCIL_TOOL_VSO_OPENED

MESSAGE_DATA contents:

Attribute	Meaning
MOUSE_X	X-coordinate where the mouse was clicked (in screen coordinates)
MOUSE_Y	Y-coordinate where the mouse was clicked (in screen coordinates)
VSCIL_VSO	Name of VSO file that was opened
VSCIL_OBJECT_NAME	Name of VSO object that was opened
VSCIL_STYLE	Windows style of the VSO dialog that was opened (Modern look&feel)
VSCIL_SGTS	Semigraphic tool size that has been set to VSO dialog

Example:

LIST(MESSAGE="VSCIL_TOOL_VSO_OPENED",MESSAGE_DATA=LIST(MOUSE_X=200,-

MOUSE_Y=200,VSCIL_VSO="c:\sc\apl\demo\pict\test.vso",VSCIL_OBJECT_NAME="MAIN",-

VSCIL_STYLE="1",VSCIL_SGTS="0"),MSG_ID=7)

Visual SCIL tool is opened with tool ID

Message name: VSCIL_TOOL_ID_OPENED

MESSAGE_DATA contents:

Table 174:

Attribute	Meaning
MOUSE_X	X-coordinate where the mouse was clicked (in screen coordinates)
MOUSE_Y	Y-coordinate where the mouse was clicked (in screen coordinates)
VSCIL_TOOLID	Visual SCIL Tool ID that was opened
VSCIL_SGTS	Semigraphic tool size that has been set to tool

Example:

LIST(MESSAGE="VSCIL_TOOL_ID_OPENED",MESSAGE_DATA=LIST(MOUSE_X=200,-

MOUSE_Y=200,VSCIL_TOOLID="USR_MGR",VSCIL_SGTS="0"),MSG_ID=8)

18.11.2.2 Methods for general Visual SCIL dialog design

There are several methods that can be utilized when making the **Visual SCIL** dialog design. These methods are called with the following syntax:

```
Parent.'method name'
```

The following methods are available:

GetColorsInfo
Returns a list of background, foreground and high light color values as defined in the framewindow.

Example of a return value:

```
LIST(-
BGCOLOR = LIST(B = 200, G = 208, R = 212),-
FGCOLOR = LIST(B = 0, G = 0, R = 0),-
HLCOLOR = LIST(B = 214, G = 194, R = 187))
```

GetCustomData
Returns the defined custom data argument similar to the SetCustomData. See Figure 211.

GetDisplayInfo
Returns the same displayinfo list that is also set by the DisplayInfo method. See Figure 212.

GetFontInfo
Font info that is used by the Monitor Pro.

Example of a return value:

```
LIST(-
CHARSET = 0,-
NAME = "MS Shell Dlg",-
SIZE = 8)
```

GetIconInfo
Icon size used in the Monitor Pro.

Example of a return value:

```
LIST(SIZE = "16x16")
```

GetLanguage
Configured user's language, for example EN.

GetWorkplaceNumber
Workplace number, for example 1.

GetUserName
User name, for example DEMO.

GetMonitorNumber
Monitor number, for example 101.

GetClickPosition
Cursor position of the mouse click which opened the **Visual SCIL** dialog. This information is not updated afterwards, so it is mainly useful for initial placement only.

Example of a return value:

```
LIST(-
X = 1648,-
Y = 583)
```

GetFrameWindowPosition
Current Monitor Pro window position. This is updated if the window is moved.

Example of a return value:

```
LIST(-
X = 1490,-
Y = 125)
```

GetDialogPositionHint
Recommended placement for the dialog. If the dialog does not handle the SetDialogPosition method call, this is the offset where the dialog will be placed when opened.

Example of a return value:

```
LIST(-
X = 1590,-
Y = 225 )
```

GetFrameWindowDimensions
Dimensions of the frame window. This information is updated when the frame window size and position changes.

Example of a return value:

```
LIST(-
BOTTOM = 941,-
LEFT = 1490,-
RIGHT = 2537,-
TOP = 125)
```

GetSymbolDimensions
Symbol size and position at the time of the click.

Example of a return value:

```
LIST(-
BOTTOM = 598,-
LEFT = 1633,-
RIGHT = 1657,-
TOP = 576)
```

GetColorScheme
Returns the color scheme file currently in use.

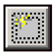
Example of a return value:

```
LIST(FILE =
"C:\sc\apl\demo\PAR\APL\ColorSchemes\SYS600_colors_grey.clut")
```

18.11.3 Creating subdrawing for Visual SCIL tool

1. Create a subdrawing, for example copy the existing IED.sd file.
2. To define the custom key, first open the subdrawing in Display Builder.
    2.1. Right-click the subdrawing background and select **Properties**.
    2.2. Select the **Custom** tab and define the custom key by double-clicking the key name on the line.
    2.3. Type the appropriate custom key name to the line.

18.11.4 Taking subdrawing into use

1. Create a new Process Display and import the new subdrawing.
2. Select the object that presents the subdrawing.
3. Add the OpenVso custom key for the object.
    3.1. Right-click the object and select **Properties**.
    3.2. Click the **Custom** tab and define the custom key.
    3.3. Create a new custom key by clicking the  button and by typing the key name on the line.
    3.4. Type OpenVso on the line.
    3.5. Define the VSO tool ID as a key value, for example EXAMPLE, see

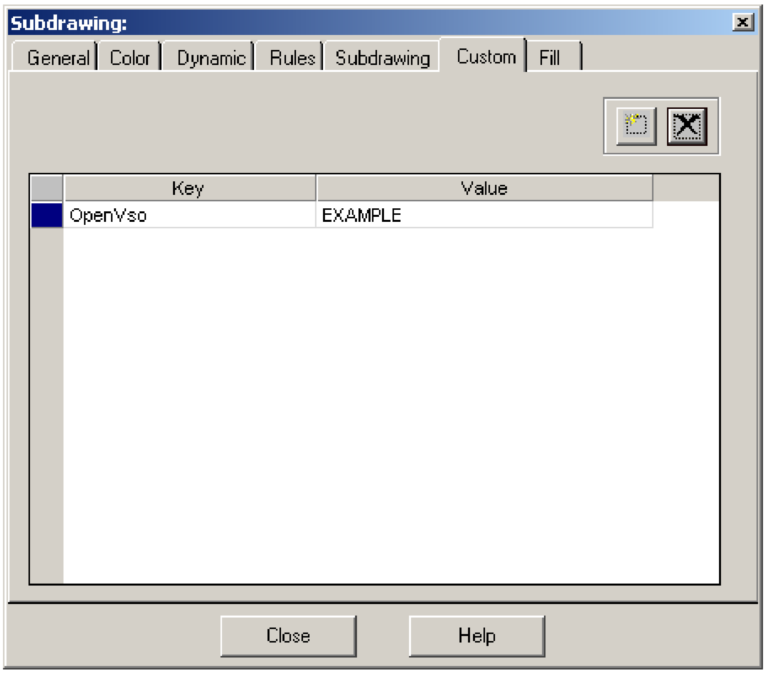


Figure 213: Defining the OpenVso custom key

    3.6. Tool IDs can be found in the Tools.ini file and each part of the file is a tool ID. If LIB 500 is installed, the file is located in \sc\lib4\base\bbone\use, otherwise it is located in \sc\Stool\Misc.
4. Add the VsoAuthGroup custom key for the subdrawing object, and define a authorization group name that specifies the authorization level for the tool in the User Management. The authorization group can be MY_TOOLS, for example.
5. In User Management, add the new authorization group for the users, for example MY_TOOLS.
6. Define the authorization level for the group.
7. Create an ini file for the tool, for example EXAMPLE.INI.

```
[EXAMPLE]
Tool_Type_=OBJ
Tool_Description_=Example Tool
Tool_VSO_File_=EXAMPLE.vso
Tool_VSO_Dialog_=Main
Tool_VSO_Method_=
Tool_Default_Path_=
Tool_Ver_Maj_=1.0

Tool_Ver_Min_=2
Tool_Icon_=Product_Icon_M
Tool_Exclusive_=1
```

8. Create the new tool ID to the Tools.ini file by adding the following lines:

```
[EXAMPLE]
Tool_Dir_ = \sc\apl\510_403_1\APLMOD4\USE
```

18.12 Command line support

Open Monitor Pro from the Command Prompt by using the following parameters:

Table 175: Parameters for opening Monitor Pro from Command Prompt

Parameter	Definition
.v-file	Open Process Display. Use "-chars around the file if there are spaces included.
alarmlist_temp1	Open Alarm Display Template 1. Preconfiguration can be given as an additional argument.1)>1)
alarmlist_temp2	Open Alarm Display Template 2. Preconfiguration can be given as an additional argument.1)
blockinglist	Open Blocking Display. Preconfiguration can be given as an additional argument.1)
eventlist	Open Event Display. Preconfiguration can be given as an additional argument.1)
reports_graphical	Open Reports Display Graphical View. Preconfiguration can be given as an additional argument.1)
reports_tabular	Open Reports Display Tabular View. Preconfiguration can be given as an additional argument.1)
trends_graphical	Open Trends Display Graphical View. Preconfiguration can be given as an additional argument.1)
trends_tabular	Open Trends Display Tabular View. Preconfiguration can be given as an additional argument.1)
uallist	Open User Activity Log Display. Preconfiguration can be given as an additional argument.
-closeonce{:all}	Close all Monitor Pros with appropriate user that have the argument defined. Ignore user with all argument.
-coordsys:world|screen	Defines the coordinate system if flags -ll and -ur are defined, default is world coordinate system.
-debug{:[process id]}	Print traces to DbgView window. Process id can be defined to enable/disable trace printing in running Monitor Pro process.
-disablemousemiddlepan	Disable panning with middle mouse button in Process Display.
-disablemousewheelzoom	Disable mouse wheel zooming in Process Display.
-disabletouchflicks	Disable flick gestures (quick, linear pen movements) in Monitor Pro. 2)2)
-disabletouchpan	Disable panning gestures in Process Display.2)
-disabletouchzoom	Disable zooming gestures in Process Display.2)
-display:ulx, uly, width, height	Monitor Pro upper left coordinates (ulx,uly), width and height (width,height).
-displaymode: [none | zoom | pan | select]	Define display mode for Monitor Pro when started with -light argument. Choice "none" applies only to Process Display. Choice "select" applies only to Trends and Reports Displays.
Table continues on next page	

Parameter	Definition
-lang:[lang code]	Define the default language (two-letter language codes) used in Monitor Pro when not logged in to any SYS600 application.
-layout [file]	After successful login the named layout file for the user is loaded.
-light	Start Monitor Pro without toolbars and menus (cannot be used with -noclose).
-ll:x1,y1 -ur:x2,y2	Lower left (x1,y1) and upper right coordinates (x2,y2) of Process Display area to be zoomed to.
-login	Login to SYS600 application. If only SYS600 application is given login dialog is shown. Session role is optional argument and when defined has to be available for the user. Additional argument can be one of the displays.
-loginonce	Login to SYS600 application that last Monitor Pro has logged to. Additional argument can be one of the displays.
-loginscript [file]	After successful login the contents of file is executed.
-loginruntoolid:[id,size]	After successful login scil tool is executed with selected size (0,1,2,3,4). Tool sizes: 0=640x480,1=960x720,2=1280x960,3=1600x1200,4=1920x1440
-loginrunvso: [file,object,look&feel,size,custom]	After successful login vso file executed with appropriate object name, modern look&feel (0,1), selected size (0,1,2,3,4) and custom scil attribute value. Tool sizes: 0=640x480,1=960x720,2=1280x960,3=1600x1200,4=1920x1440. Mask the "-chars inside custom data using \" - chars.
-logoutduration: [closemonitorpro| closetssession]	Close Monitor Pro or Terminal Server Session when autologout occurs.
-logoutonce{:all}	Logout from all Monitor Pros with appropriate user that have the argument defined. Ignore user with all argument.
-logoutscript [file]	After logout the contents of file is executed.
-noborder	Start Monitor Pro without borders (cannot be used with -noclose).
-noclose	Start Monitor Pro without possibility to close the application window from Windows system menu (cannot be used with command line arguments -noborder or -light).
-nomaximize	Start Monitor Pro without possibility to maximize the application window.
-nominimize	Start Monitor Pro without possibility to minimize the application window.
-nomove	Start Monitor Pro without possibility to move the application window.
-noresize	Start Monitor Pro without possibility to resize the application window.
-notify{:[process id]}	Print traces to SYS600 Notification Window. Process id can be defined to enable/disable trace printing in running Monitor Pro process.
-pid	Show Monitor Pro process id in application window title.
-scold:[.bat file]	When SYS600 application turns to cold the contents of .bat file is executed.
-sdown:[closemonitorpro| closetssession]	Close Monitor Pro or Terminal Server Session when SYS600 is stopped.
-setlogin	Initialize login dialog with appropriate username and SYS600 application.
-silentexit	Exit Monitor Pro without information messages in following cases: • SYS600 is forced stopped • SYS600 is stopped and -sdown:closemonitorpro is defined • Autologout occurs and it is defined to close Monitor Pro in such situation
-title:[string]	Replace the Monitor Pro application window title.
-topmost	Open Monitor Pro as topmost application window.
-tsid	Show current terminal server session id in application window title.
-wait	Wait for SYS600 to start.
-version	Show version information.
-windowname:[stringid]	Start Monitor Pro as named window.

1) Use "-chars around preconfiguration if there are spaces included.
2) Not needed if already disabled in Windows Control Panel - Pen and Touch.

Examples:

```
FrameWindow Eastwick.v
FrameWindow Eastwick.v -ll:-7832,-7832 -ur:861,1334
FrameWindow Eastwick.v -display:0,0,400,400
FrameWindow Eastwick.v -light
FrameWindow alarmlist_temp1
FrameWindow -loginscript C:\sc\prog\sa_lib\samplefile.bat
FrameWindow Eastwick.v -loginscript C:\sc\prog\sa_lib\samplefile.bat
FrameWindow eventlist -loginscript C:\sc\prog\sa_lib\samplefile.bat
FrameWindow -login AplOpera
FrameWindow -login demo "" AplOpera
FrameWindow -login demo "" AplOpera Eastwick.v
FrameWindow -login demo "" AplOpera trends_graphical my_trend_preconf
FrameWindow -login demo "" AplOpera alarmlist_temp1 my_alarm_preconf
FrameWindow -login demo "" AplOpera eventlist my_event_preconf
FrameWindow -loginonce Eastwick.v
FrameWindow -loginonce alarmlist_temp1
FrameWindow -loginonce -loginscript C:\sc\prog\sa_lib\samplefile.bat
FrameWindow -loginonce eventlist -loginscript C:\sc\prog\sa_lib
\samplefile.bat
FrameWindow -loginrunvso:c:\sc\stool\Misc\TestDlg.vso,Main,0,1,""
FrameWindow -loginrunvso:c:\sc\tool.vso,Main,0,1,"LIST(VAR1 = 8)"
FrameWindow -loginrunvso:c:\sc\tool.vso,Main,0,1,"LIST(VAR1 = \"TEXT VALUE
\")"
FrameWindow -loginrunvso:c:\sc\tool.vso,Main,0,1,"LIST(VAR1 = """TEXT
VALUE""")"
```

18.12.1 Starting Monitor Pro with a delay

If Monitor Pro is started by using the -wait command line argument, SYS600 is started with a delay and the Monitor Pro - Waiting dialog is displayed, see Figure 214. The command line argument can be used when the Monitor Pro program file (exe) is added to the **Start** menu, for example.

- Clicking **Skip Waiting** opens Monitor Pro but the connection to SYS600 is not established.
- Clicking **Exit Monitor Pro** closes Monitor Pro.

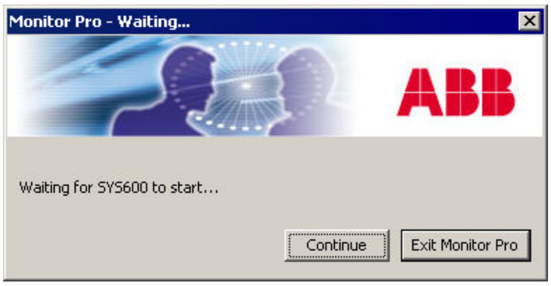


Figure 214: Monitor Pro Waiting dialog

When SYS600 is running, the Monitor Pro login dialog is displayed. For more information about login in Monitor Pro see SYS600 Operation Manual.

18.13 Customizing Monitor Pro appearance Setting toolbar visibility

Authorized users can customize Monitor Pro, for example by creating new menus and toolbars. After customizing the appearance, the corresponding file is updated. For more information about the customization file, see Table 176. When logging on, the project and application specific appearances are synchronized with the user specific appearance. Therefore, the user specific appearance is up-to-date. For more information about authorization levels, see Section 24.

The availability of the application and system specific appearances are checked when the user logs in. The toolbars are also synchronized. The unique IDs of the toolbar in the application and the system specific appearances are compared with the IDs in the user specific appearance. The unique ID is generated when the toolbar is set as application or system specific. The ID is also saved to the user specific appearance. When the user logs in and the ID does not match with the user specific appearance, the contents of application or system specific toolbar or menu is copied to the user specific appearance.

Toolbars and menus can be set as system or application specific in the customization mode in Monitor Pro. Making application specific changes to appearance requires engineering level user rights, and system manager rights are required for system level changes. For more information about authorization levels, see Section 24.

To change the appearance, right-click, for example, a toolbar and select **Set as System Specific** or **Set as Application Specific**, see Figure 215. For more information about the customization mode, see Section 18. The whole toolbar or menu is set as system or application specific, and it is updated for other users when they log in. By default, the toolbar is visible in every display. If the toolbar is closed, it is not visible in the display in question.

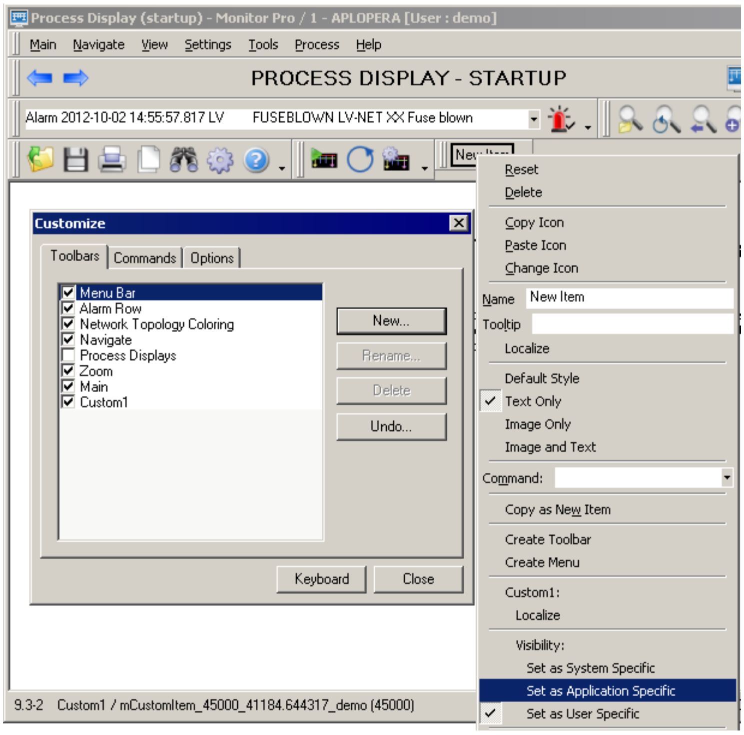


Figure 215: Setting toolbar visibility

Toolbar names support translation. For more information about translating the application specific appearance, see Section 19.7.

Table 176: Customization files

File path	Use
`\sc\prog\sa_lib\SDIActiveBar.tb2`	A default appearance for the user.
`\sc\prog\sa_lib \default_SDIActiveBar_Prj.tb2`	System specific appearance, which includes system specific toolbars and menus and which is synchronized with the user specific appearance when user logs in.
`\sc\prog\sa_lib \default_SDIActiveBar.tb2` `\sc\prog\sa_lib \default_SDIActiveBar.chg`	The appearance used when the user is logged off.
`\sc\apl\[appl]\par\[user] \SDIActiveBar.tb2` `\sc\apl\[appl]\par \[user]\SDIActiveBar.chg`	The user specific appearance.
`\sc\apl\[appl]\par\apl \SDIActiveBar_Apl.tb2`	Application specific appearance, which includes application specific toolbars and menus and which is synchronized with the user specific appearance when the user logs in.

Menu commands can be customized as follows:

- Moving, deleting, copying and resetting menu commands.
- Changing the caption and tooltip.
- Copying, pasting and changing an icon.
  While changing an icon, select a new icon from a file. For more information about icons, see Section 18.7.2.
- Changing a style to one of the following: default, text and image, or image and text. For more information about appearance modes, see Section 18.3.
- Creating a new menu command.
  A new menu uses an existing menu as a template, that is, the menu name and style is copied from the existing menu. For more information about creating a new menu command, see Section 18.5
- Adding custom menu commands, for example, execute command line and execute SCIL. For more information about custom commands, see Section 18.14.
- Authorizing
  If the main menu, a custom command with display or a picture specific menu with Process Display, is not defined. For more information about authorization levels, see Section 24.

Toolbar buttons can be customized as follows:

- Creating, copying, moving, renaming, deleting and resetting toolbar buttons.
  Resetting loads the last appearance saved.
- Changing the caption and tooltip.
- Copying, pasting and changing an icon.
  For more information about icons, see Section 18.7.2.
- Changing the appearance mode to one of the following: default, text and image, or image and text. For more information about styles, see Section 18.3.
- Creating a new menu command or a toolbar button.
  For more information about creating a new menu command, see Section 18.14.
- Authorizing
  It is not possible to authorize the Process Displays Toolbar buttons. For more information about authorization levels, see Section 24.

18.13.1 User specific appearance

The user specific appearance defines the basic appearance for the user, for example toolbars and menus. The layout file is SDIActiveBar.tb2 and it can be found in \sc\apl\[appl]\par\[user].

The user specific appearance consists of the following files:

- The TB2 file is the layout file including menus, toolbars, menu commands, toolbar buttons, icons and so on.
- The CHG file includes the layout changes, for example, toolbar positions.
- The MNU file is used when the personalized menus and toolbars are taken into use in the **Customize** dialog.

The default layout file for all users is located in \sc\prog\sa_lib. By default, the system and application specific layout files do not exist in a user specific folder. Copy the files to the folder. In addition, copy the files to the user specific folder used for translation. The Monitor Pro appearance changes are saved in the user specific file. For more information about translating user specific appearance, see Section 19.7.

18.13.2 Application specific appearance

The application specific layout file is SDIActiveBar.tb2 and it is located in \sc\apl\[appl]\par \apl. If a toolbar is set as application specific, the toolbar is copied to the application specific layout either from the user specific layout or from the system specific layout. The application specific layout is generated, if it does not exist. When logging in to the appropriate application next time, the toolbar is copied to the user specific layout. For more information about translating the application specific appearance, see Section 19.7.

18.13.3 System specific appearance

The system specific layout file is SDIActiveBar_Prj.tb2 and it can be found in \sc\prog\sa_lib \.If a toolbar is set as system-specific, it is copied either from the user specific layout or from the application specific layout to the system specific layout. The system specific layout is generated, if it does not exist. When logging in to an application next time, the toolbar is copied to the user specific layout. For more information about translating the system specific appearance, see Section 19.7.

18.14 Custom commands in Monitor Pro

New menu commands and toolbar buttons can be configured to act as custom commands. The function for the menu command and toolbar button can be changed in the customization mode in Monitor Pro. Custom commands can also be defined for the Process Display objects. For more information about creating a new menu, see Section 18.5.

Custom commands provide functions for:

- opening a display
- executing a command line
- executing a SCIL
- opening a VSCIL tool
- setting an alarm indicator

Custom commands support translation. For more information about translating the application specific appearance, see Section 19.7.

Monitor Pro supports drag and drop from Windows Explorer. Files of the appropriate format can be selected in Windows Explorer, dragged and then dropped to any toolbar in Monitor Pro.

The supported file types are:

- .v - open display, see Section 18.14.1.
- .vso - visual scil tool opening, see Section 18.14.6.2.
- .scil, see Section 18.14.5.

Files that are not supported by Monitor Pro will be executed from the command line and opened by the operating system, see Section 18.14.4.

The user should have at least authorization level 2 in authorization group PRO_CONFIG_COMMANDS in order to add Custom Commands. Authorization level 1 is required in PRO_MENU_CUSTOMIZE authorization group if custom commands are to be dropped to menus and PRO_TOOLBAR_CUSTOMIZE if dropped to toolbars.

18.14.1 Opening display

The open display function opens a defined display with specific settings. The display can be opened in the same or in a new application window.

To define a display to be opened:

1. Select **Settings/Customize...** to open the **Customize** dialog.
2. Right-click a menu or a toolbar and select **Open Display**.
3. The **Specify Display** dialog opens. See Figure 216.

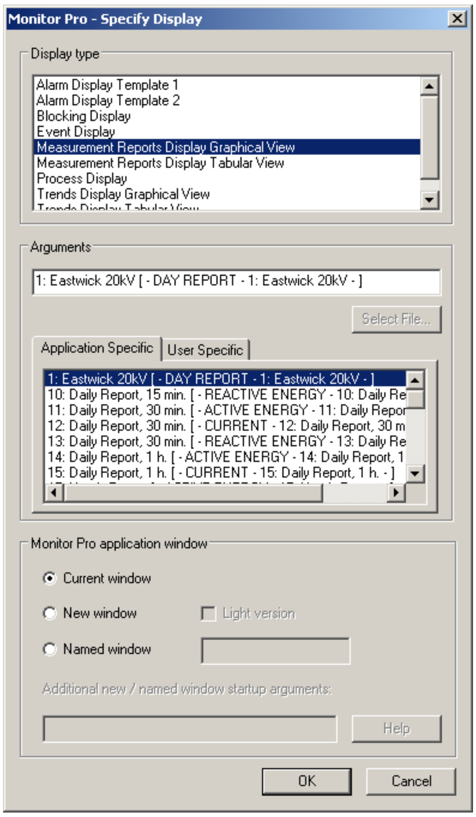


Figure 216: Specifying an opened display

4. Select a display to be opened in the **Display type** list.
5. Select a display to be opened in the **Arguments** field.
6. Select the necessary check boxes in the Monitor Pro application window field.
   - If the **New window** check box is selected, the Process Display is opened in a new window.

If the **Light Version** is selected, the window does not contain buttons or menus, and therefore only the Process Display is displayed in a window.
In the **Additional new / named window startup arguments** field, the user can write command lines, or command line options from the opened dialog can be selected by pressing the **Help** button.

- If the **Named window** check box is selected, the display will be opened to a corresponding Monitor Pro application window.
Define the window name in the text box. The new window opens with the command line argument -windowname:[stringid]. The stringid must match the one defined in the text box.
Additional startup arguments can be given as for **New window**.

7. Click **OK** to confirm the selection.

A .v file can also be dragged from Windows Explorer and dropped to the required toolbar or menu in Monitor Pro.

18.14.1.1 Alarms

The alarms function opens the Alarm List Template 1 or Alarm List Template 2 display with defined filters. The display can be opened in the same or in a new application window.

Alarm List Template 1

- Select **Alarm List Template 1** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- If filters are defined for the selected display, they are displayed in the **Predefined Values** box.

Alarm List Template 2

- Select **Alarm List Template 2** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- If filters are defined for the selected display, they are displayed in the **Predefined Values** box.

18.14.1.2 Blockings

The blocking function opens the Blocking Display with defined preconfigurations. The display can be opened in the same or in a new application window.

- Select **Blocking Display** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- If preconfigurations are defined for the selected display, they are displayed in the **Predefined Values** box.

18.14.1.3 Events

The events function opens the Event Display with defined filters. The display can be opened in the same or in a new application window.

- Select **Event Display** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- If filters are defined for the selected display, they are shown in the **Predefined Values** box.

18.14.1.4 Trends

The trends function opens the Trends Display in a Graphical or Tabular View. A display can contain preconfigurations. The display can be opened in the same or in a new application window.

Trends Display Graphical View

- Select **Trends Display Graphical View** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- If preconfigurations are defined for the selected display, they are displayed in the **Predefined Values** box.

Trends Display Tabular View

- Select **Trends Display Tabular View** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- If preconfigurations are defined for the selected display, they are displayed in the **Predefined Values** box.

18.14.1.5 Measurement Reports

The measurement reports function opens the Measurement Reports Display in a Graphical or Tabular View. The display can contain preconfigurations. The display can be opened in the same or in a new application window.

Measurement Reports Display Graphical View

- Select **Measurement Reports Display Graphical View** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- If preconfigurations are defined for the selected display, they are displayed in the **Predefined Values** box.

Measurement Reports Display Tabular View

- Select **Measurement Reports Display Tabular View** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- If preconfigurations are defined for the selected display, they are displayed in the **Predefined Values** box.

18.14.1.6 Process Display

Custom commands can also be defined for the Process Display objects.

18.14.2 Process Display with defined zoom area

Opens the Process Display with the defined zoom area. The display can be opened in the same or in a new application window.

- Select **Process Display** in the **Display type** box.
- Select a display to be opened in the **Arguments** field.
- Select the defined zoom area from application or from the user specific tab sheets.

18.14.3 Renaming Process Display

To rename Process Display:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click on the Process Display name.
3. Select **Name** and enter new name for the Process Display. See Figure 217:

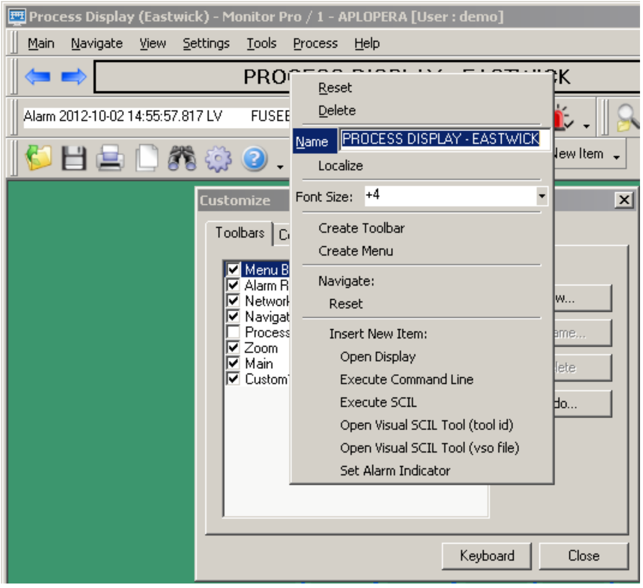


Figure 217: Renaming Process Display

18.14.4 Running command line

Runs the defined command lines. Table 177 shows the parameters of the command lines that can be defined with a BAT, COM, SCIL, VBS or VSO file. It is possible to define whether the parameters are provided for the EXE files.

Table 177: Parameters for the command lines

Parameter	Description
1	SYS600 user name
2	SYS600 user authorization level (GENERAL)
3	SYS600 user authorization level (TOOLS)
4	SYS600 user authorization level (ALARM_HANDLING)
5	SYS600 user authorization level (EVENT_HANDLING)
6	SYS600 user authorization level (REPORTS)
7	SYS600 user language
8	SYS600 user path
9	SYS600 application name
10	SYS600 application number
11	SYS600 installed drive
12	SYS600 application path
13	SYS600 application is a LIB application
14	SYS600 application is a prepared LIB application
15	Current open picture file path
16	Monitor Pro window handle
Table continues on next page	

Parameter	Description
17	Monitor Pro process ID
18	Terminal server sessions identifier
19	Mouse pointer X-coordinate, Mouse pointer Y-coordinate
20	Object name where applicable
21	OPC item ID where applicable
22	Configured authorization group for current item
23	Configured authorization level for current item
24	Real authorization for the current item in the SYS600 user management
25	SA object name where applicable
26	Object dimensions (top,left,bottom,right) when opened from context menus.
27	Object custom attributes when opened from context menus. Key and value separated with "=", key and value pairs with ",". If no value for the key exists the "-" is used.
28	Object variable mappings when opened from context menus. Key and value separated with "=", key and value pairs with ",". If no value for the key exists the "-" is used.
29	OPC Item ID (21) in SCIL format when opened from context menus
30	Computer name when running locally, terminal server client name in terminal server session
31	Custom argument that is given for Custom Commands or Tool Launcher (if applicable). If there are several arguments (=includes spaces), the rest of the attributes in this table will not match unless capsulated inside quotes, for example "My Arguments Passed to Executables".
32	VSCIL tool offset (x,y), defined in Application Settings dialog. Can be used for other tools also for harmonization.
33	Monitor Pro application window topleft (x,y) in screen coordinates
34	Monitor Pro application window dimensions (top,left,bottom,right)
35	Monitor Pro font. Consists of font name, character set and size separated with comma.
36	Monitor Pro icon size. Possible values are 16x16, 24x24 and 32x32
37	Monitor Pro background color (r,g,b)
38	Monitor Pro foreground color (r,g,b)
39	Monitor Pro highlight color (r,g,b)
40	Control Dialog offset (x,y), defined in Application Settings dialog. Can be used for other tools also for harmonization.
41	Control Dialog (inactivity) timeout, defined in Application Settings dialog. Can be used for other tools also for harmonization.
42	Monitor Pro instance ID

Define a command line to be run:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a menu or a toolbar and select **Execute Command Line**.
3. The **Specify Command Line** dialog is displayed, see Figure 218.

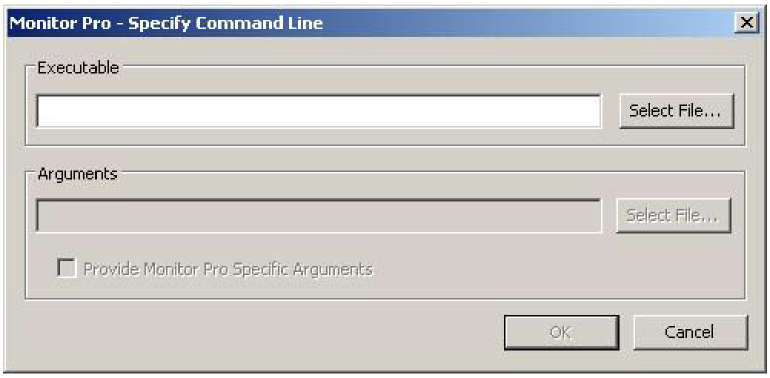


Figure 218: Specifying a command line

4. Select a file to be run in the **Executable** field.
5. Select a file to be opened in the **Arguments** field.
6. If you want to use the Monitor Pro specific arguments, select the **Provide Monitor Pro Specific Arguments** check box.
7. Click **OK** to confirm the selection.

Any executable file can be dragged from Windows Explorer and dropped to the required toolbar or menu in Monitor Pro.

18.14.5 Running SCIL

To define a SCIL file to be run:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a menu or toolbar and select **Execute SCIL**.
3. The **Specify SCIL File** dialog is displayed, see Figure 219.

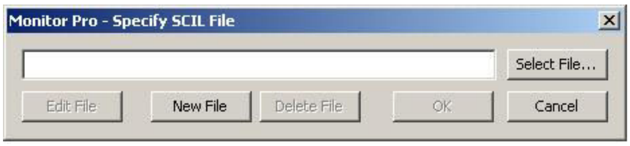


Figure 219: Specifying a SCIL file

4. Select a SCIL file to be run by clicking **Select File**.
    - the full path is required.
5. After selecting a SCIL file, the user can:
    - edit the SCIL file by clicking **Edit**.
    - create a new SCIL file by clicking **New File**.
    - delete the SCIL file by clicking **Delete File**.
6. Click **OK** to confirm the selection.

A .scil file can also be dragged from Windows Explorer and dropped to the required toolbar or menu in Monitor Pro.

Table 178 shows the arguments for executing SCIL.

Table 178: Arguments for executing SCIL

Argument	Description
USER_NAME	SYS600 user name
GENERAL_AUTH_LEVEL	SYS600 user authorization level (GENERAL)
TOOLS_AUTH_LEVEL	SYS600 user authorization level (TOOLS)
ALARM_HANDLING_AUTH_LEVEL	SYS600 user authorization level (ALARM_HANDLING)
EVENT_HANDLING_AUTH_LEVEL	SYS600 user authorization level (EVENT_HANDLING)
REPORTS_AUTH_LEVEL	SYS600 user authorization level (REPORTS)
USER_LANG	SYS600 user language
USER_PATH	SYS600 user path
MSCADA_APP_NAME	SYS600 application name
MSCADA_APP_NUM	SYS600 application number
MSCADA_INSTALLED_DRIVE	SYS600 installed drive
MSCADA_APP_PATH	SYS600 application path
IS_LIB_APP	SYS600 application is a LIB application
IS_PREPARED_LIB_APP	SYS600 application is a prepared LIB application
PROCESS_DISPLAY_PATH	Current open picture file path
WINDOW_HANDLE	Monitor Pro window handle
PROCESS_ID	Monitor Pro process ID
TERMINAL_SERVER_ID	Terminal server sessions identifier
MOUSECOORDS	Mouse pointer X coordinate, Mouse pointer Y coordinate
OBJECT_NAME_EXT	Object name where applicable
OPCITEMID	OPC item ID where applicable
CURRENT_CONF_AUTH_GROUP	Configured authorization group for current item
CURRENT_CONF_AUTH_LEVEL	Configured authorization level for current item
CURRENT_REAL_AUTH_LEVEL	Real authorization for the current item in the SYS600 user management
OBJECT_NAME	SA object name where applicable
OPCITEMID_SCIL	OPC Item ID (OPCITEMID) in SCIL format
WORKPLACE_NAME	Computer name when running locally, terminal server client name in terminal server session.
TOOL_OFFSET	VSCIL tool offset (x,y), defined in **Application Settings** dialog. Can be used for other tools also for harmonization.
APPWND_TOPLEFT	Monitor Pro application window topleft (x,y) in screen coordinates.
CONTROLDLG_OFFSET	Control Dialog offset (x,y), defined in **Application Settings** dialog. Can be used for other tools also for harmonization.
CONTROLDLG_TIMEOUT	Control Dialog (inactivity) timeout, defined in **Application Settings** dialog. Can be used for other tools also for harmonization.
Table continues on next page	

Argument	Description
INSTANCE_ID	Monitor Pro instance ID
MAPPED_DATA_VARS	Object variable mappings. Data packed to a vector with each name and value pair as a list element, e.g. MAPPED_DATA_VARS(VECTOR(LIST(Name = "ItemName", Value = "ItemValue"))).
CUSTOM_ATTRS	Object custom attributes. Data packed to a vector with each name and value pair as a list element, e.g. CUSTOM_ATTRS(VECTOR(LIST(Name = "ItemName", Value = "ItemValue"))).

18.14.6 Opening Visual SCIL tool

Opening a Visual SCIL tool can be defined by selecting the tool from the list or by defining the file used.

18.14.6.1 Selecting Visual SCIL tool from list

To select a Visual SCIL tool to be opened from the list:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a menu or a toolbar and select **Open Visual SCIL Tool (tool id)**.
3. Select the SCIL tool used from the **Predefined Values** list.
4. Select Tool size from the drop-down list.
5. Click **OK** to confirm the selection.

18.14.6.2 Defining Visual SCIL tool file

To define a file of the Visual SCIL tool:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a menu or toolbar and select **Open Visual SCIL Tool (vso file)**.
3. The **Specify SCIL Tool** dialog is displayed, see Figure 220.

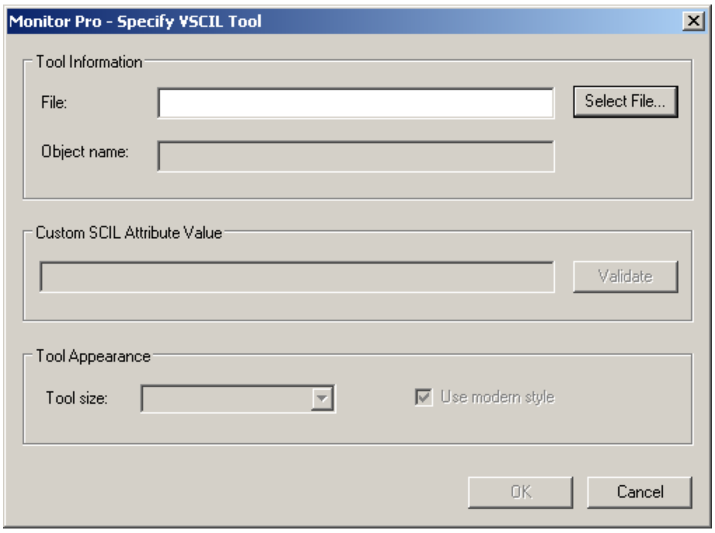


Figure 220: Selecting a VSCIL tool to be run

4. Define the VSCIL tool used in the **Specify VSCIL Tool** dialog.

- **File**
- **Object Name**
- **Custom SCIL Attribute Value**

Table 179: Examples of SCIL expressions

Expression	SCIL data type
"Some text"	TEXT
VECTOR(“A”,!)	VECTOR
LIST(LN=”ABC”, IX=10)	LIST
5.0	REAL

In option group **Tool Appearance**, some settings for the visual scil tool can be defined.

- **Tool Size**
  Defines size setting used for semigraphics tools opened from this visual scil tool. This setting is equivalent to setting monitor size when opening classic monitor. Note that this setting affects only the VSO tool opened from the menu item/toolbar button and it does not affect appearance of VSO tool itself.
- **Use Modern Style**
  Defines the look and feel of Visual SCIL controls that are used in the **VSO** dialog. When the **Use Modern Style** is checked, dialog controls are displayed in the same style as Windows dialogs, as shown in the left image of LINK IS MISSING. The default style is shown to the right of LINK IS MISSING.

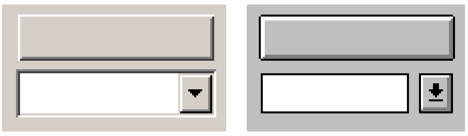


Figure 221: Effect of Use Modern Style

A .vso file can also be dragged from Windows Explorer and dropped to the required toolbar or menu in Monitor Pro.

18.14.7 Setting alarm indicator

Define and monitor the state of the configured alarm indicators based on the value in the OPC Item. The configured alarm states are described in Table 180.

Table 180: Configured alarm indicator states

State	Description	Example
0	No active, unacked alarms	The configured command is disabled
1	Active, acknowledged alarms	The configured command is enabled (not blinking)
2	Active, unacked alarms	The configured command is blinking

To set an alarm indicator for the current tool:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a menu or a toolbar and select **Set Alarm Indicator**.
3. The **Set Alarm Indicator** dialog is displayed, see Figure 222.
   - Used OPC items are displayed in a **Used OPC Items** field.
   - Free OPC items are displayed in a **Free OPC Items** field.

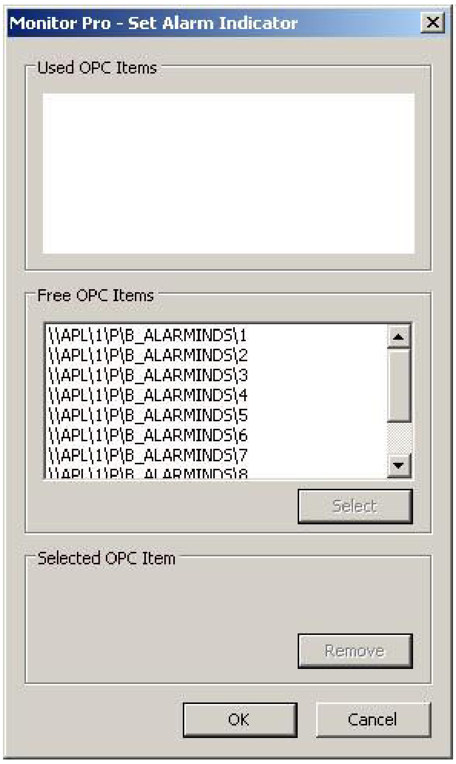


Figure 222: Setting an alarm indicator

4. Set an alarm indicator by selecting a free OPC item.
5. Click **Select** to confirm the selection.
   - The selected alarm indicators are displayed in the **Selected OPC Items** field.
   - Remove the selected alarm indicators by clicking **Remove**.
6. Click **OK** to confirm the selection.

There are 20 alarm indicators by default. More alarm indicators can be added by creating new B_ALARMINDS process objects.

18.15 Customizing Monitor Pro login

18.15.1 Configuring a shortcut key for the Login dialog

A global shortcut key can be defined for opening the **Login** dialog to \sc\prog\sa_lib \default_framewindow.ini file:

1. Find the following text in the \sc\prog\sa_lib\default_framewindow.ini file.
```
[SHORTCUTS]
;Keyboard shortcut for showing the Login dialog (Ctrl+[character]). If
not defined using the default value Ctrl+L
LoginDlg=
```
2. Write the shortcut key letter after LoginDlg key name.

18.15.2 Configuring new window login

To change Monitor Pro to automatically login when **New window** is selected:

1. Select **Settings**> **Customize...** to open the Customize dialog.
2. Right-click on **Main/New Monitor**.
3. Select the desired action, see Figure 223.

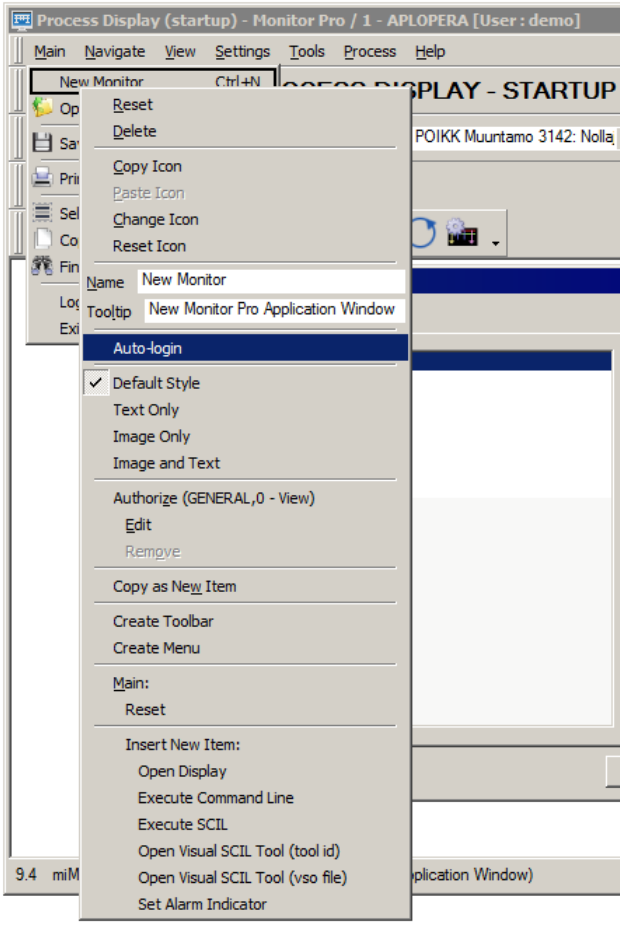


Figure 223: Configure new window login

4. Click **Close** to exit the **Customize** dialog.

18.16 Defining shortcuts to Process Displays

The user access rights can be restricted by showing the needed Process Displays as shortcuts. The user can only open the Process Displays shown in a toolbar, if the shortcut files have been defined. Process Display shortcuts are configured using the Process Display Shortcut Configuration Tool on the Process Display toolbar, see Figure 224.

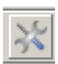


Figure 224: Process Display Shortcut Configuration Tool

Clicking on the Process Display Shortcut Configuration Tool opens the **Process Display Visibility** dialog, shown in Figure 225.

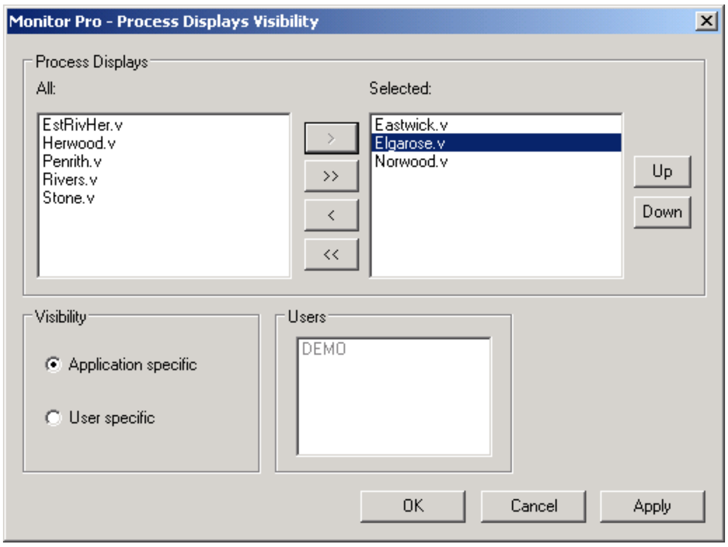


Figure 225: Process Displays Visibility dialog

The Process Displays currently available for current SYS600 application are shown in the left window. The right window displays the currently visible Process Display Shortcuts. Visibility is configured to be either Application specific of User specific.

Application specific visibility allows all the users in current SYS600 application to access the configured Process Displays.

User specific visibility allows the selected user access to the configured Process Displays.

To enable the Process Display shortcut:

1. Select the required Visibility by checking the appropriate radio button.
   If User specific visibility is selected, select the required user from the Users window to the right.
2. Select the required Process Display shortcut from the **All** list on the left.
3. Click the button with right-facing arrow (>).
   The shortcut is moved to the **Selected** list.
4. Click **OK** to apply the changes.

An authorization level 5 defined in GENERAL authorization group is required to make Process Display Shortcut modifications for other users. At least authorization level 2 is required to make application and at least authorization level 1 to make user specific Process Display Shortcut modifications.

The default order of the shortcuts is alphabetical, but it can be changed in the Process Displays Visibility Dialog by selecting a shortcut in the **Selected** list and using the **Up** and **Down** buttons on the right.

By enabling Process Display shortcuts, the following operations can be restricted from the user for appropriate Process Displays:

- Opening Process Displays by using the menu operation **Main/Open File**.
- Dragging Process Displays to an application window.
- Custom commands related to opening the Process Displays.

Shortcut files must be shortcut files at the operating system level. Copying the picture is not the right way to create the shortcuts.

Application specific Process Display shortcuts are shown to all users. The shortcut files are located in [Appl path]\PAR\APL\PROCESS\

User specific Process Display shortcuts are shown individually to each user. The user specific shortcut files are located in [User path]\PROCESS\ TOOLBAR_SHORTCUTS. The user is not allowed to open Process Displays that are not defined on toolbar of the user specific Process Displays.

If the user specific shortcut files exist, the application specific files are ignored.

18.17 Customizing Process Displays toolbar

18.17.1 Hiding Process Displays

It may also be required to hide a Process Display from the Process Displays toolbar. This is achieved by changing the file attributes of the specific Process Display file (.v file), through the file properties context menu in Windows Explorer.

18.17.2 Enabling Process Displays combo box

Process Displays combo box can be enabled by accessing the **Add or Remove** menu and clicking the **Process Displays** menu item, see Figure 226.

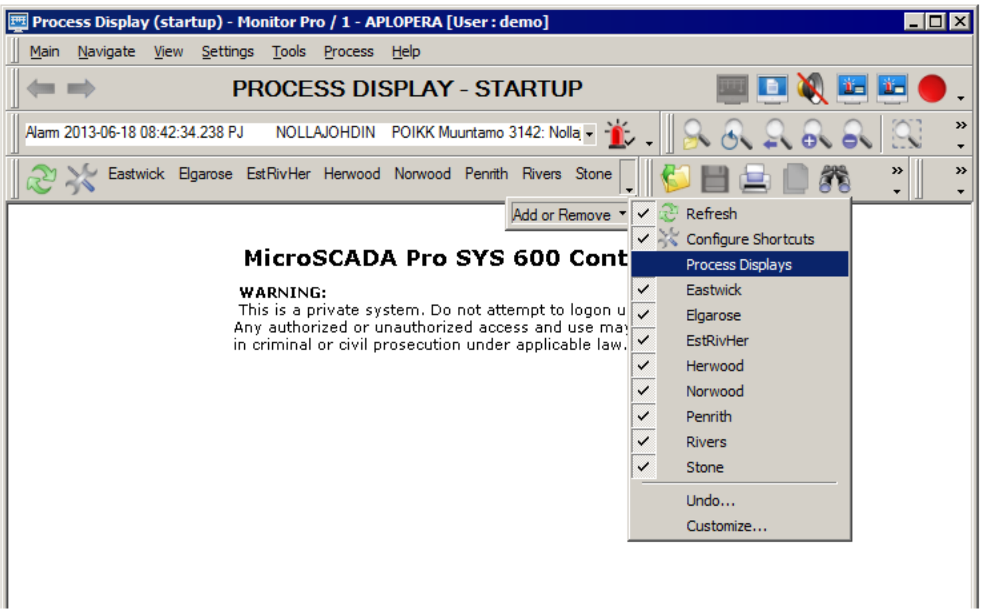


Figure 226: Process Displays combo box

Process Displays combo box can be disabled by clicking the **Process Displays** menu item again.

18.18 Resetting icons

Icons are reset in following cases:

- When there is no layout specified for the user or the layout is reset.
- Reset Layout

  - When the resetting is done based on the installation layout, the icons from the file system (icon storage) are loaded.
  - When resetting is done to backup layout file, the icons are reset based on the icons found from the backup layout file.
  - Resetting layout resets not only menus and toolbars (structure, captions, tooltips, shortcuts, and so on) but also icons based on the definitions found in the file system.
- Reset Tool (Customization context menu):
  - Resetting icon is done based on the backup layout file. If backup layout is not found, the icon is reset based on the file system (icon storage).
- Reset Icons/Icon (Customization context menu):
  - Reset is based on the icons found in the file system (icon storage). Possible user specific definitions from the menu item or toolbar button are removed (pasted or changed icon information).
  - Icon resetting is one of the items that will be done when Reset Layout is executed. For more information, see Section 18.8.
- Icon sizes:
  - If the user has not defined a user specific icon for a menu item or a toolbar button (not pasted or changed icon), the icons are reset based on the file system (icon storage). If the path to the icon is available in the user specific icon settings (change icon functionality) the specific icon size is loaded from the icon file. For ABB.ico, only size 32x16 is supported.

18.19 Customizing Alarm Row

18.19.1 Changing combo box width

To change the width of Alarm Row or any combo box:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click the Alarm Row.
3. Type in the desired width, see Figure 227.

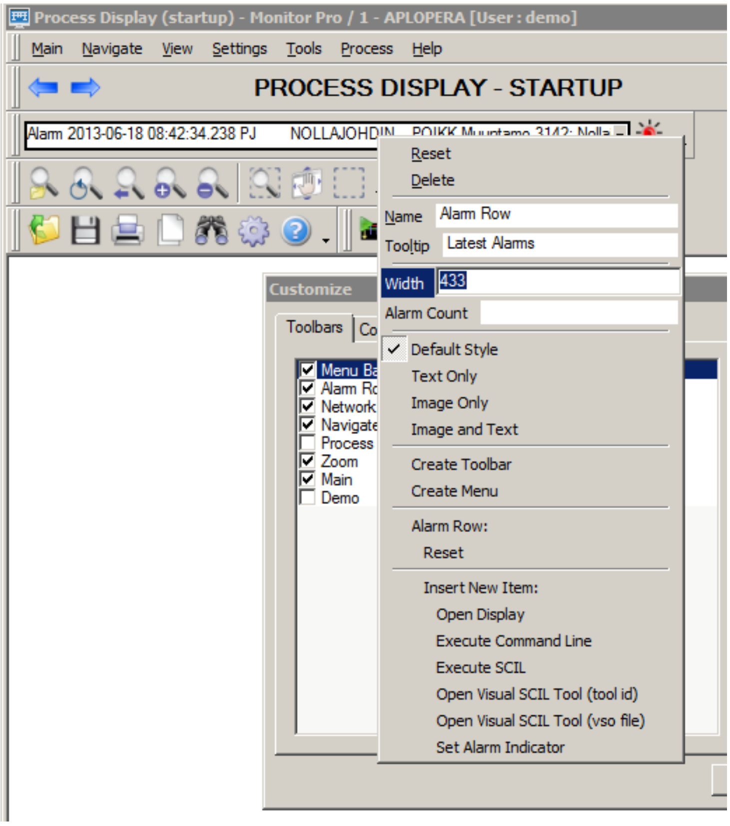


Figure 227: Setting alarm row width

4. The Alarm Row is resized.
5. Click **Close** to exit the Customize dialog.

18.19.2 Changing visible alarm count

To change the visible alarm count of Alarm Row:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click the Alarm Row.
3. Type in the desired alarm count, see Figure 228.

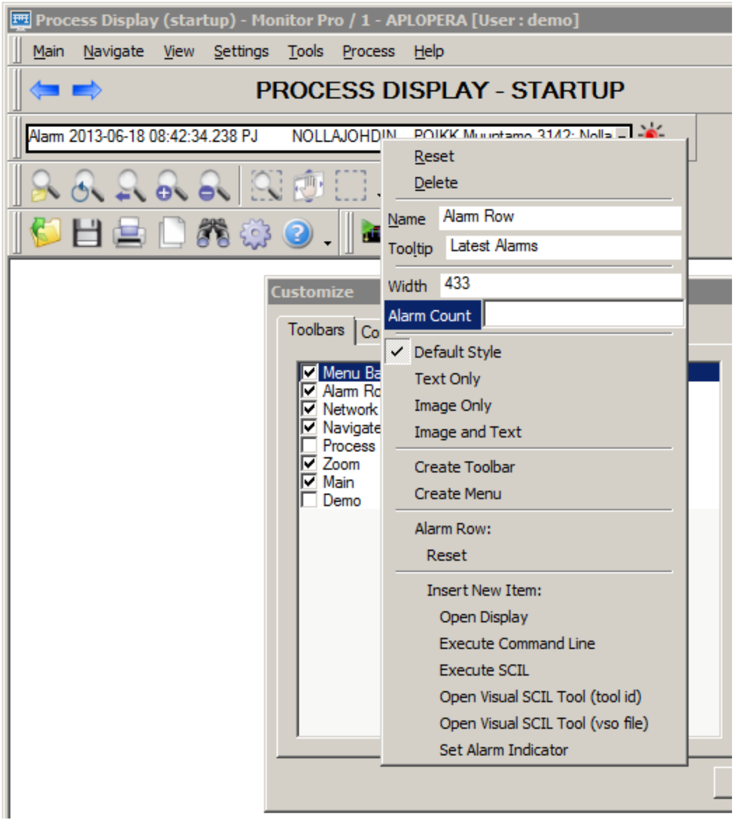


Figure 228: Changing visible alarm count

4. The visible alarm count of Alarm Row is set to the configured one.
5. Click **Close** to exit the Customize dialog.

Section 19 Localizing application

Translations of the text are included in SYS600 Visual SCIL Objects. For example, button labels and menu texts are normally defined by using the Dialog Editor and stored in same objects. For more information on using language dependent texts, see SYS600 Visual SCIL Application Design.

Other translations of application texts, for example texts describing process objects and their states, are stored in the text databases. The texts are translated explicitly by the TRANSLATION function, or implicitly by referencing the object's language sensitive attribute. Text Translation Tool translates the texts.

To save the translation, a Windows based localization tool, such as RCWinTrans, Multilizer or Lingobit Localizer can be used. For example, the Lingobit Localizer can be downloaded from http://www.lingobit.com/.

19.1 Language identifiers

It is recommended, but not required, that the two-letter language identifiers that are used in the applications are defined by the ISO standard 639.

When ISO 639 language identifiers are used, the system is able to map the Windows language IDs, which are derived from the Windows locale IDs, to the language identifiers of applications. Consequently, the OPC clients connected to the SYS600 OPC Data Access Server may define their language by the means specified in the OPC standard. The SYS600 base system automatically converts the Windows language IDs to the ISO 639 language identifiers. For more information on using language functions, see SYS600 Programming Language SCIL.

19.2 Text Translation Tool

In the SYS600 application, there is a large amount of language dependent text, which has to be translated. Text Translation Tool can be used to translate the SYS600 system's different file formats. For more information on using Text Translation Tool, see SYS600 Installation and Administration Manual.

19.3 Text databases

The translations of application texts are stored in data files called text databases. The databases have three different scopes for different needs of software components using translated texts:

1. The application text database, APL_TEXT.sdb, is designed to contain the site specific texts of the application.
2. The text databases listed by the application attribute, APL:BTD, are used by various software products, such as LIBxxx products and their localizations.
3. The system text database, SYS_TEXT.sdb, is delivered with the product and should not be modified.

The databases are searched in the scope order, APL_TEXT.sdb first and SYS_TEXT.sdb last.

It is not allowed to modify SYS_TEXT.sdb by any tool or SCIL programming language. A later version of SYS600 always overrides this file.

19.4 Monitor Pro, tools and components

Monitor Pro and some other SYS600 components use the satellite resource Dynamic Linking Libraries (.dll) to store resources. In this method, each supported language has its resources in a separate dll. Each dll uses the same resource identifiers, but also contains strings in the appropriate language.

Each dll's name and directory indicates the language of its resources. For example, Monitor Pro has executable \sc\prog\sa_lib\FrameWindow.exe, which loads its EN resources from \sc\prog\sa_lib\EN\FrameWindow_EN.dll. These language dlls can be translated by using a third party software.

Table 181: DLLs that can be translated

DLL name	Path	Description
alarmctl_EN.dll	\sc\prog\sa_lib\en	Resources for the Alarm List
Basket_texts_EN.dll	\sc\prog\sa_lib\en	Resources for the Trend Basket
blockingctl_EN.dll	\sc\prog\sa_lib\en	Resources for the Blocking List
ColorSettingTool_EN.dll	\sc\prog\graphicsEngine\system\en	Resources for the Color Setting Tool
ControlDialog2_EN.dll	\sc\prog\sa_lib\en	Resources for the Control Dialogs
eventctl_EN.dll	\sc\prog\sa_lib\en	Resources for the Event List
FrameWindow_EN.dll	\sc\prog\sa_lib\en	Resources for the Monitor Pro
IedPcm_EN.dll	\sc\prog\pcm_client\en	Resources for the IED Tools Dialog
MeasReports_texts_EN.dll	\sc\prog\sa_lib\en	Resources for the Measurement Reports
MSCGraph_texts_EN.dll	\sc\prog\sa_lib\en	Resources for the Measurement Reports and Trends
OpenRemoteDesktop_EN.dll	\sc\prog\utils\OpenRemoteDesktop\en	Resources for the OpenRemoteDesktop
Options_EN.dll	\sc\prog\sa_lib\en	Resources for the Application Settings
ReportPrint_EN.dll	\sc\prog\sa_lib\en	Resources for the printouts related to Monitor Pro Displays
TrendApp_texts_EN.dll	\sc\prog\sa_lib\en	Resources for the Trends

If some of the text still displays in English in spite of the fact that the localization file exists, the resource file is likely not up-to-date and must be updated towards the _EN.* file. The file may have new strings that are not yet localized.

19.5 Basic workflow of localization

All the separate tasks listed below are not needed, but they are recommended in order to get the whole Operator user interface translated for the selected application.

- SYS600 system tools' translation. All SYS600 tools (\sc\stool\), which the operator uses daily, need to be translated either by using Visual SCIL Editor or Text Translation Tool.
- If the application is LIB 500 based, the LIB 500 and other needed LIBxxx based applications need to be translated by using Text Translation Tool.
- Object Identifier structure can by translated by editing the SYS_BASCON.COM file.
- Translate Monitor Pro, tools and its components.

19.5.1 Translating the SYS600 System Texts

In order to use localized sdb files, \sc\prog\sa_lib\[lang code]\FrameWindow_[lang code].dll file must exist.

1. Shut down SYS600.
2. Copy \sc\sys\active\sys_\sys_text.sdb to your application folder, for example C:\sc\apl\tutor\apl_.
3. Name the file with a descriptive name, for example syst_text_de.sdb.
4. Define the new language database file for the application object. Modify the application creation routine C: \sc\sys\active\sys_\sys_bascon.com in a way that includes a definition for the new text database file added.
   For more detailed information about attribute text databases, see SYS600 System Objects.
   The following example illustrates what the application creation code may look like:

```
#CREATE APL:V = LIST(-

TT = "LOCAL",- ;Translation Type

NA = "TUTOR",- ;Name of application directory

AS = "HOT",- ;Application state (COLD,WARM,HOT)

PH = %l_Global_Paths,-

TD = VECTOR("APL_/SYST_TEXT_DE.SDB","APL_/SYST_TEXT_DE.SDB"),-

; Text databases

-; PQ = 15,- ;Number of parallel queues/ Needed in COM500 Applications

-; QD = (1,1,0,0,0,0,1,1,1,1,1,1,1,1,1),-

SV = %SV,- ;System variable (RESERVED)

CP = "SHARED",- ;Color Allocation Policy

-; RC = VECTOR("FILE_FUNCTIONS_CREATE_DIRECTORIES"),-

HP = "DATABASE",- ;History Logging Policy ("DATABASE", "EVENT_LOG",
"NONE")

EE = 1,- ;System Events & Operating System Events (1=Enabled,
0=Disabled)
```

```
AA = 1,- ;Number of APL-APL servers

MO = %MON_MAP,- ;Monitor mapping

PR = (1,2,3)) ;Printer mapping

#CREATE APL1:B = %APL
```

5. Start up the SYS600 and open Visual SCIL monitor.
6. Open Text Tool by using the Tool Manager and open the defined apl_/syst_text_de.sdb. File apl_/syst_text_de.sdb can be opened by selecting **File/Open** and navigating to the correct object in the appearing dialog, see Figure 229.

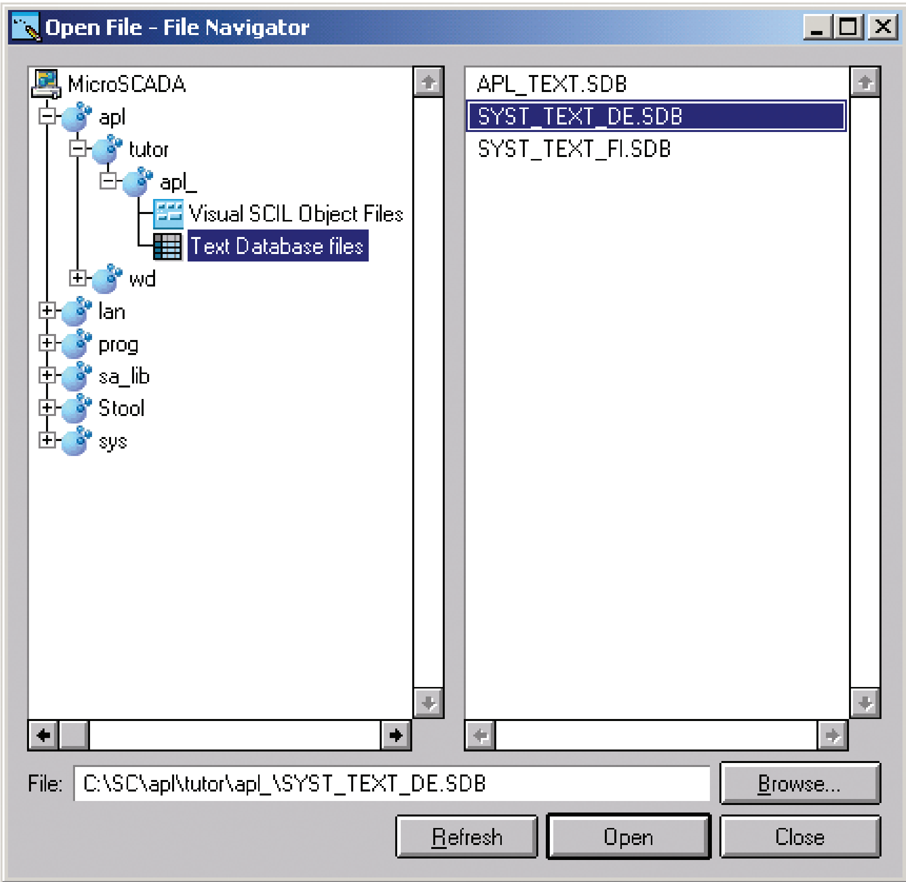


Figure 229: File Navigator dialog

For more information on how to use Text Tool, see SYS600 Installation and Administration Manual.

7. Shutdown SYS600 and modify the application language:

```
#CREATE APL:V = LIST(-

TT = "LOCAL",- ;Translation Type

NA = "TUTOR",- ;Name of application directory

AS = "HOT",- ;Application state (COLD,WARM,HOT)
```

```
PH = %l_Global_Paths,-

TD = VECTOR("APL_/SYST_TEXT_DE.SDB","APL_/SYST_TEXT_DE.SDB"),-

; Text databases

LA = "DE",-

-; PQ = 15,- ;Number of parallel queues/ Needed in COM500 Applications

-; QD = (1,1,0,0,0,0,1,1,1,1,1,1,1,1,1),-

;Parallel queue dedication/ Needed in COM500 Applications

SV = %SV,- ;System variable (RESERVED)

CP = "SHARED",- ;Color Allocation Policy

-; RC = VECTOR("FILE_FUNCTIONS_CREATE_DIRECTORIES"),-

;Revision compatibility

HP = "DATABASE",- ;History Logging Policy ("DATABASE", "EVENT_LOG",
"NONE")

EE = 1,- ;System Events & Operating System Events (1=Enabled,
0=Disabled)

AA = 1,- ;Number of APL-APL servers

MO = %MON_MAP,- ;Monitor mapping

PR = (1,2,3)) ;Printer mapping

#CREATE APL1:B = %APL
```

If the system tools are translated by using Visual SCIL Editor, all changes are lost when installing a later version of SYS600.

19.6 Translating Monitor Pro, tools and components

This section presents an example on how to use the Lingobit Localizer in translation. When Lingobit Localizer is started, it creates a new project or loads an existing project automatically. The c:\data project location is used in the example below.

1. Fill in the Name field with the project name and click **Browse** to find the project location, see Figure 230.

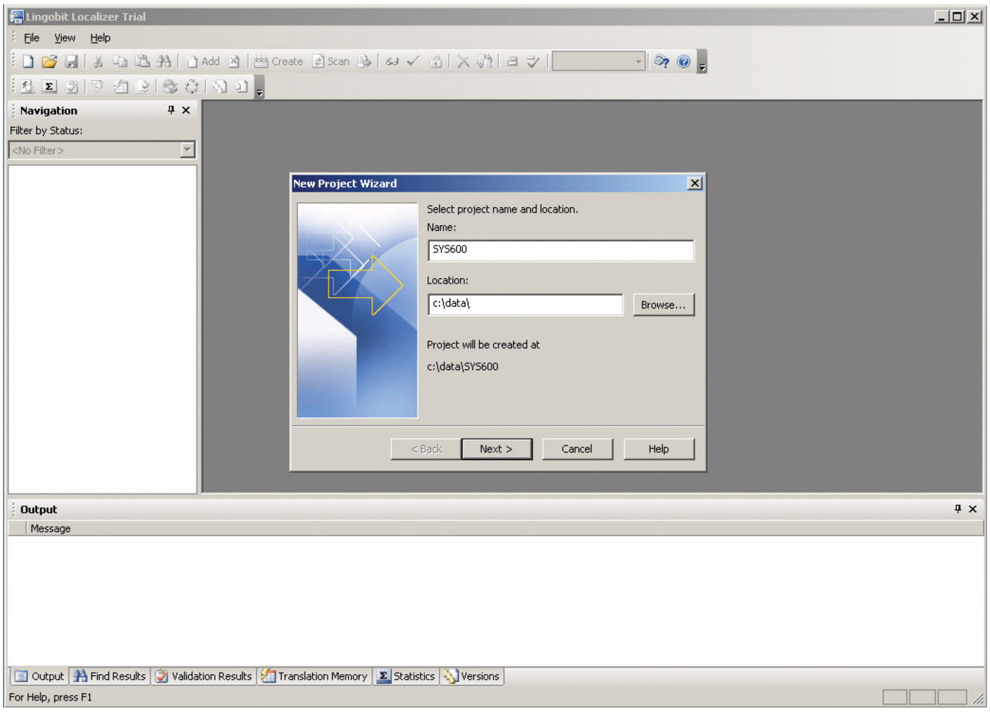


Figure 230: Main window of the Localizer and the Project Wizard dialog

2. Click **Next** to open the list of dll files, see Figure 231.
3. Click **Add** to add more files, and **Remove** if there are unnecessary files on the list.

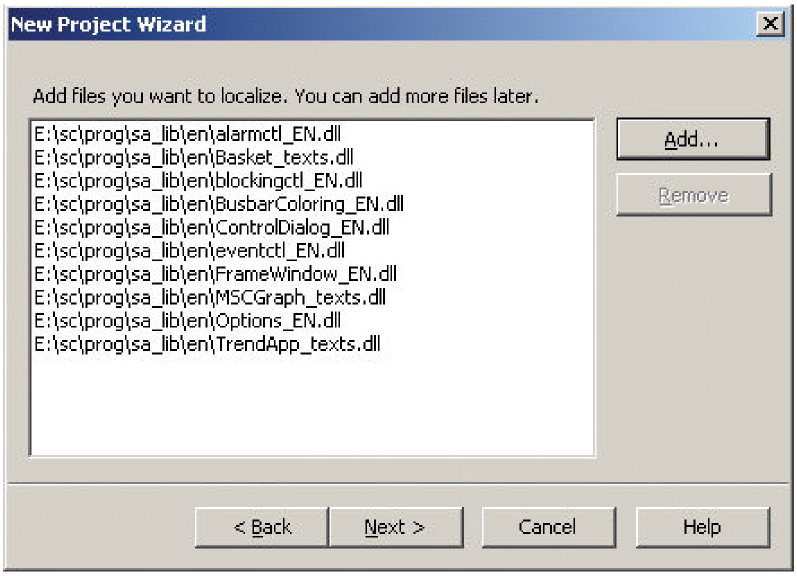


Figure 231: Example of list of dll files to be translated

4. Click **Next** to open the dialog for selecting the languages, see Figure 232.
5. Select the original language from the drop-down list and click **Add** to select the target language.

Figure 232: Selecting the original and target languages

6. Click **Next** to open the list of dll files in the application window. The list is shown on the left side of the window, see Figure 233.

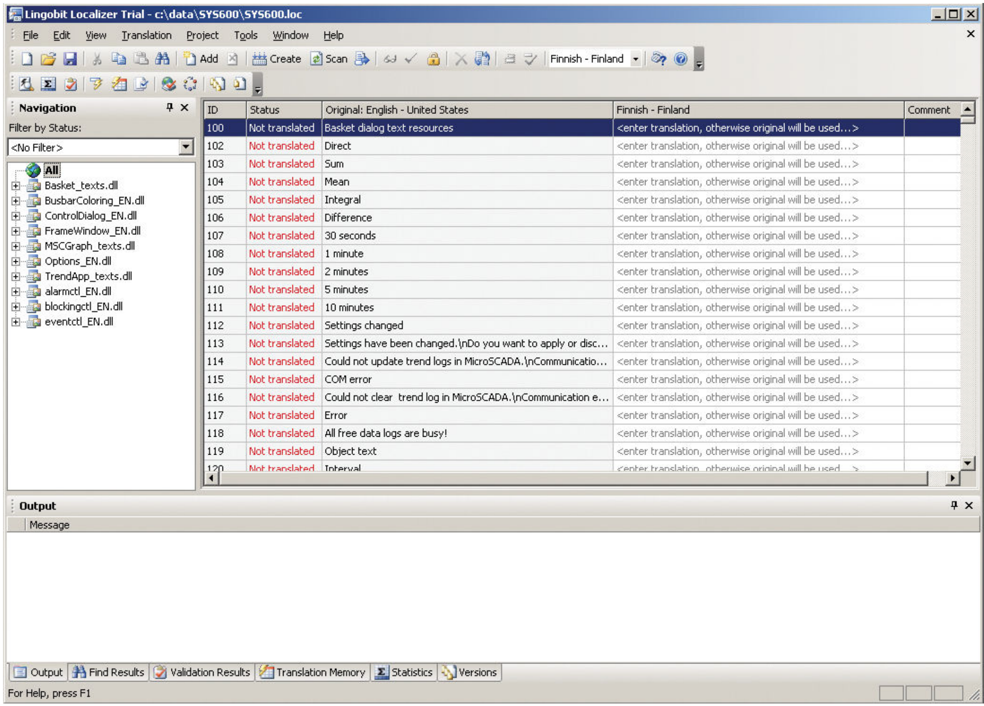


Figure 233: List of dll files in the application window

7. Click the dll files to see the strings to be translated. The resources are collected to a string table, and all the strings in each table should be translated, see Figure 234.

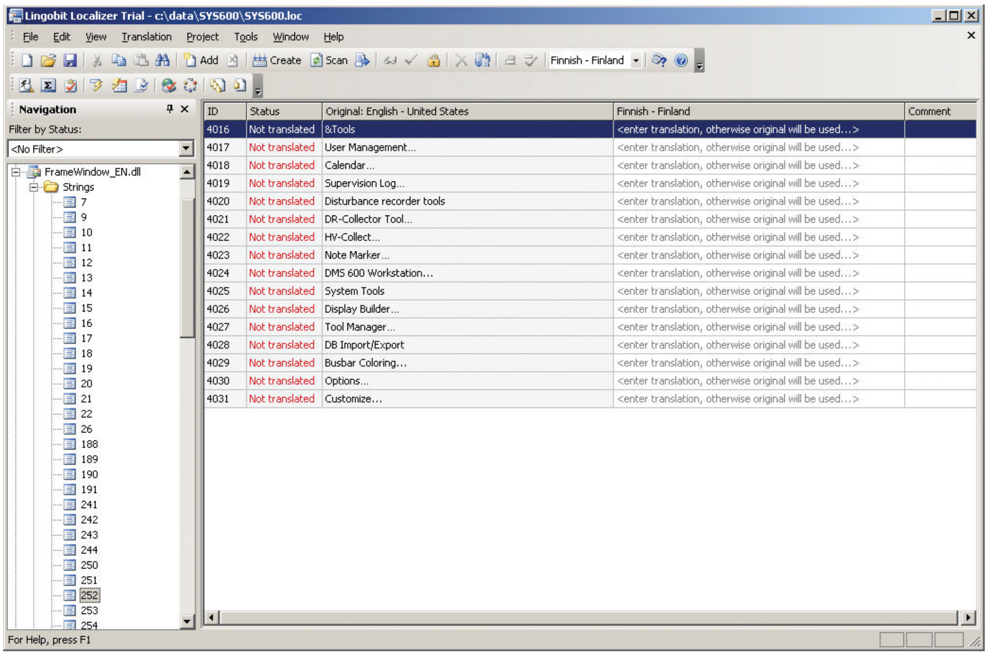


Figure 234: String table in the application window

8. Click the string to be translated and type in the translation in the language column. The strings that are not translated use the original language, see Figure 235.

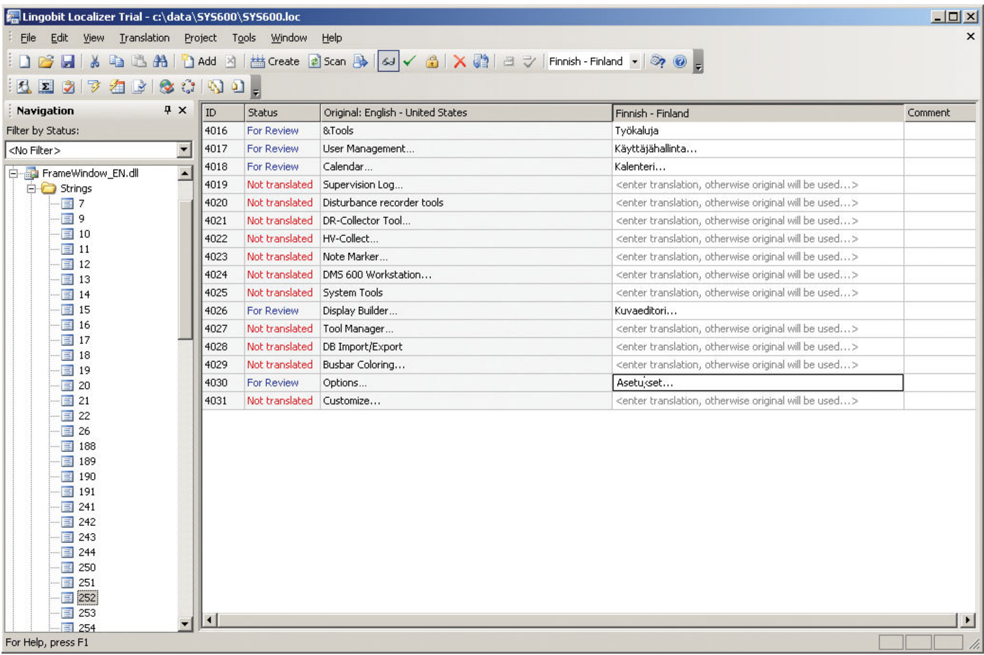


Figure 235: EN translation in the application window

9. When the translations are done, save them by selecting **Edit/Save**.
10. To generate the dll files, open the Language to export dialog by selecting **Project/Create Localized Files**, see Figure 236.
11. Select the language to which the dll files are generated to and click **OK**.

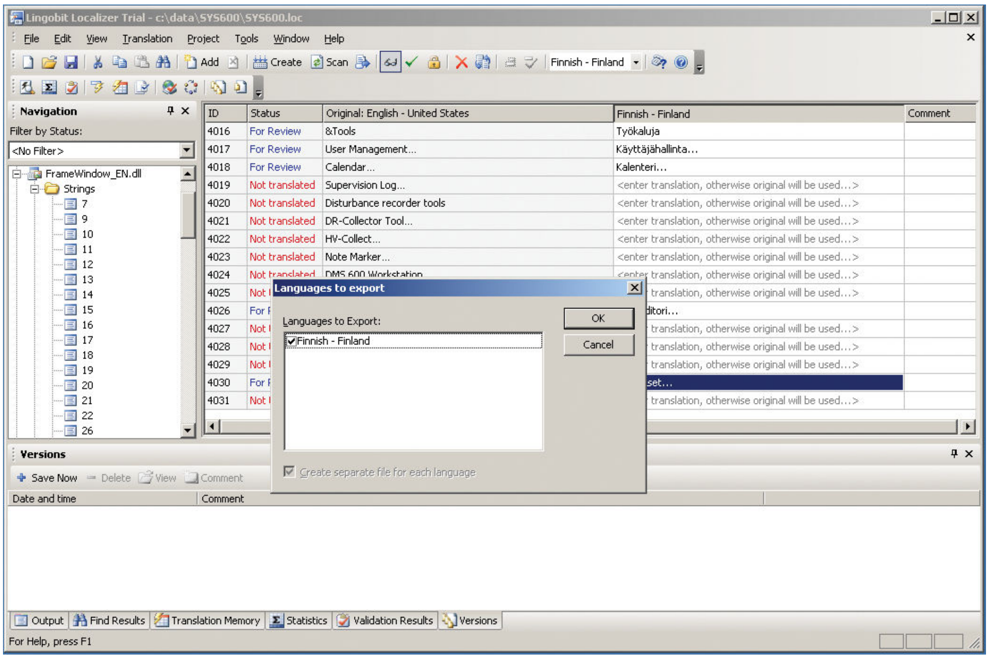


Figure 236: Generating the translation

12. The translated dll files can be found where the project was created, which is c:\data \SYS600 in this example, see Figure 237. Depending on the project setup and the generated languages, there is a directory for each language. As only Finnish language was generated in this example, there is only one subdirectory in the project structure c:\data \SYS600\Finnish-Finland.

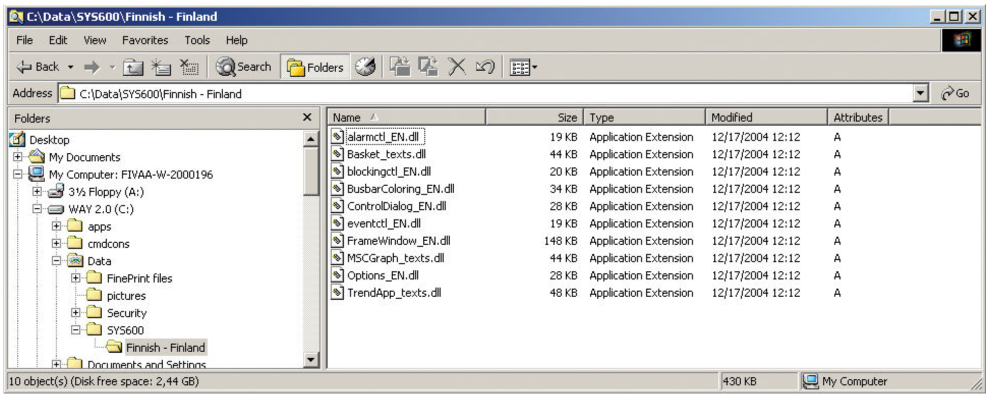


Figure 237: Contents of the localized files

13. When the files have been generated, they should be copied from the project directory to c:\sc\prog\sa_lib\FI and renamed according to Table 182.

Table 182: Renamed DLL files

DLL name	New name, if translated to Finnish-Finland
Alarmctl_EN.dll	Alarmctl_FI.dll
Basket_texts_EN.dll	Basket_texts_FI.dll
Blockingctl_EN.dll	Blockingctl_FI.dll
ColorSettingTool_EN.dl	ColorSettingTool_FI.dl
ControlDialog2_EN.dll	ControlDialog2_FI.dll
Eventctl_EN.dll	Eventctl_FI.dll

Table continues on next page

DLL name	New name, if translated to Finnish-Finland
FrameWindow_EN.dll	FrameWindow_FI.dll
IedPcm_EN.dll	IedPcm_FI.dll
MeasReport_texts_EN.dll	MeasReport_texts_FI.dll
MSCGraph_texts_EN.dll	MSCGraph_texts_FI.dll
OpenRemoteDesktop_EN.dll	OpenRemoteDesktop_FI.dll
Options_EN.dll	Options_FI.dll
ReportPrint_EN.dll	ReportPrint_FI.dll
TrendApp_texts_EN.dll	TrendApp_texts_FI.dll

19.7 Translating user interface components

Display names, menus, toolbars and custom commands can be translated in the customization mode by right-clicking the appropriate component. For more information about the **Customize** dialog, see Section 18.

Define the visibility for the custom commands before translation. For more information about visibility, see Section 18.13.

The localization information for the system specific toolbars, menus, and custom commands is saved to \sc\prog\sa_lib\default_FrameWindow.ini.

The localization information for the application specific toolbars, menus, and custom commands is saved to \sc\apl\[appl]\par\apl\FrameWindow.ini.

The localization information for the user created menus, toolbars and custom commands is saved to \sc\apl\[appl]\par\[user]\FrameWindow.ini.

For more information about Monitor Pro appearances, see Section 18.13.

For the default toolbars and menus, the localization information is presented in the language specific satellite resource Dynamic Linking Library (DLL). For more information about DLLs, see Section 19.4.

19.7.1 Translating a component

To translate a component:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a component and select **Localize** to open the Localize dialog, see Figure 238.

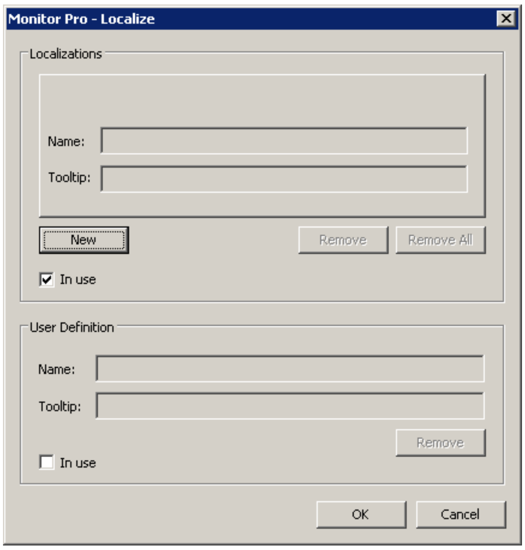


Figure 238: Localization dialog

3. Select the **In Use** check box in the **Localizations** field or the **User Definition** field.
   - To use the translations, select the check box in the **Localizations** field.
   - To use the user specific definition, select the check box in the **User Definition** field.

19.7.2 Using a translation

To use a translation:

1. Select the **In Use** check box in the Localizations field, see Figure 238.
2. Click **New** to add a new translation.
3. Select the target language, see Figure 239.

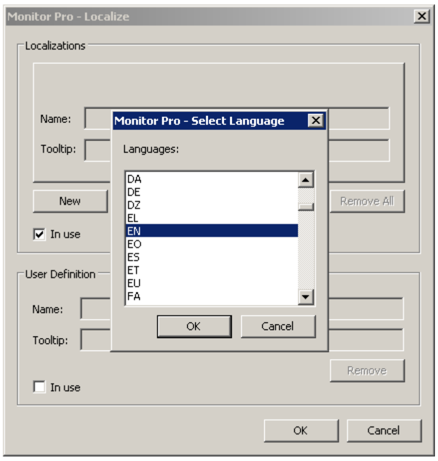


Figure 239: Selecting a used language

4. Click **OK** to confirm the target language.
5. Type the translation in the **Name** box, see Figure 240. A translated tooltip text, if available, can also be included on the **Tooltip** box.

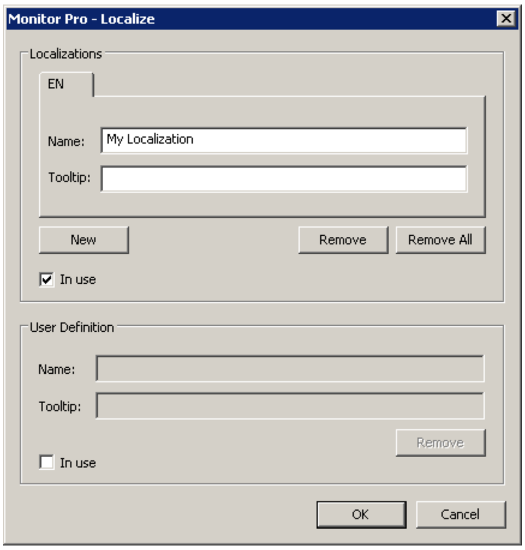


Figure 240: Typing a translation

6. Click **OK** to save the translation.

To use a user specific definition:

1. Select the **In Use** check box in the **User Definition** field, see Figure 241.
2. Type the user specific definition in the **Name** box and a short description of the component in the **Tooltip** box if one is available.
3. Click **OK** to save the definition.

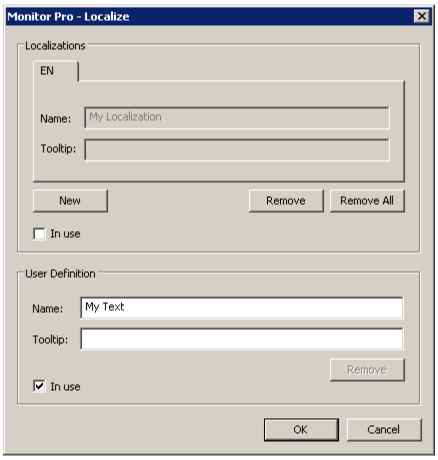


Figure 241: Defining a user specific translation

19.7.3 Removing a translation

To remove a translation:

1. Select **Settings/Customize...** to open the Customize dialog.
2. Right-click a component and select **Localize** to open the Localize dialog.
3. Click **Remove** either in the Localizations field or in the User Definition field.
4. Click **OK** to remove the translation.

19.8 Adding new language in Monitor Pro

In Monitor Pro, the operator language is based on the current user language. The user language can be changed from the User Management Tool and taken into use in the next login.

19.9 Using Asian characters

To view the Asian characters correctly:

1. To change the Windows language settings, select **Start > Control Panel > Change display language**.
2. Select the **Keyboards and Languages** tab to set the input language according to the local language to enable writing of the special characters:
    - 2.1. Click **Change keyboards...** to open the Text Services and Input Languages dialog.
    - 2.2. Click **Add** to open the Input Language dialog.
    - 2.3. To add an input language, select the appropriate language from the Input Language drop-down list. If the Keyboard layout/IME check box is selected, it is possible to write, for example, the Chinese characters with the keyboard.
    - 2.4. Click **OK**.
    - 2.5. If the Visual SCIL tools do not accept the special characters from the keyboard, set the language used as a default input language by selecting the appropriate language from the Default input language drop-down list.
    - 2.6. Click **OK**.
3. Select the **Administrative** tab to set the local language as a system locale and load the code pages:
    - 3.1. Select **Change system local...** to open the Region and Language Settings dialog.
    - 3.2. Select the appropriate language from the Language for non-Unicode programs drop-down list.

      Load the Asian characters, before selecting them as system locale. To load the characters select the Languages tab and select the Install files for East Asian languages check box. Click **Apply**.
    - 3.3. To display the text correctly, use the fonts that contain the local language characters or load more code pages. In the Code page conversion tables field, select the appropriate code pages to be loaded.
    - 3.4. Click **OK**.

      If it is not possible to write the characters directly, keystrokes can be used. To write a special character, hold down the ALT key and type the character code, for example 0230. Use the Windows Character Map tool, to find out the correct character code. The character can be pasted to the Visual SCIL text field. To open the Character Map tool, select **Start > All Programs > Accessories > System Tools > Character Map**.
4. Click **OK** and restart the computer so that the new setting takes effect.

Use Text Translation Tool to translate a new language in MicroSCADA X. For more information about Text Translation Tool, see SYS600 Installation and Administration Manual.

To take the user language into use, see Section 19.8.

19.10 Setting the default input language

The default input language can be set in the Windows Control Panel.

To set the default input language:

1. Open the Windows Control Panel **Change keyboards or other input methods**.

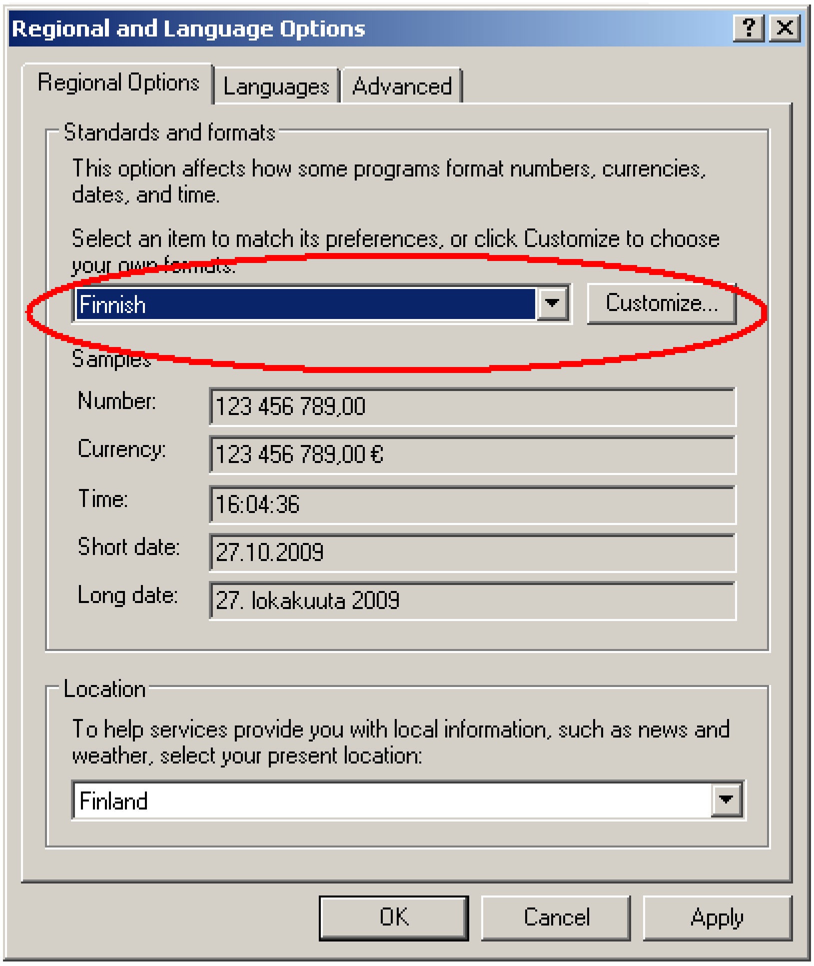


Figure 242: Regional and Language Options dialog

2. Select the **Formats** tab and select the language for the time format accordingly.
3. On the **Keyboards and Languages** tab, select **Change keyboards...**
4. In the Text Services and Input Languages dialog, select the default input language.

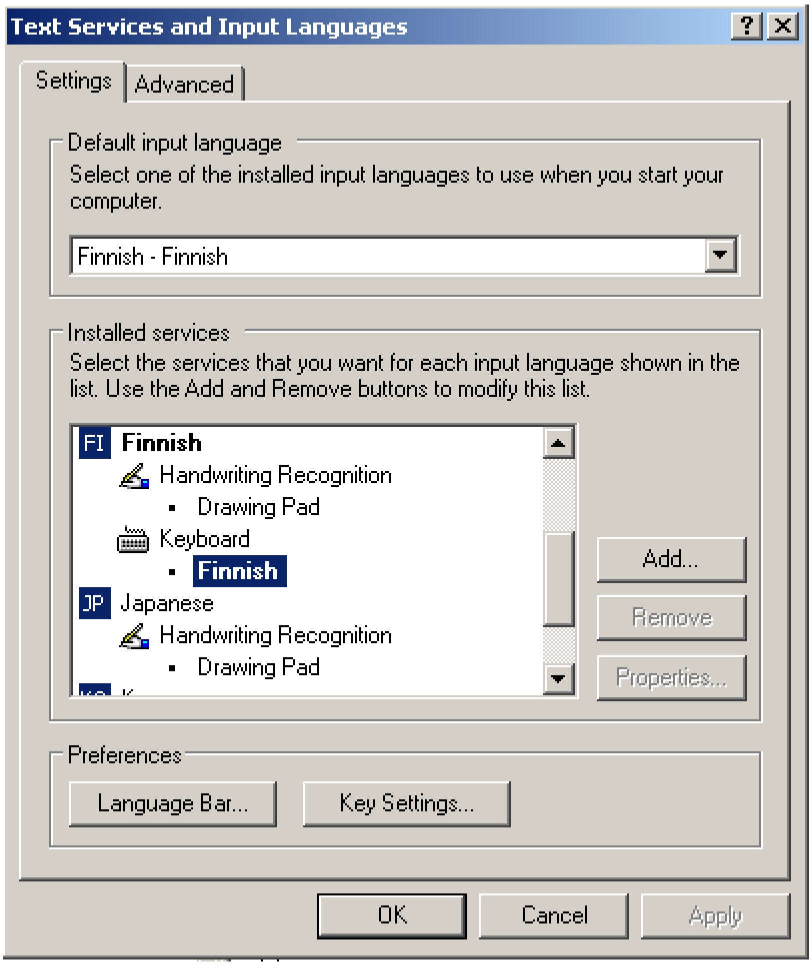


Figure 243: Selecting input language

5. On the **Administrative** tab select **Change system local...** to open the Region and Language Settings dialog.
6. Select the language for non-unicode programs.
7. Click **OK**.

19.11 Translating Workplace X user interface

19.11.1 Overview

Translating Workplace X concerns only the text content of the user interface. Graphical elements, like logos or icons are not affected by the translations.

Single Line Diagrams are an exception. Texts in SLDs and their translations are handled using a separate tool. For more information, see Process Picture design manual.

Translations are specific to content libraries. For example, if “system” and “trends” libraries have same text, both libraries need to be harvested and translated separately, and the translations for same text in different libraries can be different.

19.11.2 Translation Process

Translation of Workplace X user interface is done in three stages:

1. Harvesting. In this stage a harvesting tool scans view files and collects all translatable strings to POT files.

2. Translation. The harvested text is translated, and PO files are generated.

3. Deployment. Translation files are deployed to the Workplace X and displayed in the user interface.

19.11.2.1 Harvesting

Harvesting scans views for translatable texts and collects them to Portable Object Template (.POT) files. POT file is a plain text file that contains meta information, all harvested texts, a short reference to the location of the text and an empty translation entry.

```
msgid ""
msgstr ""
"Project-Id-Version: ABB SYS600 Web UI Library system\n"
"Content-Type: text/plain; charset=UTF-8\n"
"POT-Creation-Date: 2019-06-03T13:21:30.885Z\n"
"Language-Team: \n"
"MIME-Version: 1.0\n"
#: system/dialogs/about.xml: /document/head/title
#: system/dialogs/top-menu/top-menu.xml: /document/body/div/div/ul/li/span
msgid "About"
msgstr ""
#: system/dialogs/about.xml: /document/body/div/div/if/div/p/p
msgid "Version:"
msgstr ""
```

Harvesting Workplace X libraries
Workplace X comes with POT files already provided, harvesting of installed Workplace X libraries is not needed. The POT files can be found in the sc/prog/WebUI/libs/translations directory. The harvested files are in a subfolder with the same name as the content library with a “.pot” file name extension, for example, “system/system.pot”, “alarms-events/alarms-events.pot” etc.

Harvesting custom libraries
Workplace X comes with a command line tool harvest_library.bat for harvesting libraries. The tool can be found in the installation directory.

Harvest_library needs single argument: the name of the library to be harvested.

```
harvest_library.bat <library>
```

For example, harvesting a library “foo”:

```
C:\Program Files\ABB\MicroSCADA Pro\WebUI> harvest_library.bat foo
Building POT file for foo
Writing POT to: C:/sc/prog/WebUI/libs/foo/foo.pot
```

Harvesting application views
In case of views that are specific to single application, Workplace X provides a command line tool harvest_application.bat. The tool can be found in installation directory.

Harvest_application needs single argument: the name of the application to be harvested.

```
harvest_application.bat <application-name>
```

Example: Harvesting an application “MyApp”:

```
C:\Program Files\ABB\MicroSCADA Pro\WebUI> harvest_application.bat MyApp
Building POT file for MyApp
Writing POT to: C:/sc/apl/MyApp/views/translations/MyApp.pot
```

19.11.2.2 Translation

In the translation process, empty translations in a POT file are filled and the result is stored into a Portable Object (PO) file. PO file is a plain text file that is largely identical to POT file. The PO files contain translated texts in msgstr entries (those are empty in POT file) and they have added language dependent metadata in the header:

```
msgid ""
msgstr ""
"Project-Id-Version: ABB SYS600 Web UI Library trends\n"
"Content-Type: text/plain; charset=UTF-8\n"
"POT-Creation-Date: 2019-05-31T08:29:19.095Z\n"
"Language-Team: \n"
"MIME-Version: 1.0\n"
"PO-Revision-Date: \n"
"Content-Transfer-Encoding: 8bit\n"
"X-Generator: Poedit 2.2.1\n"
"Last-Translator: \n"
"Plural-Forms: nplurals=2; plural=(n != 1);\n"
"Language: fi_FI\n"
#: trends/dialogs/trends/graph.xml: /document/head/script
#: trends/dialogs/trends/statistics.xml: /document/body/div/grid/column
msgid "Latest value"
msgstr "Viimeisin arvo"
```

The PO file header contains information about translator tool used, translator name, plural handling information and language of the translation. These entries are usually filled by the translation tool.

Translated PO files must have same name as the language code of the target language and a ".po" postfix. Both single part and two-part language codes are permissible. For example, "en_US.po" and "en.po" are both valid file names.

POT and PO files are plain text files, so the translation can be done with a text editor. However, using dedicated translation tool will make the process much easier.

POEdit is recommended tool for Workplace X translations. It is not included in the installation package but can be downloaded for free from http://www.poedit.org.

Variable references

Translatable text may contain references to variables, values of which are supplied during run time. In these cases, translation template will contain variable reference which must be included in the translated text exactly as they are.

```
#: system/dialogs/control/secondary/common/blocks/messages.xml:
#: /document/body/details/summary
msgid "System messages (%1$s)"
msgstr "Järjestelmäviestit (%1$s)"
```

Plurals

If text contains a numerical variable reference as part of the text, the translator may need to define plural forms.

```
#: alarms-events/dialogs/alarms/alarm-global-header.xml:
#: /document/body/if/div/div/div/if/div/a
msgid "View full list (%1$s alarm)"
msgid_plural "View full list (%1$s alarms)"
msgstr[0] "Näytä kaikki (%1$s häly)"
msgstr[1] "Näytä kaikki (%1$s hälyä)"
```

Context
Same text can have different meaning depending on the context it is being used. For example, the word “Open” can have different translations depending whether the context is opening a valve, a lock, a door or a switch.

The context is defined during view creation. Harvester will automatically find the context information and includes it in POT files. Translation tool then informs the translator about context during translation process. Translating text with context does not require any extra actions – context is there simply to guide the translation.

19.11.2.3 Deployment

Translation files (.po) should be placed into translations directory to be taken into use in Workplace X. The translations directory is determined by a start parameter “—translations-path”. In SYS600 installation the translations directory is:

```
"$(regValue.Path.SOFTWARE\ABB\Products\SYS_600)prog\webUI\libs
\translations"
```

Library translation files should be placed into the translation directory under a subdirectory with the same name as the content library. For example, translation for library “userlibrary” should be placed in the directory <translations-path>\userlibrary. For installed Workplace X content libraries, the harvested POT files already exist in this directory and the translation files (.po) should be placed into the same directory.

Application translation files should be placed into the application translation directory, which is the same place where harvest_application.bat creates the .pot file. For example, MyApp spanish translation file should be placed into sc/apl/MyApp/views/translations/es.po.

19.11.3 Translations for new languages

To provide a translation for a new language, create translation for it and deploy the translation file (.po) as described in above sections. Workplace X will use the new translation when the user language is set appropriately.

Section 20 System Self Supervision

20.1 General

System Self Supervision (SSS) provides the means in MicroSCADA X systems to supervise and monitor the system itself. Hardware and software status information is shown by the application objects and supervision symbols in the product.

The status of the supervised objects can be received to MicroSCADA via system messages or predefined event channels like APL_EVENT, SYS_EVENT and OS_EVENT.

If necessary, the information is refined by supervision routing (a set of command procedures, event channels and time channels) and the end result (application objects) is used for showing events, alarms and status information.

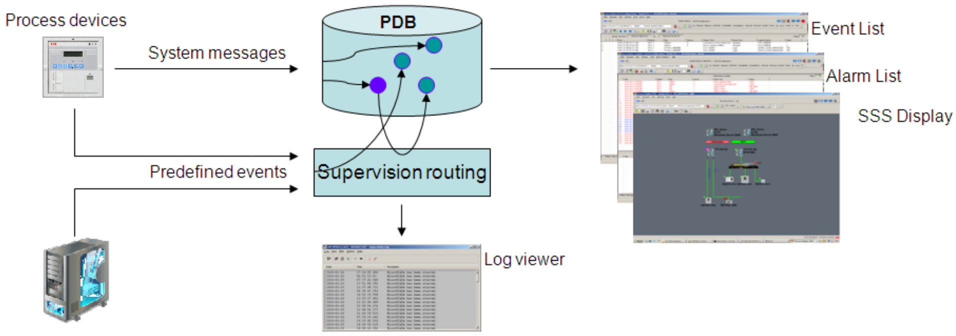


Figure 244: Basic dataflow of System Self Supervision

The following object types can be supervised:

- Base system
- Application
- PC-NET
- PC-NET line
- 61850 Node
- IED
- SNMP device
- DuoDriver
- LON Clock Master
- Display
- Printer

Typically, dedicated system supervision display is designed for MicroSCADA X system self supervision purposes. When designing such a display, the Process Display Design manual is valuable during the display engineering.

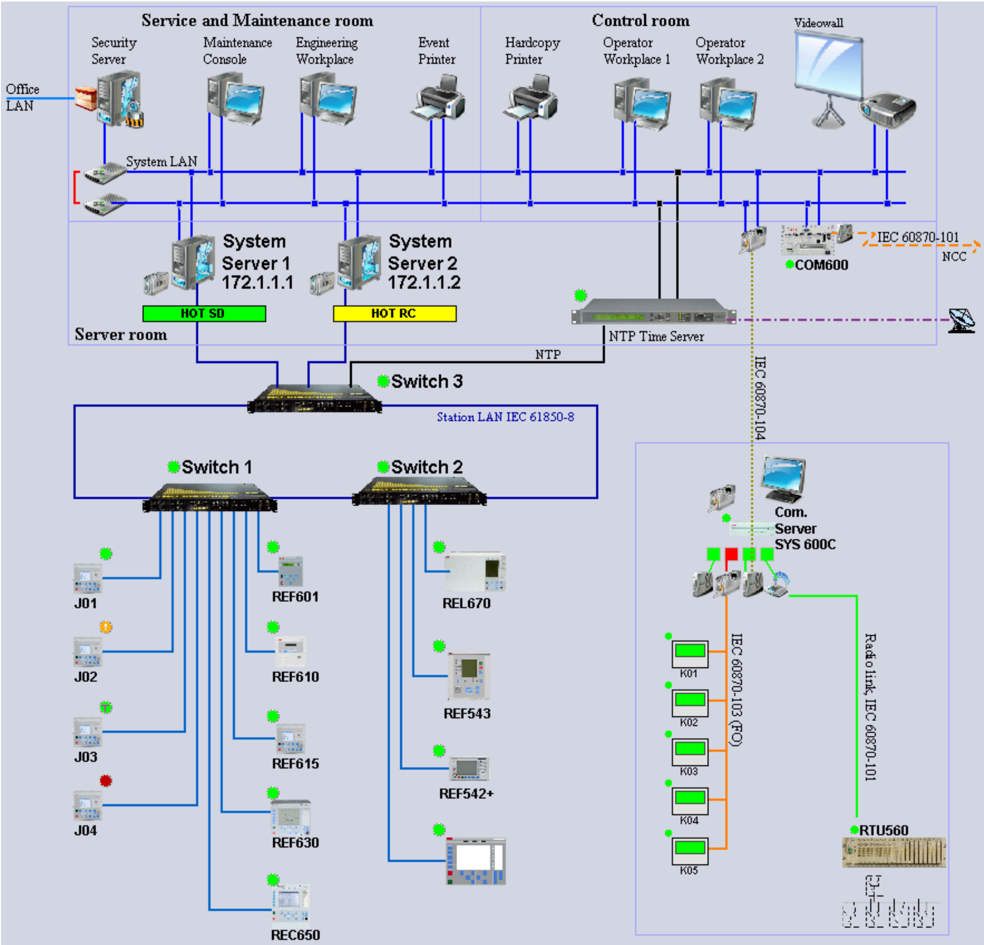


Figure 245: Example of system supervision display

20.1.1 Installation

The Supervision Base functionality is installed to the system in one of the following ways:

- Installation of SSS functionality is enabled from the Tools menu of the System Configuration Tool. The SSS base functionality will be installed according to the system configuration.
- The SSS functionality will be installed when the SSS object is installed in Object Navigator. The SSS base functionality is then installed only for that specific object at the time.

Supervision objects for Application and SNMP devices can be installed only in Object Navigator. MicroSCADA SCL Importer can create process objects for DuoDriver supervision. For other objects, SSS functionality is created by System Configuration Tool.

20.1.2 Environmental requirements

In order to obtain full operability, System Self Supervision sets the following environmental requirements:

- System Self Supervision routing switch is defined and enabled in SYS_BASCON.COM (System Variable attribute):

```
SV = (0,- ;System Variables

list(t_System_Configuration_File = "sys_/SysConf.ini",- ;PC-NET
Configuration information

b_Conf_Mech_In_Use = TRUE,- ;enables/disables start-up configuration

b_SSS_Mech_In_Use = TRUE,- ;enables/disables SSS

t_Version = "9.3")),-
```

- Operating System events are enabled in SYS_BASCON.COM (Enable Operating System Event Handler and Operating System Event Handler Filter attributes):

```
-; Operating System Event Handles Attributes

OE = 1,- ;1=Enabled, 0=Disabled

OT = (Bit_Mask(0,1,2,3,4),-

Bit_Mask(0,1,2,3,44),-

Bit_Mask(0,1,2,3,4)))
```

- The number of parallel queues used in MicroSCADA application has to be at least 4 in SYS_BASCON.COM. However, note that several tools and applications demand more queues, for example COM500i has 15 queues:
  Parallel queues and queue dedication (defined by the attributes (0...15)

```
PQ = 4,- ;Number of parallel queues

QD = (1,1,1,1),- ;Parallel queue dedication
```

  For more information about parallel queues, see SYS600 System Objects manual.
- System and Operating System events are enabled in MicroSCADA application (System Event Enabled attribute in SYS_BASCON.COM):

```
EE = 1,- ;System Events & Operating System Events (1=Enabled,
0=Disabled)
```

- Node diagnostics is enabled:

```
Nod_Permanent = list(SA = Nod_SA, LI = LAN_link, DI = 10, DT = 5)
```

- Additionally, application diagnostics can be enabled. This way, the status information of the application state from external application is received as events:

```
Apl_DI = 10,- ;Application diagnostic interval (0 = Disabled)

Apl_DT = 5 ;Application diagnostic timeout
```

20.2 Preliminary work

20.2.1 System Configuration

Installation and configuration of the communication related parts is done by using the System Configuration Tool.

Before building the application, the user should consider whether the system messages are sent as binary information, analog information or both. The required configuration is done in file SSS_SETUP.SCL. For more information about SSS_SETUP.SCL, see Section 20.5.2.4.

In practice, the SSS symbols supports binary model, so using binary model is recommended. However, it should be noted that if analog model is disabled, it is not possible to get events to the Log Viewer.

In Hot-stand-by systems, the system configuration can be done in WD application to speed up the take over time.

When the system configuration is done in WD application, some configuration is needed from the perspective of SSS in order to get the SSS related information to the main application. One possible sequence is described below:

1. Make the system configuration in WD application with the System Configuration Tool.
2. Open the configuration in the main application.
3. Enable System Self Supervision and save the configuration.
4. Open the configuration in WD application. Disable System Self Supervision and save the configuration.

If the **Supervision Filter Editor** dialog is used for setting the event filter, SSSFilter.ini file has to be copied manually to the superv directory of the main application.

Resending system messages needs to be initiated during take over. The required configuration is done in command procedures APL_INIT_1:C and APL_INIT_H. Below is an example code of how this is done for communication node and station objects.

```
#SET NET3:SOS = 1 ;updates the status of communication node

#SET STA1:SOS = 1 ;updates the status of station
```

In IEC61850 based systems, the System Messages Enabled (SE) attribute has to be set to value 4 in External OPC DA Client configuration when binary model is used.

For more information about configuring IEC 61850 based systems, see IEC 61850 System Design manual.

20.2.2 Object identification in database

SSS objects support the partitioning of the Object identification in a similar way as Power Process Library functions. By default, the following conventions are used.

Table 183: OI- structure

Field	Purpose	Comment
Station name	System name (SYS:BCX)	System name can be defined in Application Settings dialog
Bay name	Monitoring object name e.g. Station	Monitoring object names by default: OS,SYS, APL, Node, NET, Line, Station, SLCM and Printer
Device name	Monitoring object number	-

If there are more than 3 levels in OI-hierarchy, "SSS" identifier will be used for the additional levels.

20.3 Protocol support

The following communication protocols are supported by System Self Supervision:

- ASCII protocol for printer
- SPA
- LON
- RP570
- IEC 60870-5-101/104
- IEC 60870-5-103
- IEC 61850-8
- LCU 500
- MODBUS RTU/ASCII/TCP
- DNP 3.0

SNMP devices can be connected to MicroSCADA by using 3rd party SNMP OPC server.

When configuring SNMP devices, System Messages Enabled (SE) attribute has to be set to value 1 in External OPC DA Client configuration.

20.4 Display building

After system configuration is done and process objects are created, SSS Display can be build by dragging and dropping the SSS symbols in Display Builder. For more information about display building and available symbols for SSS, refer to Process Display Design manual.

20.5 System Self Supervision Base

20.5.1 Configuration

The configuration of System Self Supervision base functionality is done in System Configuration Tool. Dialog for enabling System Self Supervision can be found by selecting **Settings/System Self Supervision**.

For more information, see System Configuration manual.

20.5.2 Application Engineering information

20.5.2.1 Structure of supervision base

This section describes the structure of the Supervision Base. All files, form pictures and database objects are included. The SSS Backbone related files will be located under System Tools in the /STOOL/SUPERV folder.

20.5.2.2 Language text file

The following Text Translation Tool compatible text files are used by the SSS Backbone. The path is /STOOL/SUPERV/LANG_TEXTS.

Table 184: Text Translation Tool compatible text files

File	Functionality
SSSTexts.ini	Text file for the database creation and for log event presentations of the SSS Backbone functionality

20.5.2.3 Initialization files

The following initialization files are used by the System Self Supervision. The path is /STOOL/SUPERV/.

Default initialization files are copied under the application folder /SUPERV as the System Self Supervision functionality is applied.

Table 185: Initialization files used by System Self Supervision

File	Functionality
SSSCONF.INI	Initialization file for the configuration data of the System Self Supervision.
SSSEVENT.INI	Initialization file for the event and alarm handling of the System Self Supervision.
SSSFILTER.INI	Initialization file for the event filtering data of the System Self Supervision.

The main function of the initialization files is to include configuration data concerning the objects to be supervised in the system, and event filtering data concerning the events to be notified in supervision routing.

The data stored into these files is utilized actively during supervision routing through Supervision Routing Interpreter via System Configuration Manager Services. Initialization file for configuration data is updated during the installation of the System Self Supervision functionality.

Initialization file for event filtering data has the default settings after installation of the System Self Supervision. Default event filtering file includes references to communication units, communication unit lines, stations, LON Clock Masters, printers, predefined and specified events, that is, references to the supported objects in the System Self Supervision.

These default settings can be modified as project specific by using Supervision Filter Editor. Supervision Filter Editor can be opened from System Configuration Tool (Tools/System Self Supervision Filter Editor menu item).

20.5.2.4 Configuration files

The following configuration file is used by the System Self Supervision. The path is: /STOOL/SUPERV/.

Table 186: Configuration file used by System Self Supervision

File	Functionality
SSS_SETUP.SCL	Contains the configuration data for the general parameters of the System Self Supervision.

The configuration file contains the configuration data for the general parameters of System Self Supervision. In the specified, file the following settings are defined:

If the application specific modifications to the general parameters are needed, the configuration file can be copied under the application folder /SUPERV and do the modifications there.

The contents of general parameters in running MicroSCADA system are shown by the Supervision Filter Editor tool. This tool is accessible from System Configuration Tool.

To view these parameters:

1. Select **Tools/System Self Supervision Filter Editor**.
2. Select **View/Supervision Settings**.

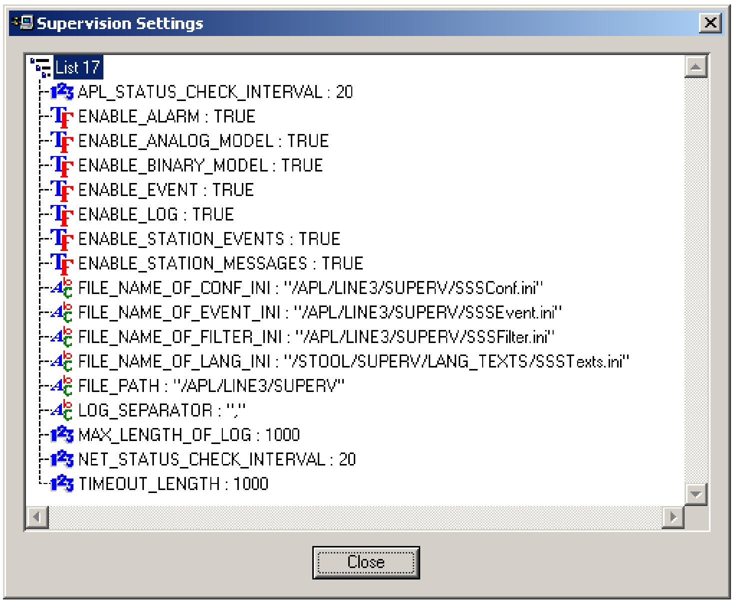


Figure 246: General parameters shown by Supervision Filter Editor

ENABLE_ALARM
In addition to primary process object receiving the system message, another process object is created for showing the alarm in Alarm List. Process objects are created dynamically as new system messages are coming to system.

ENABLE_ANALOG_MODEL
A process object type of analog input is created for receiving system messages.

ENABLE_BINARY_MODEL
A process object type of binary input is created for receiving system messages.

ENABLE_EVENT
In addition to primary process object receiving the system message, another process object is created for showing the event in Event List. Process objects are created dynamically as new system messages are coming to system.

ENABLE_LOG
Log file is updated when the system message is coming to system. In order to have this functionality, analog model needs to be enabled.

ENABLE_STATION_MESSAGES
With this flag the rerouting of system messages for stations can be enabled/disabled.

20.5.2.5 System configuration manager files

The following System Configuration Manager files are used by the System Self Supervision. The path is /STOOL/SUPERV/.

Table 187: System Configuration Manager files used by System Self Supervision

File	Functionality
SSS_MGR.MOD	Contains the source code of the System Configuration Manager services used in the System Self Supervision.
SSS_MGR2.MOD	Contains the source code of the System Configuration Manager services used in the System Self Supervision.

20.5.2.6 Other text files

The following text files are used by the System Self Supervision.

Table 188: System Configuration Manager files used by System Self Supervision

File	Functionality
SSS_APLINIT.SCL	Contains the source code for initialization of local applications at system start-up for the System Self Supervision.
SSS_APLOBJ.SCL	Contains the source code for creating command procedures, time channels, data objects and event channels for System Self Supervision.
SSS_INIT.SCL	Contains the source code for enabling/disabling the use of command procedures for System Self Supervision.
SSS_LOGINIT.SCL	Contains the source code for initialization of log files at system start-up for the System Self Supervision.
SSS_REV	Contains the revision info for the System Self Supervision
SSS_UPDATE.SCL	Contains the source code for updating application objects after revision update for the System Self Supervision.

20.5.2.7 Process objects

In System Self Supervision, the process objects are created for the following purposes:

- For receiving system messages from the communication units.
- For indicating the state of the object.
- For event and alarm handling.

The process object groups, as well as default naming and index conventions used in the database are described in more detailed below.

Table 189: Communication unit (NET) and base system nodes

Process object	Type of usage	Indexes
SYS_N<number>I	Communication unit (PC NET)	10 - indication for communication unit (PC NET) 11 - indication for redundant frontend pair 110 - routing point for communication unit (PC NET) 111 - routing point for redundant frontend pair 121...128 - routing points for communication unit lines 210 - Communication status of PC NET (BI)
SYS_N<number>I	Base system	10 - indication for global memory pool supervision
SYS_N<number>E	Event objects	Indexes are created in ascending order
SYS_N<number>A	Alarm objects	Indexes are created in ascending order

<number> = object number with 4 digits

Table 190: Station and LON Clock Master (SLCM) objects

Process object	Type of usage	Indexes
SYS_S<number>I	Station, e.g. SPA, REX, IEC, DNP	10 - indication for station (AI) 110 - routing point for station 210 - indication for station (BI)
	Station RTU	10 - indication for station (AI) 11 - indication for RTU terminal status 110 - routing point for station 111 - routing point for RTU terminal status 210 - indication for station (BI)
	LON Clock Master (SLCM)	10 - indication for station communication 11 - indication for active SLCM 12 - indication for redundant SLCM 110 - routing point for station communication 111- routing point for active SLCM 112 - routing point for redundant SLCM
SYS_S<number>E	Event objects	Indexes are created in ascending order
SYS_S<number>A	Alarm objects	Indexes are created in ascending order

<number> = object number with 4 digits

Table 191: Printer objects

Process object	Type of usage	Indexes
SYS_P<number>I	Printer	10 - indication for printer 110 - routing point for printer (only printer connected to communication unit)
SYS_P<number>E	Event objects	Indexes are created in ascending order
SYS_P<number>A	Alarm objects	Indexes are created in ascending order

<number> = object number with 4 digits

Table 192: Application objects

Process object	Type of usage	Indexes
SYS_A<number>I	Application (Local)	10 - indication for application state 11 - indication for application state (in case of a hot-stand-by system, application in the external base system node) 20 - indication for application shadowing phase (in case of a hot-stand-by system) 21 - indication for application shadowing phase (in case of a hot-stand-by system, application in the external base system node) 30 - indication for local memory pools [PICO_POOL, REPR_POOL and PRIN_POOL] (BI) 40 - indication for local application queues [Event channel queue, time channel, event, parallel and delayed execution queues, process/SCIL printouts] (BI) xxxx - Presentation object (*)
SYS_A<number>E	Event objects	Indexes are created in ascending order
SYS_A<number>A	Alarm objects	Indexes are created in ascending order

<number> = object number with 4 digits

(*) index for Presentation object is calculated with the following formulas

- Single application: 100 + application number
- Hot Stand-By application: SYS:BND * 100 + 1000 + application number

Table 193: Operating System objects

Process object	Type of usage	Indexes
SYS_O<number>	Operating System	10 - indication for System Events (BI) 11 - indication for Security Events (BI) 12 - indication for Application Events (BI)
SYS_E<number>	Event objects	Indexes are created in the ascending order
SYS_A<number>	Alarm objects	Indexes are created in the ascending order

20.5.2.8 Scale objects

At first installation of System Self Supervision, the scale 1_1 (linear 1:1 scale) is created (if it does not exist).

20.5.2.9 Data objects

At first installation of System Self Supervision, the data object for the log file counters is created (SYS_CNTR).

Each log file will contain maximum of 1000 events by default. If the number of events is exceeded, a backup file is created and the collecting of events is started again. The dedicated data object for the log file counters is allocated for counting the cumulative number of events in each of the log file categories.

20.5.2.10 Command procedures

At first installation of System Self Supervision, the following command procedures are created (the path for the source code files of the command procedures is /STOOL/SUPERV/):

Table 194: Command procedures

Command procedure	Functionality
SYS_APLER	Command procedure for receiving APL_EVENT events
SYS_NETLMR	Command procedure for receiving system messages for communication unit lines (NET lines)
SYS_NETMR	Command procedure for receiving system messages for communication units (NET)
SYS_NETSR	Command procedure for communication unit supervision.
SYS_OSER	Command procedure for receiving OS_EVENT events
SYS_PRIMR	Command procedure for receiving system messages for printers (printers that are connected to a communication unit)
SYS_SLCMR	Command procedure for receiving clock status information from the LON Clock Master (SLCM)
SYS_SSSINI	Command procedure for initialization of the System Self Supervision at system start-up
SYS_SSSREV	Command procedure for revision info for the System Self Supervision
SYS_SSSRT	Command procedure for updating functionality of the System Self Supervision at system start-up
SYS_STAMR	Command procedure for receiving system messages for stations
SYS_SYSER	Command procedure for receiving SYS_EVENT events
SYS_UPOER	Command procedure for receiving UNDEF_PROC events

20.5.2.11 Time channels

At first installation of System Self Supervision, the time channel for the communication unit supervision is created (SYS_NETSR).

20.5.2.12 Event channels

At first installation of System Self Supervision, the following event channels are created:

Table 195: Event channels

Event channel	Functionality
APL_EVENT	Event channel for receiving APL_EVENT events
OS_EVENT	Event channel for receiving OS_EVENT events
SYS_EVENT	Event channel for receiving SYS_EVENT events
SYS_NETLMR	Event channel for communication unit line system messages
SYS_NETMR	Event channel for communication unit system messages
SYS_PRIMR	Event channel for printer system messages
SYS_SLCMR	Event channel for LON Clock Master system messages
SYS_STAMR	Event channel for station system messages
UNDEF_PROC	Event channel for receiving UNDEF_PROC events

20.5.2.13 Visual SCIL object files

The following Visual SCIL object files are used by System Self Supervision. The path is /STOOL/SUPERV/.

Table 196: Visual SCIL object files

File	Functionality
SSSLVIEWER.VSO	**Supervision Log** dialog
SSSLVIEWER.INI	Initialization file for the **Supervision Log** dialog
SSS_CTRLDLG.VSO	Control dialog for SSS objects

20.6 Application supervision

20.6.1 Standard function installation

This section describes the installation of the Application status standard function from the Power Process Library. The standard function for the Application status is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Supervision/Application status).

20.6.2 Symbol installation

The Power Process symbol for the Application status standard function is installed by using the following Display Builder functions:

1. Object Browser
2. Palette

For more information, see SYS600 Process Display Design manual.

When configuring HSB systems, Application status standard function needs to be installed for both MicroSCADA nodes in Standard Function Tool. In other words, in both installations, the configuration is the same except for NODE_NUMBER. In this way, a unique process object is created for showing the application state in different nodes.

Base System symbol can be used together with Application status symbol. By default, Base System symbol is connected to attributes SYS:BCX, SYS:BNN and SYS:BON. Therefore, in HSB systems it might be necessary to change the mapping in order to get the right information from the base system. This can be done, for example, by mapping the text labels to constants.

Table 197: Symbol for application status standard function

File Name	Symbol
SYS600 Application.sd	

20.6.3 Standard function Configuration

20.6.3.1 Configuration with tools

The application standard function has the following attributes to be configured with the Standard Function Tool:

Table 198: Configurable attributes

Attribute	Meaning	Default
AUTHORIZATION_GROUP	The name of authorization group	SUPERVISION
APL_NUMBER	The physical number of the supervised applications.	APL:BAN
HOT_STAND_BY	Indicates whether the system is a single or a Hot Stand-By (Warm-Stand-By) system.	FALSE
NODE_NUMBER	The node number of the MicroSCADA base system containing the supervised application.	SYS:BND
EXT_WD_APL_NR	The logical number of the watchdog application in the "remote" (i.e. other than the current) base system, "adjacent watchdog".	0

20.6.3.2 Configuration of process objects

The database objects related to the application object are created automatically in background as Apply button is being clicked. The following process objects are created:

Table 199: Local application

Index	Obj. type	Process object	Remarks	Group identifier
10	AI	Application state	-	BSWAPLIAPL
20	AI	Application shadowing phase	-	BSWAPLIAPL
30	BI	Application local memory pools state	-	BSWAPLIAPL
40	BI	Application local queues state	-	BSWAPLIAPL
10	BI	Global memory pool state of MicroSCADA base system	-	BSWSYSISYS
1xx	AI	Application State	This index can be used for showing the application state e.g. in Monitor Pro. See Section 20.5.2.7.	BSWAPLIAPL

Table 200: External application

Index	Obj. type	Process object	Remarks	Group identifier
11	AI	Application state	-	BSWAPLIAPL
21	AI	Application shadowing phase	-	BSWAPLIAPL
xxxx	AI	Application State	This index can be used for showing the application state e.g. in Monitor Pro. See Section 20.5.2.7.	BSWAPLIAPL

20.6.3.3 Example of application status configuration

Table 201: Example of application status standard function configuration

Attribute	Value
AUTHORIZATION_GROUP	SUPERVISION
APL_NUMBER	1
HOT_STAND_BY	FALSE
NODE_NUMBER	9
EXT_WD_APL_NR	0

20.6.4 Application engineering information

20.6.4.1 Structure of application standard function

This section describes the structure of the application status standard function. All subdrawing files, form pictures, help and other text files as well as database objects are included.

The application status is a part of the standard functions of Power Process Library and has the directory /SA_LIB/BASE/BBONE and the standard Power Process Library subdirectories INST, LANG0 and USE.

20.6.4.2 Files

Table 202: Application status standard function related files

File	Functionality	Path
SYS600 Application.sd	Power Process symbol for application status standard function	/PROG/GRAPHICSENGINE/PALETTE/06-SA_Supervision
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

20.6.4.3 Help text file

The path to the application status help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 203: Application status help file

File	Functionality
SAI_APL.HLP	Standard function installation help file

20.6.4.4 Configuration files

The following configuration files are used by the Application status standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 204: Configuration files used by the application status standard function

File	Functionality
SAI_APL.DAT	Contains the configuration data for application status when it is created, modified or deleted by the configuration tools.
SAI_APLDEL.VSO	Tool for deleting process objects and SSS configuration for application status.

20.6.4.5 Command procedures

At first installation, the application object creates the following command procedures if the supervised application refers to an external application or there is a hot-stand-by (warm-stand-by) system in question.

Table 205: Command procedures

Command procedure	Functionality	Remarks
SYS_APLE1	Command procedure for application supervision.	In case of external application
SYS_APLE2	Command procedure for application supervision.	In case of external application
SYS_APLH1	Command procedure for application supervision.	In case of hot-stand-by system
SYS_APLH2	Command procedure for application supervision.	In case of hot-stand-by system

20.6.4.6 Time channels

At first installation, the application object creates the following time channels if the supervised application refers to an external application or there is a hot-stand-by (warm-stand-by) system in question.

Table 206: Time channels

Command procedure	Functionality	Remarks
SYS_APLE	Time channel for application supervision.	In case of external application
SYS_APLH	Time channel procedure for application supervision.	In case of hot-stand-by system

20.6.4.7 Event channels

At first installation, the application object creates the event channel SYS_EVENT, if it does not exist.

20.7 DuoDriver Server status

20.7.1 Standard function installation

This section describes the installation of the DuoDriver Server status standard function from the Power Process Library. The standard function for the DuoDriver Server status is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Supervision/DuoDriver Server status).

20.7.2 Symbol installation

The Power Process symbol for the DuoDriver Server status standard function is installed by using the following Display Builder functions:

1. Object Browser
2. Palette

For more information, see SYS600 Process Display Design manual.

Table 207: Symbol for DuoDriver Server status standard function

File Name	Symbol
Status-101 with label.sd	

20.7.3 Standard function Configuration

20.7.3.1 Configuration with tools

The DuoDriver Server status standard function has the following attributes to be configured with Standard Function Tool:

Table 208: Configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	SYS:BCX
BAY_NAME	Name of the bay	DuoDriver
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of process objects	SYS_D0001I
Table continues on next page		

Attribute	Meaning	Default
INSTANCE_NAME	The name of DuoDriver server instance	DD1
LINE_A	Identification of line A	Line A
LINE_B	Identification of line B	Line B
AUTHORIZATION_GROUP	Name of the authorization group	SUPERVISION

20.7.3.2 Configuration of process objects

Depending on the configuration of the DuoDriver Server status standard function, the tools create a certain set of process objects in the database. Process objects that will be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

Table 209: Process objects for DuoDriver Server status

Index	Obj. type	Process object	Remarks	Group identifier
11	BI	Line A status	Server Instance 1	BSWDDSIDUO
12	BI	Line B status	Server Instance 1	BSWDDSIDUO
13	BI	Line A status	Server Instance 2	BSWDDSIDUO
14	BI	Line B status	Server Instance 2	BSWDDSIDUO
15	BI	Line A status	Server Instance 3	BSWDDSIDUO
16	BI	Line B status	Server Instance 3	BSWDDSIDUO

20.7.3.3 Example of DuoDriver Server status configuration

Table 210: Example of DuoDriver server status standard function configuration

Attribute	Value
STATION_NAME	NCC 1
BAY_NAME	DuoDriver
DEVICE_NAME	-
P_OBJECT_LN	SYS_D0001
INSTANCE_NAME	("DD1", "DD2")
LINE_A	("Line A", "Line A")
LINE_B	("Line B", "Line B")
AUTHORIZATION_GROUP	SUPERVISION

20.7.4 Application engineering information

20.7.4.1 Structure of DuoDriver Server status standard function

This section describes the structure of the DuoDriver Server status standard function. All subdrawing files, form pictures, help and other text files as well as database objects are included.

The DuoDriver Server status is a part of the standard functions of Power Process Library and has the directory /SA_LIB/BASE/BBONE and the standard Power Process Library subdirectories INST, LANG0 and USE.

20.7.4.2 Files

Table 211: DuoDriver Server status standard function related files

File	Functionality	Path
Status-101 with label.sd	Power Process symbol for DuoDriver Server status standard function	/PROG/GRAPHICSENGINE/PALETTE/06-SA_Supervision
FORM5SA GR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

20.7.4.3 Help text file

The path to the DuoDriver Server status help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 212: DuoDriver Server status help file

File	Functionality
SAI_DUOS.HLP	Standard function installation help file

20.7.4.4 Configuration files

The following configuration files are used by the DuoDriver Server status standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 213: Configuration files used by the DuoDriver Server status standard function

File	Functionality
SAI_DUOS.DAT	Contains the configuration data for DuoDriver Server status when it is created, modified or deleted by the configuration tools.
SAI_DUOS.POT	Contains the process object definitions for the DuoDriver Server status.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for the installation tool.

20.8 DuoDriver Device status

20.8.1 Standard function installation

This section describes the installation of the DuoDriver Device status standard function from the Power Process Library. The standard function for the DuoDriver Device status is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Supervision/DuoDriver Device status).

20.8.2 Symbol installation

The Power Process symbol for the DuoDriver Device status standard function is installed by using the following Display Builder functions:

1. Object Browser
2. Palette

For more information, see SYS600 Process Display Design manual.

Table 214: Symbol for DuoDriver Device status standard function

File Name	Symbol
Status-101 with label.sd	

20.8.3 Standard function Configuration

20.8.3.1 Configuration with tools

The DuoDriver Device status standard function has the following attributes to be configured with Standard Function Tool:

Table 215: Configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of process objects	-
OPC_ITEM_PREFIX	OPC path to logical device level	IEC61850 Subnetwork.IED1.LD0
OPC_LN_INSTANCES	The instance name of IEC 61850 logical node for physical device information	("LPHD1")
DATA_OBJECT	Common Data class, Data Object and Data attribute for DuoDriver port status information	("RSS;SrcSt.stValA","RSS;SrcSt.stValB")
DATA_TEXT	Signal texts for data objects	("Port Status 1/2 Line A","Port Status 1/2 Line B")
AUTHORIZATION_GROUP	Name of the authorization group	SUPERVISION

20.8.3.2 Configuration of process objects

Depending on the configuration of the DuoDriver Device status standard function, the tools create a certain set of process objects in the database. Process objects that will be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

Table 216: Process objects for DuoDriver Device status

Index	Obj. type	Process object	Remarks	Group identifier
21	BI	Port status Line A	Server Instance 1	BSWDDDI MSC
22	BI	Line B status	Server Instance 1	BSWDDDI MSC

20.8.3.3 Example of DuoDriver Device status configuration

Table 217: Example of DuoDriver Device status standard function configuration

Attribute	Value
STATION_NAME	NCC 1
BAY_NAME	DuoDriver
DEVICE_NAME	D1
Table continues on next page	

Attribute	Value
P_OBJECT_LN	SYS_DDEV1
OPC_ITEM_PREFIX	IEC61850 Subnetwork.IED1.LD0
OPC_LN_INSTANCES	("LPHD1")
DATA_OBJECT	("RSS;SrcSt.stValA","RSS;SrcSt.stValB")
DATA_TEXT	("Port Status Line A","Port Status Line B")
AUTHORIZATION_GROUP	SUPERVISION

20.8.4 Application engineering information

20.8.4.1 Structure of DuoDriver Device standard function

This section describes the structure of the DuoDriver Device status standard function. All subdrawing files, form pictures, help and other text files, as well as database objects, are included.

The DuoDriver Device status is a part of the standard functions of Power Process Library and has the directory /SA_LIB/BASE/BBONE and the standard Power Process Library subdirectories INST, LANG0 and USE.

20.8.4.2 Files

Table 218: DuoDriver Device status standard function related files

File	Functionality	Path
Status-101 with label.sd	Power Process symbol for DuoDriver Device status standard function	/PROG/GRAPHICSENGINE/PALETTE/06-SA_Supervision
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

20.8.4.3 Help text file

The path to the DuoDriver Device status help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 219: DuoDriver Device status help file

File	Functionality
SAI_LPHD.HLP	Standard function installation help file

20.8.4.4 Configuration files

The following configuration files are used by the DuoDriver Device status standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 220: Configuration files used by the DuoDriver Device status standard function

File	Functionality
SAI_LPHD.DAT	Contains the configuration data for DuoDriver Device status when it is created, modified or deleted by the configuration tools.
SAI_LPHD.POT	Contains the process object definitions for the DuoDriver Device status.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for the installation tool.

20.9 Device status via SNMP

20.9.1 Standard function installation

This section describes the installation of the Device status via SNMP standard function from the Power Process Library. The standard function for the Device status via SNMP is found in the directory /SA_LIB/BASE/BBONE/INST and it should be installed by using the Power Process Library installation procedures (SA_LIB/Supervision/ Device status via SNMP).

20.9.2 Symbol installation

The Power Process symbol for the Device status via SNMP standard function is installed by using the following Display Builder functions:

1. Palette

20.9.3 Standard function Configuration

20.9.3.1 Configuration with tools

The Device Status via SNMP standard function has the following attributes to be configured with Standard Function Tool:

Table 221: Configurable attributes

Attribute	Meaning	Default
STATION_NAME	Name of the substation	-
BAY_NAME	Name of the bay	-
DEVICE_NAME	Name of the device	-
P_OBJECT_LN	Logical name of process objects	-
SNMP_DEVICE_TEMPLATE	Template for the SNMP device	-
AUTHORIZATION_GROUP	Name of the authorization group	SUPERVISION

20.9.3.2 Configuration of process objects

Depending on the configuration of the Device status via SNMP standard function, the tools create a certain set of process objects in the database. Process objects that will be linked to the actual process should be edited to have a station number, an address and a proper switching state. Otherwise, by default, all other attributes should have suitable values for normal operation.

By default, there are four templates provided for SNMP devices creating the followings sets of process objects.

Table 222: Device status via SNMP (Network switch 20 ports)

Index	Obj. type	Process object	Remarks	Group identifier
1 … 20	DB	Port status		BSSNMPIDEV
100	AI	Temperature		BSSNMPIDEV
102	DB	Power supply 1		BSSNMPIDEV
103	DB	Power supply 2		BSSNMPIDEV

Table 223: Device status via SNMP (GPS)

Index	Obj. type	Process object	Remarks	Group identifier
1	AI	NTP state		BSSNMPIDEV
2	DB	GPS State		BSSNMPIDEV
3	AI	Satellites		BSSNMPIDEV
4	DB	LAN Port1		BSSNMPIDEV
5	DB	LAN Port2		BSSNMPIDEV
6	DB	LAN Port3		BSSNMPIDEV

Table 224: Device status via SNMP (Computer)

Index	Obj. type	Process object	Remarks	Group identifier
1	AI	Cooling devices		BSSNMPIDEV
2	AI	Power supply		BSSNMPIDEV
3	AI	Temperature status		BSSNMPIDEV
4	AI	Temperature value		BSSNMPIDEV
5	DB	RAID status		BSSNMPIDEV
6	DB	LAN Port1		BSSNMPIDEV
7	DB	LAN Port2		BSSNMPIDEV

Table 225: Device status via SNMP (Printer)

Index	Obj. type	Process object	Remarks	Group identifier
1	DB	Paper low		BSSNMPIDEV
2	DB	Paper empty		BSSNMPIDEV
3	DB	Toner low		BSSNMPIDEV
4	DB	Toner empty		BSSNMPIDEV
5	DB	Door		BSSNMPIDEV
6	DB	Jammed		BSSNMPIDEV
7	DB	On line		BSSNMPIDEV
8	DB	Service requested		BSSNMPIDEV

20.9.3.3 Opening of Control dialog

The following predefined templates for SNMP device control dialogs can be found from file \sc \apl\'apl name'\SUPERV\SSS_CTRLDLG.INI:

- SNMP_SWITCH_20
- SNMP_SWITCH_8
- SNMP_SWITCH_4

- SNMP_GPS
- SNMP_COMPUTER
- SNMP_PRINTER

The launching of the control dialog for SNMP devices is done in Display builder/Tool Launcher settings.

1. In Display Builder, drag and drop a suitable symbol for SNMP device.
2. Open the Tool Launcher by right-clicking the symbol.
3. Select **Open/Visual SCIL**.
4. Define VSO file = SYS_TOOL/SSS_CTRLDLG.VSO.
5. Define Visual SCIL object name = MAIN and Custom argument = LIST(LN="ln", OBJECTTYPE="object type'"), where 'ln' is the logical name of process objects and object type is one of the objects types defined in file SSS_CTRLDLG.INI, e.g LIST(LN="GPS", OBJECTTYPE="SNMP_GPS").

If default OBJECTTYPECALLBACK program is changed in the configuration file, it might be necessary to adjust custom argument accordingly. Presented configuration depends on that the callback returns the contents of OBJECTTYPE list attribute.

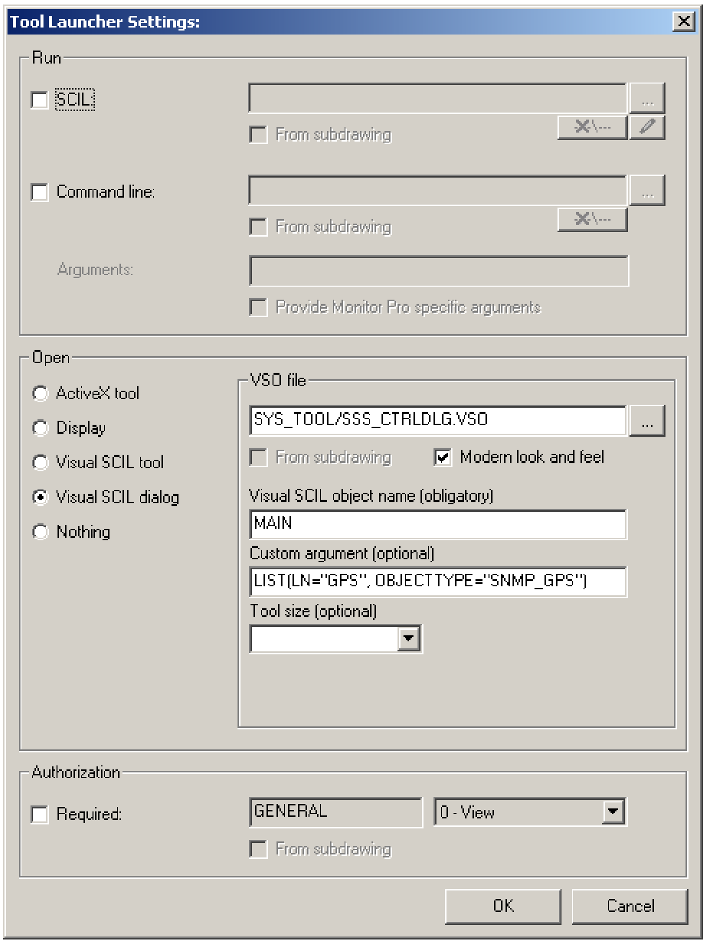


Figure 247: Tool Launcher configuration for opening Control dialog

20.9.3.4 Example of SNMP device status configuration

Table 226: Example of Device status via SNMP standard function configuration

Attribute	Value
STATION_NAME	NCC 1
BAY_NAME	Network Switch
DEVICE_NAME	1
P_OBJECT_LN	SYS_NSW1
SNMP_DEVICE_TEMPLATE	Network switch 20 ports
AUTHORIZATION_GROUP	SUPERVISION

20.9.4 Application engineering information

20.9.4.1 Structure of Device status via SNMP standard function

This section describes the structure of the Device status via SNMP standard function. All subdrawing files, form pictures, help and other text files as well as database objects are included.

The Device status via SNMP is a part of the standard functions of Power Process Library and has the directory /SA_LIB/BASE/BBONE and the standard Power Process Library subdirectories INST, LANG0 and USE.

20.9.4.2 Files

Table 227: Device status via SNMP standard function related files

File	Functionality	Path
Router.sd Status -101.sd Communication Port x 4.sd Router - RuggedRouter.sd Wireless Router.sd Switched Hub.sd Modem.sd Firewall.sd Printer.sd	Power Process symbol for Device status via SNMP standard function	/PROG/GRAPHICSENGINE/PALETTE/06-SA_Supervision
FORM5SAGR1.PIC	Generic format picture for printing	/SA_LIB/BASE/BBONE/USE

20.9.4.3 Help text file

The path to the Device status via SNMP help text file is /SA_LIB/BASE/BBONE/LANG0.

Table 228: Device status via SNMP help file

File	Functionality
SAI_SNMP.HLP	Standard function installation help file

20.9.4.4 Configuration files

The following configuration files are used by the Device status via SNMP standard function. The path is /SA_LIB/BASE/BBONE/INST.

Table 229: Configuration files used by the Device status via SNMP standard function

File	Functionality
SAI_SNMP.DAT	Contains the configuration data for Device status via SNMP when it is created, modified or deleted by the configuration tools.
SAI_SNMP.POT	Contains the process object definitions for the Device status via SNMP.
SAI_SNMP.TPL	Contains device templates for SNMP devices.
INDEX5_B1.TXT	Selection list of Power Process Library standard functions for the installation tool.

20.9.4.5 Modifying template file

It is possible to modify the existing SNMP device templates or add new templates in file /SA_LIB/BASE/BBONE/INST/SAI_SNMP.TPL. Copy the file to \sc\apl\'apl name'\APLMOD4\INST directory and make the modification there.

File format

A template file has two kinds of sections:

1. Device template section, describes the SNMP device templates in file.
2. Device types sections, describes the details of each SNMP device.

Device template section

Table 230: Device templates

Section	Meaning
DEVICE_TEMPLATES	In this section, the name of SNMP devices in the template file are described. This information is used when SNMP devices are installed with Standard Function Tool.

Device template section has the following key:

Table 231: Device types

Key	Meaning
DEVICE_TYPES	Each device template is described here. The names are separated with comma. The name of the device type has to be found as a section name in the file.

New device type can be added by introducing it in DEVICE_TEMPLATES/DEVICE_TYPES and adding the section with the same name to the end of the file.

Device type sections

Device type sections have the following keys:

Table 232: Keys in device type

Key	Meaning
INDEX	The index of the process object (IX-attribute).
INDEX_TYPE	The type of process object. Index type can be BI (Binary input) DB (double binary input) or AI (Analog input).
OBJECT_TEXT	Signal text that goes to OX-attribute.
HISTORY_ENABLED	Defines whether event is generated to event list or not. 0 = history disabled, 1 = history enabled.
Table continues on next page	

Key	Meaning
DYNAMIC_EVENT_TEXT	Defines the event handling object for the process object. If the event handling object is not defined, the default event handling object for the process object type will be used.
ALARMS_CLASS	Defines the Alarm class for the process object (AC). Value can be from range 0-7. If value is 0, no alarms will be generated.
ALARM_ACTIVATION	Defines the alarming value(s) for the process object. The value depends on the index_type. BI, the value of AG-attribute DB, the value of LA-attribute AI, the values of LI:LW:HW:HI

Each key value has to have the same amount of comma (,)characters.

20.10 Configuring SSS control dialog views

The configuration of the SSS control dialog views is done in file sc\apl\'apl name'\SUPERV \SSS_CTRLDLG.INI. This file contains the default views for applications, base system communication engines, stations and SNMP devices.

20.10.1 File format

Configuration file contains three kinds of sections:

1. General section, used for overriding the automatic detection of object type.
2. Object type sections, defines the layout of the control dialog.
3. Tabs sections, defines the content of separate tabs.

20.10.1.1 General section

Control dialog automatically resolves the object type based on the logical name of the process object. When resolving the object type, the naming conventions described in Section 20.5.2.7 are used as a rule.

SYS_?xxxx, where ? determines the type of object:

- A = BASE_APL_SINGLE or BASE_APL_HSB
- N = COM_PCNET, COM_61850, or BASE_SYS in case xxxx equals to SYS:BND or RX attribute contains BSWSYSYS
- S = STA
- xxxx - defines the object number argument

If, for some reason, the automatic detection is not suitable, OBJECTTYPECALLBACK can be used to extend the control dialog to support object types not supported by auto detection.

OBJECTTYPECALLBACK program should return an empty string for auto detection or a text which references to object type section in the configuration file.

By default OBJECTTYPECALLBACK program returns contents of OBJECTTYPE list attribute if defined in custom argument of the symbol.

20.10.1.2 Object type sections

Table 233: Object type sections

Section (Object types)	Meaning
BASE_APL_SINGLE	Single application
BASE_APL_HSB	Hot Stand-By application
BASE_SYS	Base system
COM_PCNET	PC-NET node
COM_61850	IEC 61850 node
STA_PCNET	Station for PC-NET based communication
STA_61850	Station for IEC 61850 based communication
SNMP_*	Templates for different kind SNMP devices (Switch, GPS, Computer, Printer

Each section has the following keys. If the key value is type of VECTOR, the length of the vector has to be same for these keys.

Table 234: Keys in objects types

Key	Meaning	Example value
TABS	Reference to TABS section, i.e. what tabs will be loaded when the dialog is opened.	Vector("BASE_MAIN")
VISIBILITY	Defines the minimum authorization level in SUPERVISION group for which this tab is shown. >5 not visible, 0 <= visible for all, 1 = for Auth.Level 1, 2 = for Auth.Level 2, 5 = for Auth.Level 5	Vector(0)
POSITION	Defines the location of tab in control dialog. "B" = Main view, "A" = Advanced view	Vector("A")
START_MODE	Defines the default view of the control dialog when opened. "BASIC"=dialog is opened in main view, "ADVANCED"=Dialog is expanded automatically to advanced view.	ADVANCED
OI1	Defines the data to be shown in 1st level of "Object Identification" field in the dialog.	SYS:BCX
OI2	Text to be shown in 2nd level of OI field. Same behaviour as OI1.	SYS:BNN

20.10.1.3 Tab sections

Keys in the TAB section depend on the type of tab (the value of OBJECT key). The common keys for all tabs and keys of the built in tabs are described below.

Table 235: Tab sections

Key	Meaning
OBJECT	The name of Visual SCIL object that is loaded from VSO. Name is prefixed with NBP_ by the dialog framework and the object must be of a type NOTEBOOK_PAGE
VSO	The path to the VSO file where the OBJECT is loaded. If this key is missing or empty SSS_CtrlDlg.VSO is defaulted to. VSO and OBJECTS keys can be used to extend the control dialog functionality without the need to modify the dialog code itself.
TABTITLE	Defines the title of the dialog tab. This key is EVALUATED by SCIL and can contain variables expansions.
PRELOAD	This key defines a function which is executed before the tab loading is started. It can be used to initialize global variables needed by the notebook page's init method. If STATUS after PRELOAD method is non-zero, the tab loading is skipped.
POSTLOAD	The function defined in this method is EVALUATED and executed after the tab is successfully loaded. In case the STATUS after the tabs LOAD call is non-zero POSTLOAD method is not executed.
ONSELECT	The function in this key is EVALUATED and executed every time user selects the tab in question.

20.10.1.4 Predefined tab types

This section lists the predefined tab types and the related configuration settings.

General table
With general table, is possible to show information from MicroSCADA base system or file system in a table view.

OBJECT = GEN_TABLE in tab definition

Table 236: Keys for general table

Key	Meaning
ATTRIBUTES	Values to show in the table. This vector specifies the attributes to be added to the table. Either objects from MicroSCADA or file system.
SOURCE	Vector with the equal length as ATTRIBUTES vector. Specifies where the value in question is searched from. Possible values are "SYS", "FILE" or "STATIC". Static definition only disables the variable expansion that is done with the SYS type.
VISIBILITY	Similar visibility vector as for Object type. Vector defines the minimum authorization level for corresponding attribute to be shown.
ATTRTITLES	Attribute name or description to be shown in the left column of table. This is key is optional and if it is missing SCIL function APPLICATION_OBJECT_INFO is attempted. In case the value of an element is "", the same SCIL function is used. If the function fails to provide a name for the attribute, the attribute itself is shown.
UPDATERATE	How often the table is refreshed. Value is in seconds. Missing key or 0 means that the table is only refreshed when the dialog is opened.
FILEx	This key has to be referenced from ATTRIBUTES key. It defines the file which is read.
FILEx_PRE	This key defines a callback function which is called every time the value of FILEx is about to be refreshed. It is not recommended to call external programs whose execution takes more than the update rate, as this would render the dialog unusable.

Table continues on next page

Key	Meaning
FILEx_POST	This key defines the callback, which is called after the file value has been refreshed.
COLSEPARATOR	Key defines the separator character used for splitting the file input into a vector. Key must be single character (suitable for SCIL function SEPARATE). If not defined, " " if used.
EVENT_UPDATE	Key defines a vector of process objects whose update causes the table to be refreshed. Refresh is similar that is done with UPDATERATE, but is event based instead time based. If more than one PO listed in the key is refreshed simultaneously, each update causes the table to be refreshed.

Force takeover

With the **Force takeover** tab, it is possible to initiate take over in Hot Stand-By system.

OBJECT = FORCE_TAKEOVER in tab definition

This tab has no specific configuration options.

PC-NET and Line control

With PC_NET and Line control tab is possible to show status of PC-NET and lines defined under it. Both PC_NET and separate lines can be set in use/out of use.

OBJECT = NET_CTRL in tab definition

Table 237: Keys for PC-NET and Line control

Key	Meaning
PCNET_CONTROL	Defines the minimum authorization level in SUPERVISION group to allow PC-NET Control. If not defined, defaults to 2.
LINE_CONTROL	Defines the minimum authorization level in SUPERVISION group to allow setting individual lines in and out of use. If not defined, defaults to 2.

Device control

With the **Device control** tab, it is possible to set station in use/out of use, send general interrogation for station and send time synchronization command to the station.

OBJECT = DEV_CTRL in tab definition

Table 238: Keys for Device control

Key	Meaning
STATION_CONTROL	Defines the minimum authorization level required in SUPERVISION group to control the station.
GI_CONTROL	Defines the minimum authorization level required to send GI command.
SYNC_CONTROL	Defines the minimum authorization level required to send Time Sync command to the device.
GICALLBACK	In case there is a need for GI for station types not supported by default implementation, this key can define a vector which will implement the GI
TIMESYNCCALLBACK	In case there is a need for time sync for time sync commands for unsupported station types, this key should define the vector which will implement the command.

Alarms tab

Shows the alarm states under one process object group (LN) and has possibility to acknowledge all or selected alarms

OBJECT = ALARMS in tab definition

Table 239: Keys for Alarms

Key	Meaning
LOGICALNAME	Overrides the logical name where alarm statuses are fetched. Defaults to LN defined in the symbol.

Diagnostic counters
It is possible to load Diagnostic counters from System Configuration Tool for Device and Line diagnostics by specifying a VSO as STOOL/SYSCONF.VSO and Object as STATION_DIAGNOSTIC_COUNTERS (for station) or LINE_DIAGNOSTIC COUNTERS (for lines).

This tab has no specific configuration options.

20.10.2 Files related to SSS control dialog

This section describes the structure of the SSS control dialog. Files will be located under System Tools in /STOOL/SUPERV folder.

Table 240: SSS Control dialog related files

File	Functionality
SSS_CTRLDLG.VSO	Main Visual SCIL dialog. Contains most of the implementation.
SSS_CTRLDLG.SCL	Contains the help functions used and called from application specific ini files. Functions such as GET_NODE_LINE_NUMBERS, LIN_DIAG_PRELOAD and LIN_DIAG_POSTLOAD are implemented in this file and called from ini file with DO(READ_TEXT … statement. It is not recommended to customize this file in projects, but instead define a project specific SCL file which is called similarly from the ini-file.
SSS_CTRLDLG.INI	Default dialog configuration template included in the product. This file is copied to APL\SUPERV during SSS initialization.

20.11 Removing SSS application objects and files

The following steps should be performed, if there is need to clean the SSS information from the application.

1. Remove SSS routing objects:
   1.1. Select **Settings/System Self Supervision** to open System Configuration Tool. Then, change SSS to Disabled state, and set Remove supervision routing objects. Click **OK**.
   1.2. Save the active configuration, which will remove the supervision routing objects from the application that has been constructed by System Configuration Tool.
2. Remove SSS event and alarm process objects:
   2.1. These process objects are following the naming convention, SYS_'nnn'E or SYS_'nnn'A, where 'nnn' denotes the dynamically constructed part indicating the functional purpose.
   2.2. Open Object Navigator, select these process objects including E or A as suffix, and delete them.
3. Remove SSS command procedures:
   3.1. These command procedures are following the naming convention, SYS_*. The amount of the command procedures depends on what kind of SSS has been constructed into application.
   3.2. In the maximum, the following command procedures are included, and they should become deleted with Object Navigator:

SYS_NETSR, SYS_SYSER, SYS_STAMR, SYS_APLER, SYS_NETLMR, SYS_UPOER, SYS_SSSINI, SYS_SLCMR, SYS_PRIMR, SYS_OSER, SYS_NETMR, SYS_SSSREV, SYS_APLH1, SYS_APLH2, SYS_APLE1 and SYS_APLE2.

SYS_INIT_1 and SYS_NETD command procedures should not be deleted, because those belong to the configuration mechanism of PC-NET with System Configuration Tool.

4. Remove SSS configuration files:
   4.1. These files are located under the application SUPERV folder. Delete all the files included into the folder \sc\apl\'appl_name'\SUPERV, where 'appl_name' denotes the name of application, for example, MAIN.

20.12 Workplace X visualization for System Self Supervision

Workplace X has views for application, IED communication and network hardware (SNMP) supervision information. The content for the views is automatically generated based on the system and application configuration.

By default, the configuration changes are updated to the Workplace X system supervision models cyclically every 5 minutes. This is done by executing BPU_SSS_UPDATE_MODELS command procedure from a time channel (BPU_SSS_UPDATE_MODELS). If system configuration is updated by some project specific workflow, the command procedure can be triggered from there to have faster update of the models.

The following describes the required system supervision process objects that need to be available for the Workplace X system self-supervision functionality to work properly. The object identifier (OI) of the related supervision process object is used in the Workplace X supervision views to identify, for example, NOD, STA and SNMP objects.

20.12.1 Application

Node diagnostics process objects for SYS nodes where the applications are running. Process object LN = “SYS_NnnnnI” where nnnn is the node number presented with 4 digits and leading zeros. For example, “SYS_N0010I” for the node 10 status. The used IX is 10. Node diagnostic need to be enabled in the base system configuration.

Application state process object for each application configured in the base system.

For applications running as single (no HSB shadowing)

Process object LN = “SYS_AaaaaI” where aaaa is the local application number presented with 4 digits and leading zeros. For example, SYS_A0001I for the local application number 1. Index of the process object is 100 + local application number. For example, 101 for the local application number 1.

For applications running in HSB configuration

Process object LN = “SYS_AaaaaI” where aaaa is the local own application number presented with 4 digits and leading zeros. For example, “SYS_A0001I” for the local application number 1. Index of the process object is 100 * node number + 1000 + application number in the node. For example, 2001 for the application number 1 in node 10.

SSS_PO_IX = 100 * node + 1000 + SSS_PO_apl_number

20.12.2 IED Communication

Communication node diagnostics process objects for nodes where the STA/IED objects are configured. For example, PC-NET node or External OPC DA client node. Process object LN = “SYS_NnnnnI” where nnnn is the node number presented with 4 digits and leading zeros. For example, “SYS_N0003I” for the node 3 status. The used IX is 10. Node diagnostic shall be enabled in the base system configuration.

Process object LN for STA/IED connection status = “SYS_SssssI” where ssss is the STA number presented with 4 digits and leading zeros. The used IX is 210.

20.12.3 Network Hardware

Process objects for the SNMP devices as described in the chapter Device status via SNMP should be created. Also, the SNMP STA communication status process object as described in the previous IED communication section shall be available.

Section 21 User Account Management

This section describes the elements and functions of user management. The authorization level for User Management tool is System manager (5).

With the User Management Tool it is possible to:

1. Add and remove users
2. Define users properties and roles
3. Set users default role
4. Reset users password
5. Define Groups and Roles
6. Define Areas of Responsibility
7. Define user roles for Areas of Responsibility
8. Define Physical Access Point Roles
9. Define Password policy
10. Define Timeouts for Monitor Pro

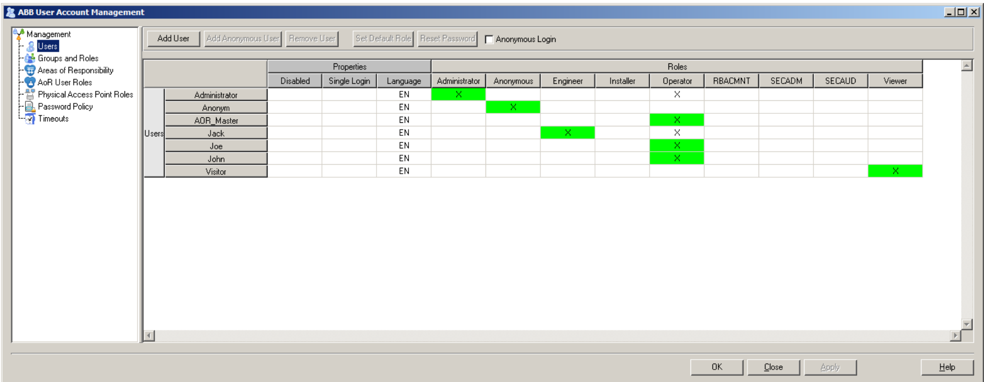


Figure 248: User Account Management tool

When Central Account Management (CAM) is in use, Groups and Roles and Timeouts are available in the tool. Only Groups can be added in the Groups and Roles leaf.

Right-click the Users, Groups and Roles, Areas of Responsibility, Physical Access Point Roles or Timeouts leaf to adjust the row header width of the table.

21.1 Windows Single Sign-on

Windows Single Sign-on can replace normal SYS600 user login. When Windows Single Sign-on is enabled, the user accessing SYS600 user interface is automatically attempted to be logged in using user's Windows credentials. Windows Single Sign-on can be enabled by setting SYS600 application attribute WS to 1 or True value.

Single Sign-on login is done by matching Windows user group names to SYS600 role names. The comparison is case insensitive. Both Windows Local and Remote Windows user groups are used. Feature works for computers joined to a domain or workgroup.

When Single Sign-on user group matching cannot be done or no Windows user groups can be matched to SYS600 roles, login defaults to the current active configuration and normal login dialog is shown.

21.2 Users

In the Users leaf it is possible to:

1. Add users
2. Remove users
3. Set user properties
4. Define users roles

21.2.1 Properties

21.2.1.1 Disabled

A user can be temporarily disabled. This means that user cannot log in to application while being disabled. Disabled user is shown with red color.

21.2.1.2 Single Login

Only one login with the same user credentials is accepted when Single Login is enabled. A second login attempt with the same user credentials will fail. Active concurrent user sessions are not forcefully ended when the user limitation is set during runtime.

The functionality can be activated or deactivated per user which means that one user might have the limitation set while another user does not have it set. The functionality cannot be activated for an Anonymous user

21.2.1.3 Language

The language, that is used for the user after login, can be selected from the available languages in the system.

By default English is available. The other languages can be added in the **Text Tool**.

21.2.2 Roles

21.2.2.1 Normal users

A user can have more than one role, but not an Anonymous role. The default role is shown as green. If the user has more than one role, the role is selected during the login. The access rights for the role are configured in the Groups and Roles leaf.

21.2.2.2 Anonymous user

An Anonymous user has only the Anonymous role and it has no password. If the anonymous user mode is enabled, an automatic login as the anonymous user is executed in the primary application on Monitor Pro.

21.2.3 Changing Properties and Roles

To change Properties or Roles of the user, double click the cell in the table. Click **OK** or **Apply** to save the changes.

21.2.4 Adding a user

To add a user click the **Add User** button. In Add User dialog define the following attributes:

1. User name. User name can be any valid file name.
2. Copy data from User. You can copy Properties and Roles settings from another user.
3. Copy Folders from user. You can copy Monitor Pro and Display Builder layout and configuration settings.
4. Password. Password must be given according to Password Policy settings.
5. User must change password at first login.

Figure 249: Add User

21.2.5 Adding an Anonymous user

To add a user, click the **Add Anonymous User** button. In the Add Anonymous User dialog define the following attributes:

1. User name. The user name can be any valid file name.
2. Only the language property can be changed for the anonymous user

The anonymous user mode is enabled by clicking the Anonymous Login checkbox.

21.2.6 Removing a user

To remove a user, select the user from row heading, click **Remove User** button and confirm operation.

21.2.7 Setting default role

To set default role for the user, select the user from row heading and click **Set Default Role** button. Save changes by clicking **OK** or **Apply**.

21.2.8 Resetting password

To reset the users password, select the user from row heading and click **Reset Password**.

21.3 Groups and Roles

In the Groups and Roles leaf define the access of the role to different functions, like Monitor Pro customization or Process controlling, by means of authorization groups and authorization level to the group.

For example, operators typically have an access level **1 - Control**, which means the Operator role should have this authorization level configured to each defined authorization group. If the authorization group dedicated to a certain function is not defined, the authorization level of the GENERAL group is used. Also, if the authorization group is defined, but the authorization level is empty, the GENERAL group is used for authority checking. For more information about authorization groups, see Section 24.2.1.

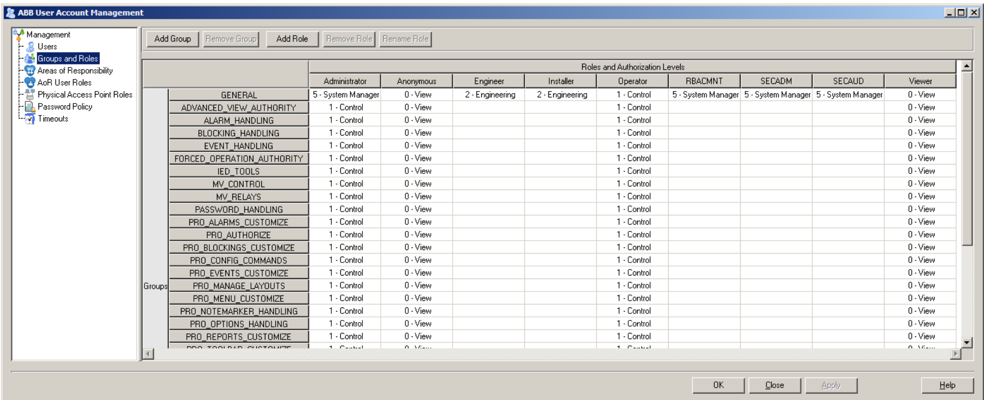


Figure 250: Groups and Roles

When the role has authorization level 5 in GENERAL group it is considered as a System Manager role.

In the Power Process Library applications, the standard functions in the Process Displays can be freely grouped into authorization groups. This means that there can be different authorization levels for different apparatus. Users can also be defined to have different authorization levels for different tools, substations, and so on.

When the Area of Responsibility functionality is used, the authorization group MV_CONTROL must be added and the authorization level must be explicitly defined for the used AoR roles. The authority level for the area is checked from this configuration.

There are the following default roles that cannot be removed or renamed.

Table 241: Roles

Role	Meaning	Remarks
Viewer	Role to be assigned to users needing access permissions only allowing read-only use of the product.	IEC 62351-8 standard role.
Operator	Role to be assigned to users needing access permissions allowing to operate the primary equipment via the product.	IEC 62351-8 standard role.
Engineer	Role to be assigned to users needing access permissions allowing the modification and installation of applications configuration parameters	IEC 62351-8 standard role.
Installer	Role to be assigned to users needing access permissions allowing the installation of software and configuration updates, but with no rights to make own changes	IEC 62351-8 standard role. At the moment no dedicated functionality behind this role.
SECADM	Role to be assigned to users needing full access permissions for almost all activities specially user accounts administrative activities, but excluding auditing	IEC 62351-8 standard role. At the moment no dedicated functionality behind this role.
SECAUD	Role to be assigned to auditing officers needing access permissions to view security logs and enabling them to generate security reports.	IEC 62351-8 standard role. At the moment no dedicated functionality behind this role.
RBACMNT	Role to be assigned to users needing permissions to change role-to-right(permissions) assignments	IEC 62351-8 standard role. At the moment no dedicated functionality behind this role.
Administrator	This is a super-set of the existing IEC 62351-8 admin roles	ABB vendor role.
Anonymous	Role automatically assigned to Anonymous user	ABB vendor role.

The access rights of the role can be fine-tuned by setting the authorization level for the groups. The following access levels can be defined:

Table 242: Authorization levels

Authorization level	Meaning
- 1 - Reserved	Reserved for future.
0 - View	The operator is allowed to view the functions, but not to make control operations.
1 - Control	The operator is allowed to make control operations, for example to control switching devices with limited rights.
2 - Engineering	All rights are granted, excluding user management, which can be performed only by the system manager.
Table continues on next page	

3 - Reserved	Reserved for future.
4 - Reserved	Reserved for future.
5 - System Manager	The user with this authority level is the system manager. All rights are granted, including the rights to add and remove users and authorization groups. The user who first logs into an application is the system manager. Thus, each application has only one system manager.

21.3.1 Adding a group

To add an authorization group, click **Add Group** and select predefined group or type in your own group.

21.3.2 Removing a group

To remove a group, select group from row heading and click **Remove Group** and confirm operation. GENERAL group cannot be removed.

21.3.3 Adding a role

To add a role click **Add Role**. Copy data from Role copies the authorization levels from selected role to the new one.

21.3.4 Removing a role

To remove a role, select the role from column heading, click **Remove Role** and confirm operation.

21.3.5 Renaming a role

To rename a role, select the role from column heading, click **Rename Role**, type in the new role name and confirm operation.

21.3.6 Authorization groups used by SYS600 Workplace X

SYS600 Workplace X uses the following authorization groups

- ALARM_HANDLING
- PRO_ALARMS_CUSTOMIZE
- PRO_EVENTS_CUSTOMIZE
- PRO_NOTEMARKER_HANDLING
- PRO_OPTIONS_HANDLING
- PRO_TOPOLOGY_HANDLING
- PRO_ZOOM_AREA_HANDLING
- TREND_HANDLING

In SYS600 Workplace X each user can change own password. **Change password** dialog does not follow the authority level defined by authorization group PASSWORD_HANDLING.

21.4 Areas of Responsibility

Areas of Responsibility (AoR) can be used to define the role of a user to a specific area. It works based on object identifiers (the OI attribute of the process objects). The AoR will be given a name and a list of objects belonging to the AoR. All objects containing the configured OI string will belong to the AoR.

Exclusive Access Rights (EAR) can be used to allow only one user at a time to have rights for the area. If the EAR is enabled, it is not possible to create or modify AoRs that may become overlapping. Overlapping means that the same object (OI) belongs to more than one AoR.

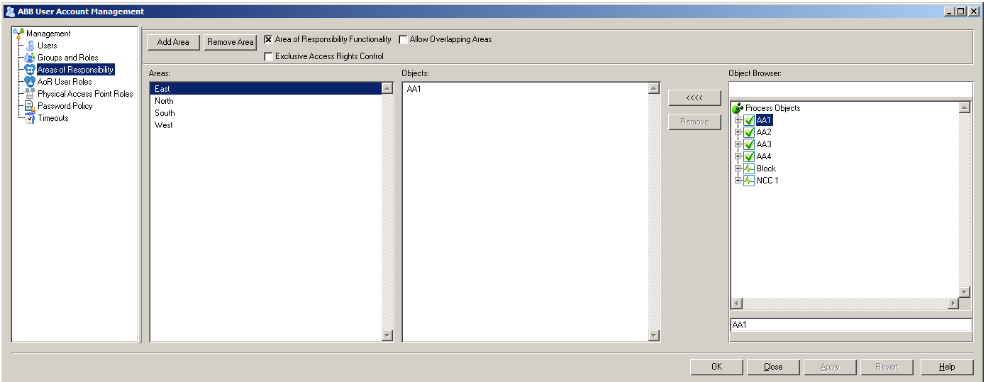


Figure 251: Areas of Responsibility

SYS600 logs the AoR events using a predefined event channel name. The event channel AOR_EVENT is activated when a significant state change takes place in an AoR. For detailed information about AOR_EVENT event channel parameters, see SYS600 Application Objects document.

21.4.1 Adding an area

To add an area, click **Add Area**, type in Area Name and click **Add**.

21.4.2 Adding objects to an area

To add an object to the area, select the area, select the node from the Process Object tree and click **<<<<**. The child nodes will automatically belong to the area. You can filter areas by clicking the text box under the Object Browser and defining the condition.

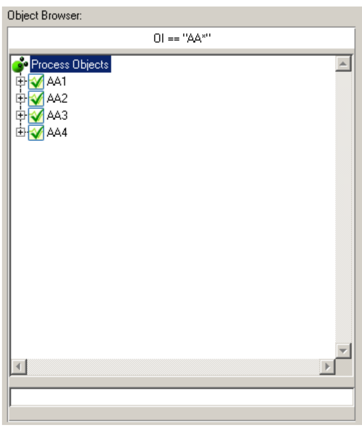


Figure 252: Filtering in Object Browser

21.4.3 Removing an area

To remove an area, select the area and click **Remove Area**.

21.4.4 AoR User Roles

A user’s rights to a certain AoR are configured by assigning an AoR role for the user for the specific AoR.

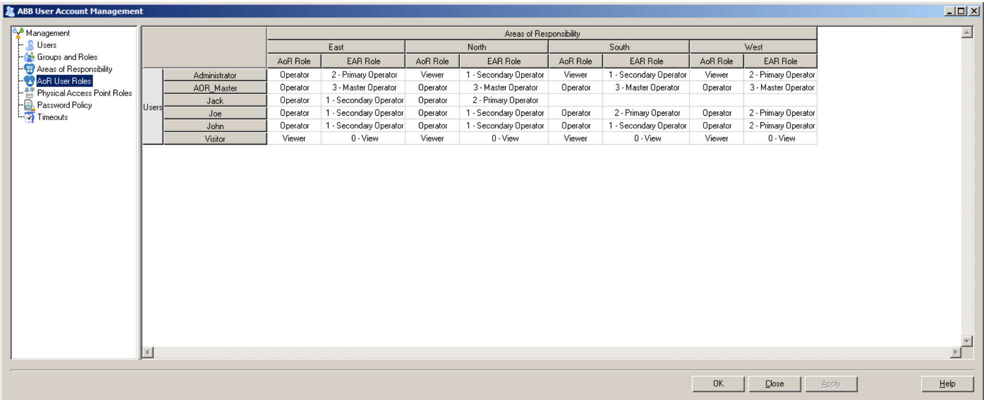


Figure 253: AoR User Roles

A user's AoR Role can be any of the roles defined in the **Groups and Roles** leaf except the Anonymous role.

It is possible to define the AoR functionality and use the configuration without enabling the EAR mode. In this mode all users having an AoR role are treated as Operators or Viewers. Operators are users having at least an authorization level 1 - Control (1-5) in a predefined authorization group. All other users are Viewers. The AoR role defines the SYS600 role used to resolve the authorization level. By default, SYS600 uses the authorization group ‘MV_CONTROL’. The authorization level check for the AoR role does not fallback to the GENERAL authorization group. It will require that the specified authorization group exists.

Figure 254 shows an example configuration for the AoR functionality mode without the EAR mode activation. Joe has a SYS600 Operator role in the AoR East. The ‘MV_CONTROL’ authorization group defines the authorization level 1 – Control for that role. The configuration makes Joe an Operator in the AoR East.

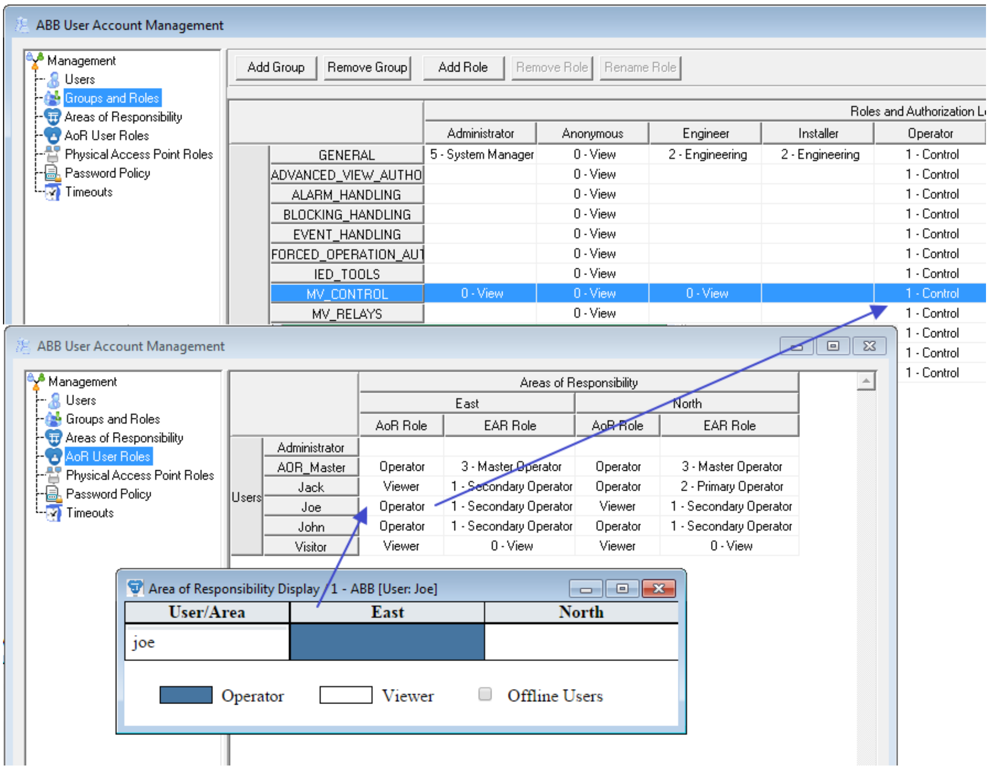


Figure 254: AoR User Interface

The EAR Role is an extra configuration attribute for the user per AoR. It specifies whether the user role is Viewer, Secondary Operator, Primary Operator or Master Operator. Based on the EAR role, the user has certain rights related to EAR operations. For example, a Viewer cannot have the EAR for an AoR (only view), a Secondary Operator and Primary Operator can have the EAR, but the Primary Operator has a priority for the EAR. A master operator can force the EAR to himself, or forcefully assign it to any user having a sufficient EAR role.

The Single Login location feature for a user is forced when the EAR mode is active. This means that the user can successfully login from only one physical location. A login attempt from another physical location will fail.

The EAR role is a role definition designed only to be used in the EAR mode. This role defines the order in which SYS600 assigns uncontrolled AoRs. A users EAR role in any area can be one of the following:

Table 243: EAR role

Authorization level	Meaning
0 - View	Can view AoR and EAR data (runtime information).
1 - Secondary Operator	Can read EAR information (e.g. the user having the EAR for the AoR). The secondary operator for an AoR is allowed to have the EAR.
2 - Primary Operator	Can read EAR information (e.g. the user having the EAR for the AoR). The primary operator for an AoR is allowed to and is the preferred operator to have the EAR.
3 - Master Operator	Can read and write EAR information. The operator can force the EAR to him/herself or assign it to someone else who is allowed to have the EAR for the AoR.

SYS600 assigns uncontrolled AORs automatically as long as a user with an EAR Role is logged in to the system. The EAR role is automatically assigned to the user who has the highest EAR user role for the AoR. If several users with an equal EAR user role are found, the EAR will be assigned to the first one found (in no particular order). The order in which the automatic assignment is done:

1. Primary operator
2. Secondary operator
3. Master operator

21.4.5 Alarm Acknowledgement Rights for Workplace X

When AoR is configured and enabled, it is used for Workplace X to detect if the user is authorized to acknowledge certain alarm. Typically, if user has AoR role operator for the area, the user is authorized to acknowledge alarms related to the area. If user has AoR role viewer for the area, the user is not authorized to acknowledge alarms related to the area. Authorization is based on the ALARM_HANDLING authorization group level value. If user’s AoR role has the ALARM_HANDLING level > 0 for the area where the alarm belongs, the user is authorized to acknowledge the alarm. This allows configuring that certain users can acknowledge alarms, for example, of certain substations, voltage level or other area specified using the object identifier.

21.5 Physical Access Point Roles

You can configure the allowed login roles for computers. Computers are identified during the login using the computer name property.

If a configuration for the computer is found, the available login roles are limited based on the configuration.

A login from the local computer (SYS600 Server) is not affected by the configuration. All defined user roles are always available. The local computer definition cannot be removed or modified.

It is possible to specify allowed roles for unknown computers by defining a configuration for an 'UNKNOWN' computer name. This configuration is used for computers which are not found in the configuration list. The 'UNKNOWN' definition cannot be removed.

The property **All User Roles** means that the user role configuration defined in the **Users** leaf will override the role defined in the **Physical Access Point** leaf.

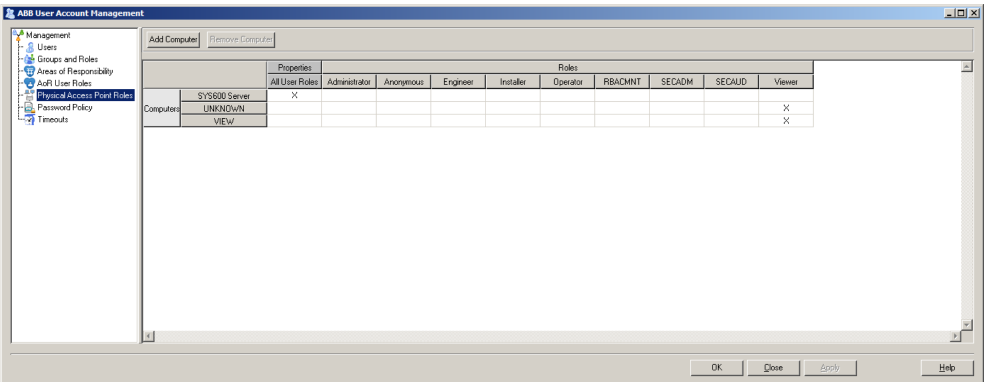


Figure 255: Physical Access Point Roles

21.6 Password Policy

By enabling the password policy, all new users of the application have to have a password that complies with the password policy settings. The count and type of characters allowed in the passwords can be defined. By default, password policy for new SYS600 application is enabled and default values shown in Figure 256 are used.

If the user name is at least three characters long, the password may not contain the entire user name. The check is not case sensitive.

If the password policy is changed and the password of the existing user does not comply with the new rules, the user's password needs to be reset.

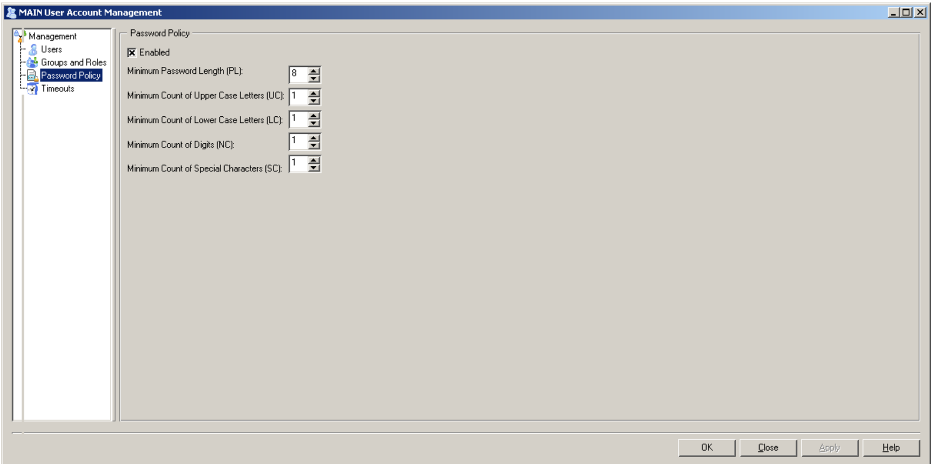


Figure 256: Password policy

21.6.1 Defining password policy

To define the password policy enable policy, set the policy settings and click **OK** or **Apply**.

Password policy cannot be saved if there is policy violation in definition. For example minimum count of upper case letters is higher than the minimum password length.

21.7 Timeouts

You can define timeouts for a role. These timeouts are used in Monitor Pro. 0 means that timeout is not in use.

Monitor Pro Session Timeout starts to run after login. If Monitor Pro Session Timeout is defined, user is logged out after the specified time has elapsed.

If Monitor Pro inactivity time is defined, user is logged out if there are no user input, like cursor movement, within the specified time.

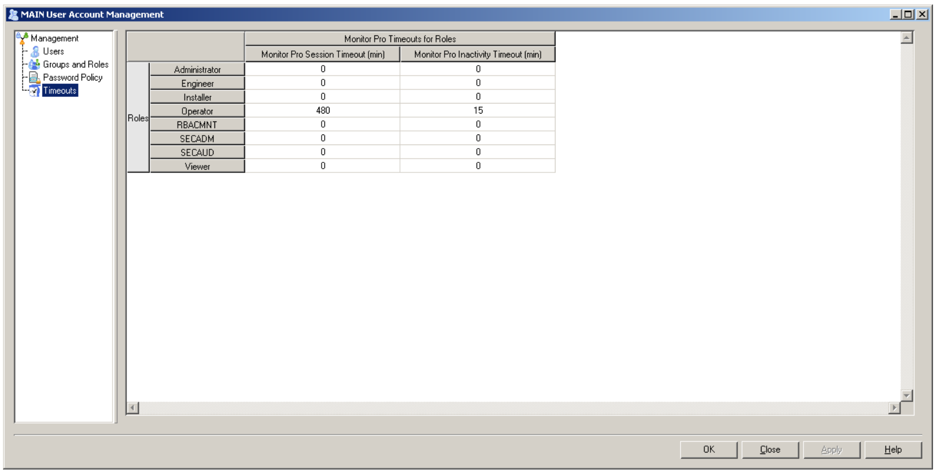


Figure 257: Timeouts

21.7.1 Defining timeouts

Define the timeouts for different roles in the table and click **OK** or **Apply**.

Section 22 SYS600 Historian

SYS600 Historian can be used for creating measurement reports, charting, and data analysis. SYS600 Historian works well with large applications and high load. Arbitrary many signals can be logged to SYS600 Historian.

Any binary input (BI), binary output (BO), double binary input (DB), digital input (DI), digital output (DO), analog input (AI), analog output (AO), and pulse counter (PC) signal or data objects with numerical value can be connected to SYS600 Historian. When a signal is connected to SYS600 Historian, the values updated to the signal are automatically sent to the SYS600 Historian. The data collected by SYS600 Historian can be further processed using aggregate functions. For example, averages, maximums, and minimums can be calculated.

Many SYS600 systems can log to same SYS600 Historian and a single SYS600 system can log to multiple SYS600 Historians. SYS600 systems can be either stand alone or HSB systems. SYS600 Historian installation is referenced using a Database Location. Location is an URL with following the form wss://<hostname>/history.

Each logged process object and data object has a Tag and a Variable in SYS600 Historian. These are created automatically when Logging objects are created. Tags and Variables contain configuration information about the signal. Relevant fields are filled and updated automatically.

22.1 Configuring Logging Objects

The configuration is done using logging objects. Logging objects are application objects. There are three different types of logging objects: DATABASE, HISTORY, and OBJECT. A DATABASE objects describes a SYS600 History installation. A HISTORY object describes the aggregates that are collected from logged signal. The last type OBJECT is used for associate signals to databases.

To configure logging, use the following procedure:

1. Install SYS600 Historian.
2. Create a DATABASE object using Object Navigator (**Logging Profile Objects/New**).
   2.1. Enter Database Location as Database Address, Username, and Password.
   2.2. Set In Use (IU).
   2.3. Click **Apply**.
   2.4. If connection parameters are valid, the Object Status (OS) should be CONNECTED. If the Object Status is DISCONNECTED, Database Address, Username, or Password is mistyped or the SYS600 cannot connect to the SYS600 Historian.
   2.5. Click **OK**.
3. Create a HISTORY object.
   3.1. List the used aggregate functions. The available aggregate can be checked from the SYS600 Historian using Vtrin user interface (**Maintenance/System/HistoryLists/History Collection Templates**).
   3.2. Set In Use (IU).
   3.3. Click **OK**.
4. Create an OBJECT object.

4.1. Connect the signals to logging profiles. Click **Add.../Select...** and choose the signals to be connected.
4.2. Set In Use (IU).
4.3. Click **OK**.

Now the Tags and Variables should be created automatically and the values should be automatically logged to the SYS600 Historian. The instructions above used the default settings for all objects. For additional configuration parameters, refer to the Application Object manual.

For pulse counter objects, two aggregate variables are created. The name of the Variable for pulse value in engineering units is prefixed with E and the name of the Variable for time derivation is prefixed with D.

The bit count (BC) and pulse scaling (SC) attributes must be properly defined for pulse counter signals to work correctly with SYS600 Historian.

SYS600 generates a name for each Tag. If there already is a tag that has a name that conflicts with the generated name, the tag gets a new name. The SYS600 generated name is a proposed name. The actual name of the tag is stored in the GN attribute of the process object or the data object.

Some attributes are defined by patterns. Patterns are defined in a DATABASE logging object. Patterns are statements of the SCIL Data Derivation Language (SDDL). The description of the SDDL can be found in Programming Language SCIL manual.

Table 244: The automatically filled fields of a Tag

Field Name	Description	Source
Name	The unique name of the Tag.	Based on the proposed name attribute.
External Name	Name of the tag in an external system.	SCIL name of the object. E.g. ESTH01_Q0:P10.
Proposed Name	Name of the Tag generated by SYS600.	Generated as defined by the NP or ND attribute in DATABASE object.
Equipment Path	A path used for defining the Tags location in the equipment model. Comparable to the OI of the process objects.	Generated as defined by the EP or ED attribute in DATABASE object.
Description	A description of the Tag.	Generated as defined by the DP or DD attribute in DATABASE object.
Type	The data type of the Tag.	Defined by the type of the logged object.
Binary Text	The source for textual representation of binary values.	The event handling object of the logged process object. Only used for BI and DB objects.
Creator	Information about the creation of the Tag.	The name of the SYS600 application.
History Collection Templates	List of aggregate collection.	Defined in the HISTORY object.

Table 245: The automatically filled fields of a Variable

Field Name	Description	Source
Name	The unique name of the variable.	Based on the proposed name attribute.
Process Path	A path used for defining the Tags location in the equipment model. Comparable to the OI of the process objects.	Generated as defined by the EP or ED attribute in DATABASE object.
Section	Information about the creation of the tag.	The name of the SYS600 application.
Unit	The name of the engineering unit.	The ST attribute of the process object.
Description	A description of the Tag.	Generated as defined by the DP or DD attribute in DATABASE object.
Value Type	The data type of the Tag.	Defined by the type of the logged object.
Binary Text	The source for textual representation of binary values.	The event handling object of the logged process object. Only used for BI and DB objects.

The automatically filled fields in Variables and Tags are described in Table 244 and Table 245.

When the signals are unconnected from a logging object, the corresponding tags are not removed from the SYS600 Historian.

Logging objects cannot be removed if there are connections to other objects. To remove a logging object with profile OBJECT, open it using Object Navigator. Select all connected objects and disconnect them. Then, the logging object can be removed from the Object Navigator main window (**Main menu/Edit/Delete**).

To remove a logging object with profile DATABASE or HISTORY, the user must first either delete all connected logging objects with profile OBJECT or disconnect them. To disconnect, open the logging object with profile OBJECT and remove the corresponding row from the storage attribute (ST).

22.2 Locating Objects in SYS600 Historian

There are several ways to find objects in SYS600 Historian. Each signal has a name, which is generated with name pattern. The SYS600 Historian requires that each signal has a unique name. If pattern generates a duplicate name, the SYS600 Historian modifies the name so that it will be unique. The actual name of the signal in SYS600 Historian is stored in signals in logging name attribute (GN).

Another way to locate signals is by using the plant model. Plant model is build based on the equipment path. The equipment path is also generated with pattern. With default settings, the equipment path for process objects (P) is built from the OI structure. For data objects, the equipment path is empty, hence data objects are not present in the plant model. It is possible to configure plant model for data objects. If the equipment path pattern for data object (DD) is set to FX, the database objects will use the FX attribute as their equipment path. Then the path in plant model can be set as dot separated string list to the FX attribute of each data object. E.g. if the data object has an FX with value Rivers.110kv.Q0, it will be placed in the plant model under Rivers, 110kv, and Q0 branches.

Patterns are stored to the DATABASE object. There are separate patterns for process objects (P) and data objects (D).

22.3 Monitoring

The system has several diagnostics attributes. Each DATABASE object has an Object Status (OS) attribute. This attribute contains the state of the connection between SYS600 and SYS600 Historian. DATABASE objects also contain a diagnostics counter attribute (DC). Object statuses and diagnostics counters are listed in Application Object manual.

SYS600 also generates application events when the status of the connection is changed. The events are emitted to the APL_EVENT channel. The list of emitted events can be found in the Application Object manual.

22.4 Viewing and Processing History

The data logged to SYS600 Historian resides on the system where SYS600 Historian is installed. The system can be accessed with the Vtrin client application. To open the Vtrin client, open https://<SYS600 Historian Host>/vtrin.

For further information on viewing and processing the data, refer to SYS600 Historian documentation.

Section 23 Integrating PCM600

The PCM600 relay symbols can be added to a Process Display from the Object Browser of the Display Builder.

Object Browser displays the relay symbols if PCM600 is installed and the relay objects exist in the PCM600 projects. For more information about installing PCM600, see the PCM600 Installation and Commissioning Manual.

To install the PCM600 relay symbols:

1. Install both SYS600 and PCM600.
2. Select **Actions/Object Browser** to open the Object Browser of the Display Builder.
3. Select **PCM600 Projects** from the Application drop-down menu.
4. Click **Select** to display the relay tools in the Object Browser tree.
5. Drag IED to the drawing area to create a symbol in a Process Display.

For more information about using Object Browser, see SYS600 Process Display Design.

If the user management is in use in PCM600, the user name and password must be the same in both SYS600 and PCM600 in order to be able to open the relay symbols in Object Browser. If the user names and passwords differ from each other, installing the relay symbols and opening the relay tools is not possible in Monitor Pro and an error message is displayed.

23.1 Opening PCM600 relay tool

The PCM600 relay tools can be opened in a Process Display in the following way:

1. Open a Process Display.
2. Click a symbol. The IED Tools dialog is displayed.
3. Double-click a tool to open it.

Table 246: Authorization group related to PCM tool launcher

Authorization Group	Level	Enabled Action
IED_TOOLS	>= 2, Engineering	Access granted

For more information about using the PCM600 tools, see PCM600 Help.

23.2 Controlling PCM600 by using SCIL

The following SCIL commands help the user to manage the PCM600 client instances:

Usage: pcmcmd <command> <command parameters>

23.2.1 Manage the PCM600 instance commands

These commands are used to manage PCM client instances.

/ListInstances

/Start <instanceName> [<user> [<password>]]

/Stop <instanceName>

/IsStarted <instanceName>

23.2.2 The PCM600 instance commands

These commands operate on running PCM client instances.

/ListProjects <instanceName>

/OpenProject <instanceName> <projectName>

/OpenDefProject <instanceName>

/IsProjectOpen <instanceName>

/CloseProject <instanceName>

/ListTools <instanceName> <objectPath>

/OpenTool <instanceName> <objectPath> <toolName>

/ListTasks <instanceName>

/RunTask <instanceName> <taskName>

/ListObjects <instanceName><structureName> [<parentObjectId>]

The return (exit code) values:

Success: 0

Invalid argument: 1

Failure: HRESULT/Win32 error code

Is* functions:

False: 0

True: -1

23.2.3 Example

```
set inst=an_instance_name

set user=a_user_name

set pwd=user_password_in_PCM

set project=a_project_name

;Start a new instance as a certain user

pcmcmd /start %inst% %user% %pwd%

;list all projects for the new instance

pcmcmd /ListProjects %inst%
```

```
;open a given project

pcmcmd /OpenProject %inst% %project%

;check if the project is open

pcmcmd /IsProjectOpen %inst%

;list all tools for a given object

pcmcmd /ListTools %inst% COMMUNICATION:CN\61850\SNW\REC

;list all tasks

pcmcmd /ListTasks %inst%

;open a given tool (SCLImport) for a given object

pcmcmd /OpenTool %inst% COMMUNICATION:CN\61850\SNW\REC SCLImport

;open another tool (PST) for the same object

pcmcmd /OpenTool %inst% COMMUNICATION:CN\61850\SNW\REC PST

;list all objects in the COMM structure root

pcmcmd /ListObjects %inst% COMMUNICATION

;list all objects in the PLANT structure root

pcmcmd /ListObjects %inst% PLANT

set object_id=<select an objectId from the listed objects

;list all child objects of the given object in the PLANT structure

pcmcmd /ListObjects %inst% PLANT %objectId%
```

Section 24 Authorization

User names are associated with a certain user profile that restricts the user's access rights, for example to access Process Displays. The authorization levels are defined in Table 247.

Table 247: Authorization levels

Authorization level	Description
0 - View	The operator is allowed to view the functions, but not to make control operations.
1 - Control	The operator is allowed to make control operations, for example to control switching devices with limited rights.
2 - Engineering	All rights are granted, excluding user management, which can be performed only by the system manager.
5 - System Management	The user with this authority level is the system manager. All rights are granted, including the rights to add and remove users and authorization groups. The user who first logs into an application is the system manager. Thus, each application has only one system manager.

The system administrator can restrict a user's access rights either by defining an authorization level for each object in a Process Display or by defining shortcuts for Process Displays.

It is possible to define the authorization level for each object in a Process Display. The authorization level for the object is defined in Display Builder. Right-click the object and select **Tool Launcher**. In the **Tool Launcher** dialog, select the **Required** check box and define the authorization level.

The user's access rights can be restricted also by defining the shortcuts for the Process Displays. The system manager can define that the user can access Process Displays only through the shortcuts and prevent the use of the menu commands. For example, the user cannot open a Process Display by selecting **Main/Open**. The shortcuts are displayed on the toolbar of each Process Display. For more information about defining shortcuts, see Section 18.16.

The user is automatically logged off if Monitor Pro is not used when a certain time has elapsed since the last login.

If the user has the access rights for the appropriate authorization level, the user can configure the authorization group and the level which is needed for using certain menu commands and toolbar buttons. The menu commands and toolbar buttons are disabled if the user is not authorized to use them. The configurable authorization level definitions override the default authorization group in Monitor Pro. If some menu commands are disabled due to the authorization group, default values in the user management. It is possible to override the default values with the configurable authorization level definitions.

The authorization levels can be managed in the customization mode in Monitor Pro. For example, right-click the menu or the toolbar button and select **Authorize**. For more information about the customization mode, see Section 18.3.

If a standard function does not have an authorization group defined, or the group is not included in the authorization definitions, a General authorization level from a predefined group is used. The authorization group GENERAL cannot be removed.

System Tools are managed from the Tools authorization group. To dedicate the system tools to certain users, define the Tools authorization group.

Only the system manager can add and remove users and grant access rights to the users. The password is selected and can be changed by the user exclusively. Password is changed in the **Change Password** dialog.

24.1 Default authorization levels

The default authorization levels for menu commands and toolbar buttons are shown in Table 248.

Table 248: Default authorization levels

Action	Authorization group	Authorization level
Tool Manager	TOOLS (GENERAL)	2 - Engineering
Display Builder	TOOLS (GENERAL)	2 - Engineering
MicroSCADA Control Panel	GENERAL	2 - Engineering
Alarm acking from the alarm row, context menu in the alarm control, menu command, toolbar button	ALARM_HANDLING (GENERAL)	1 - Control
Clear Current Trend Log	TREND_HANDLING (GENERAL)	2 - Engineering
Adding comments to events	EVENT_HANDLING	1 - Control

24.2 Authorization Groups

It is not possible to define the authorization level for a menu. Therefore, set the authorization level for the menu commands and submenus individually by right-clicking the command in the customization mode, see Figure 258 and Figure 259. The authorization groups and levels can be managed through User Management. For more information about using User Management, see the SYS600 Operation Manual.

It should be noted that Display specific items (Process, Alarm, Event, Blocking, Trends, Measurement Reports) from which Displays are opened are not possible to be authorized. This is due to the fact that it cannot be blocked from showing the actual Display, but the actions in Displays can be affected by authorization groups. For example, for Event Display there are EVENT_HANDLING and PRO_EVENTS_CUSTOMIZE authorization groups that can be used to change the behaviour of the Display from authorization point of view.

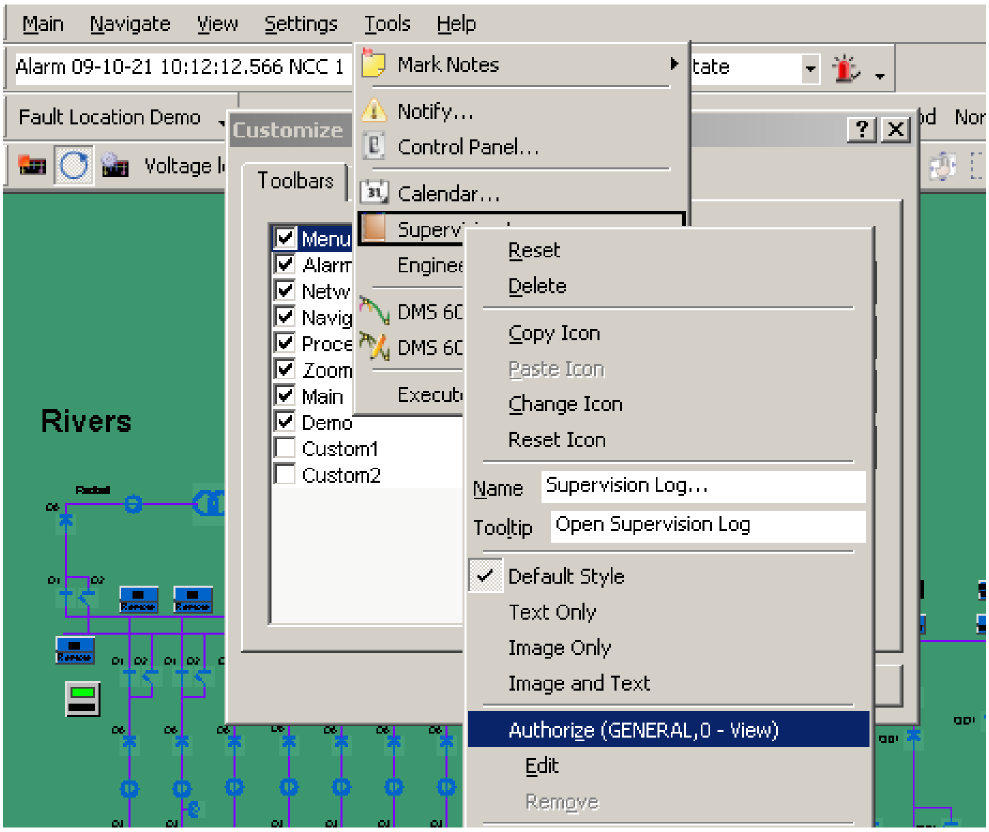


Figure 258: Setting the authorization level for the menu command

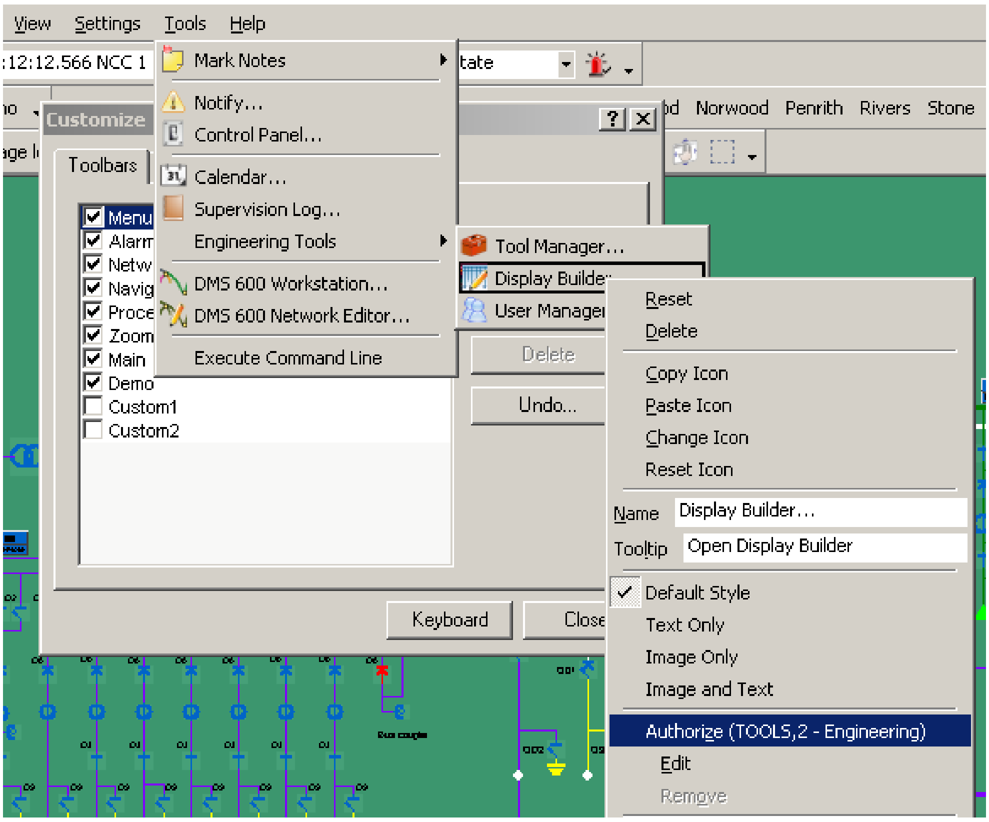


Figure 259: Setting the authorization level for the submenu command

24.2.1 Customization authorization groups

The following authorization groups define the authorization levels for customizing menus, toolbars and different displays.

24.2.1.1 PRO_MENU_CUSTOMIZE

Defines the required authorization level for customizing menus. The following operations are allowed for the PRO_MENU_CUSTOMIZE group:

- Using menu shortcut keys for copying and moving menu commands without opening the **Customize** dialog.
  - To move a menu command, press down the ALT key and drag the menu command between menus.
  - To copy a menu command, press down the CTRL key and drag the menu command between menus.
- Using the menu commands in the **Customize** dialog.
- Using the following menu operations:
  - **Tools/Set Font**.
  - **Tools/Reset/Reset Layout**.
  - **Tools/Reset/Reset Localizations**.

If the current authorization group is not found, the authorization group GENERAL is used. To customize menus, the authorization level 1 (Control) is required.

24.2.1.2 PRO_TOOLBAR_CUSTOMIZE

Defines the required authorization level for customizing toolbars. The following operations are allowed for the PRO_TOOLBAR_CUSTOMIZE group:

- Using toolbar shortcut keys for copying and moving toolbar buttons without opening the **Customize** dialog.
  - To move the toolbar button, press down the ALT key and drag the toolbar button between toolbars.
  - To copy a menu command, press down the CTRL key and drag the menu command between menus.
- Using the menu commands in the **Customize** dialog
- Locking and unlocking toolbars (**View/Lock Toolbars**).
- Hiding and showing toolbars.
  - Right-click a toolbar and select or clear a checked command from the context menu.
- Using the following menu operations:
  - **Tools/Set Font**.
  - **Tools/Reset/Reset Toolbars**.
  - **Tools/Reset/Reset Layout**.
  - **Tools/Reset/Reset Localizations**.

If the current authorization group is not found, the group GENERAL is used. To customize toolbars, the authorization level 1 (Control) is required.

24.2.1.3 PRO_ALARMS_CUSTOMIZE

Defines the required authorization level for customizing Alarm Displays.

If the current authorization group is not found, the authorization group GENERAL is used. The authorization level information for Alarm Displays is listed as follows:

Authorization levels 0 (View) and 1 (Control)

- Read-only access rights for application specific settings
- Read-only access rights for layout specific settings
- Read-only access rights for color specific settings
- Read/write access rights for user specific settings
- Read/write access rights for user specific layout settings
- Read/write access rights for user specific color settings

Authorization levels 2 (Engineering) and 5 (System Management)

- Read/write access rights for application specific settings
- Read/write access rights for layout specific settings
- Read/write access rights for color specific settings
- Read/write access rights for user specific settings
- Read/write access rights for user specific layout settings
- Read/write access rights for user specific color settings

24.2.1.4 PRO_EVENTS_CUSTOMIZE

Defines the required authorization level for customizing Event Display.

If the current authorization group is not found, the authorization group GENERAL is used. The authorization level information for Event Displays is listed as follows:

Authorization levels 0 (View) and 1 (Control)

- Read-only access rights for application specific settings
- Read-only access rights for layout specific settings
- Read-only access rights for color specific settings
- Read/write access rights for user specific settings
- Read/write access rights for user specific layout settings
- Read/write access rights for user specific color settings

Authorization levels 2 (Engineering) and 5 (System Management)

- Read/write access rights for application specific settings
- Read/write access rights for layout specific settings
- Read/write access rights for color specific settings
- Read/write access rights for user specific settings
- Read/write access rights for user specific layout settings
- Read/write access rights for user specific color settings

24.2.1.5 PRO_BLOCKINGS_CUSTOMIZE

Defines the required authorization level required for customizing Blocking Display.

If the current authorization group is not found, the authorization group GENERAL is used. The authorization level information for Blocking Displays is listed as follows:

Authorization levels 0 (View) and 1 (Control)

- Read-only access rights for application specific settings
- Read-only access rights for color specific settings
- Read/write access rights for user specific settings
- Read/write access rights for user specific color settings

Authorization levels 2 (Engineering) and 5 (System Management)

- Read/write access rights for application specific settings
- Read/write access rights for color specific settings
- Read/write access rights for user specific settings
- Read/write access rights for user specific color settings

24.2.1.6 PRO_TRENDS_CUSTOMIZE

Defines the required authorization level for customizing Trends. The following operations are allowed for the PRO_TRENDS_CUSTOMIZE group:

- Customizing the Trend display in a graphical view mode by using the **Graph Properties** dialog.
    - Right-click the display and select **Graph Properties** to open the dialog.

If the current authorization group is not found, the authorization group GENERAL is used. To customize Trends, the authorization level 1 (Control) is required.

24.2.1.7 PRO_REPORTS_CUSTOMIZE

Defines the required authorization level for customizing Measurement Reports. The following operations are allowed for the group PRO_REPORTS_CUSTOMIZE:

- Customizing the Measurement Reports display in a graphical view mode by using the **Graph Properties** dialog.
    - Right-click the display and select **Graph Properties** to open the dialog.

If the current authorization group is not found, the authorization group GENERAL is used. To customize Measurement Reports, the authorization level 1 (Control) is required.

24.2.2 Control authorization groups

The following authorization groups defines the authorization levels for controlling devices.

24.2.2.1 ADVANCED_VIEW_AUTHORITY

Used to lock the advanced view for not authorized user.

24.2.2.2 MV_CONTROL

MV_CONTROL is the default authorization group for Power Process library control dialogs. Authorization group can be configured in Standard Function Tool when installing objects.

24.2.2.3 FORCED_OPERATION_AUTHORITY

Define the required authorization level for using the forced operation function in Power Process library control dialogs.

24.2.2.4 SIMULATION_HANDLING

Block unauthorised access to the simulation function in Control Dialog.

24.2.2.5 SUBSTITUTION_HANDLING

Block unauthorised access to the substitution function in Control Dialog. Substitution is available only with IEC 61850 protocol.

24.2.3 Configure authorization groups

The following authorization groups define the authorization levels for configuring for example measurement reports, trends, alarms, events and blockings.

24.2.3.1 ALARM_HANDLING

Defines the required authorization level for handling alarms.

If the current authorization group is not found, the authorization group GENERAL is used. The authorization level information for controlling devices is listed as follows:

Authorization level 0 (View)

- Read-only access rights for application specific filter settings
- Read/write access rights for user specific filter settings

Authorization level 1 (Control)

- Read-only access rights for application specific filter settings
- Read/write access rights for user specific filter settings
- Read/write access rights for acknowledgement

Authorization levels 2 (Engineering) and 5 (System Management)

- Read/write access rights for application specific filter settings
- Read/write access rights for user specific filter settings
- Read/write access rights for acknowledgement

24.2.3.2 BLOCKING_HANDLING

Defines the required authorization level for handling blockings.

If the current authorization group is not found, the authorization group GENERAL is used. The authorization level information for controlling devices is listed as follows:

Authorization level 0 (View)

- It is not possible to change the blocking state.

Authorization level 1 (Control)

- It is possible to change the blocking state.

Authorization levels 2 (Engineering) and 5 (System Management)

- It is possible to change the blocking state.

24.2.3.3 EVENT_HANDLING

Defines the required authorization level for handling events.

If the current authorization group is not found, the authorization group GENERAL is used. The authorization level information for controlling devices is listed as follows:

Authorization level 0 (View)

- Read-only access rights for application specific filter settings
- Read/write access rights for user specific filter settings
- Read-only access rights for event commenting

Authorization level 1 (Control)

- Read-only access rights for application specific filter settings
- Read/write access rights for user specific filter settings
- Read/write access rights for event commenting

Authorization levels 2 (Engineering) and 5 (System Management)

- Read/write access rights for application specific filter settings
- Read/write access rights for user specific filter settings
- Read/write access rights for event commenting

24.2.3.4 GENERAL

The default authorization group. All the tools and standard picture functions configured to the group will automatically start using the definitions of the GENERAL group.

24.2.3.5 MV_RELAYS

Used in LIB5xx based relay tools to prevent e.g. unauthorised actions like write/change parameter.

24.2.3.6 PRO_AUTHORIZE

Defines the required authorization level for authorizing menu commands and toolbar buttons.

If the current authorization group is not found, the authorization group GENERAL is used. To configure the authorization levels for specific commands, the authorization level 2 (Engineering) is required.

24.2.3.7 PRO_CONFIG_COMMANDS

Defines the required authorization level for configuring custom commands to menu commands and toolbar buttons.

If the current authorization group is not found, the authorization group GENERAL is used. To configure the custom commands, the authorization level 2 (Engineering) is required.

24.2.3.8 PRO_MANAGE_LAYOUTS

Defines the required authorization level for managing layouts in Monitor Pro. The following operations are covered:

- Configuring the visibility of menus or toolbars in Monitor Pro
- Customizing menus and toolbars with certain visibility in Monitor Pro

Visibility means project, application or user one. Menus and toolbars with:

- Project visibility are shown in all application for all users
- Application visibility are shown in certain application for all users
- User visibility are shown for certain user

24.2.3.9 PRO_NOTEMARKER_HANDLING

Defines the required authorization level for handling Process Display Note

If the current authorization group is not found, the authorization group GENERAL is used. To handle Process Display Notes, the authorization level 1 (Control) is required. Process Display Notes can be viewed with the authorization level 0 (View).

24.2.3.10 PRO_OPTIONS_HANDLING

Defines the authorization level 2 (Engineering) for making changes in the **Application Settings** dialog. If the current authorization group is not found, the authorization group GENERAL is used.

24.2.3.11 PRO_TOPOLOGY_HANDLING

Defines the required authorization level for handling network topology coloring. The following operations are allowed for the PRO_TOPOLOGY_HANDLING group:

- Network topology coloring in the engineering mode.
- Select **Tools/System Tools/Network Topology Coloring** to open the network topology coloring.
- Network topology coloring in the non-engineering mode (read only).

If the current authorization group is not found, the authorization group GENERAL is used. To handle network topology coloring, the authorization level 1 (Control) is required.

24.2.3.12 PRO_ZOOMAREA_HANDLING

Defines the required authorization level for handling zoom area definitions.

Editing, deleting or adding the zoom area definitions in the Save/Restore Zoom dialog are allowed for the PRO_ZOOMAREA_HANDLING group.

If the current authorization group is not found, the authorization group GENERAL is used. To handle zoom area definitions, the authorization level 1 (Control) is required.

24.2.3.13 REPORTS

Defines the required authorization level for handling Measurement Reports.

- Editing application specific preconfigurations.
- Showing or hiding the measurement reports by using the **Measurement** dialog.

If the current authorization group is not found, the authorization group GENERAL is used. To handle Measurement Reports, the authorization level 1 (Control) is required. To edit the application specific preconfigurations, the level 2 (Engineering) is required.

24.2.3.14 SUPERVISION

Used in System Self Supervision control dialog.

24.2.3.15 TOOLS

Defines the required authorization level for tools to cover appropriate functionality. Used for example in Display Builder and Tool Manager.

24.2.3.16 TREND_HANDLING

Defines the required authorization level for handling Trends:

- Editing Trend Basket
- Editing application specific preconfigurations
- Trend Curves dialog modifications

If the current authorization group is not found, the authorization group GENERAL is used. To handle trends, the authorization level 1 (Control) is required. To edit the application specific preconfigurations, the level 2 (Engineering) is required.

Section 25 Saving Customized Application

25.1 Search Path and Relative Paths in Process Displays

In Monitor Pro and Display Builder, there is an internal search path that is used to convert relative file paths to absolute file paths in runtime when searching for subdrawings, Process Displays, input object layout files, bitmaps and so on, from disk. The search path contains the following directories:

\sc\apl\<application name>\PICT\

\sc\apl\<application name>\AplMod4\Palette\

:\sc\prog\graphicsEngine\Palette\

\sc\prog\graphicsEngine\etc\

\sc\prog\graphicsEngine\lib\

\sc\prog\graphicsEngine\lib\views\

\sc\prog\graphicsEngine\lib\fonts\

\sc\prog\graphicsEngine\lib\templates\

\sc\prog\graphicsEngine\lib\templates\drawings\

\sc\prog\graphicsEngine\support\

If the AplMod4\Palette\ directory is created manually created afterwards, Monitor Pro and Display Builder have to be restarted to take it in use.

When, for example, a subdrawing object is added to a Process Display, the related subdrawing file can be referenced in three different ways:

- Absolute full path, for example, C:\sc\prog\graphicsEngine\Palette\01 - SA_Common \Generator.sd
- A full path without the drive letter, for example, \sc\prog\graphicsEngine\Palette\01 - SA_Common\Generator.sd
- A relative path containing only the Palette tab name and the subdrawing file name, for example: 01 - SA_Common\Generator.sd

The relative path is the recommended way and the default when, for example, a subdrawing is added from Palette or Object Browser. The absolute full path is the default whenever, for example, a subdrawing is added or changed by browsing the file system. It should always be manually changed to back to relative path, or at least the drive letter should be extracted away.

Section 26 Tool Manager

The primary use of the Tool Manager is to start tools of various categories. The Tool Manager is divided into several pages with tool icons. The user is allowed to add both pages and tools. The geometry of the Tool Manager is saved at the end of a session. From an application, the Tool Manager is opened by choosing **ToolManager** from the Tools & Engineering Tools menu in Monitor Pro.

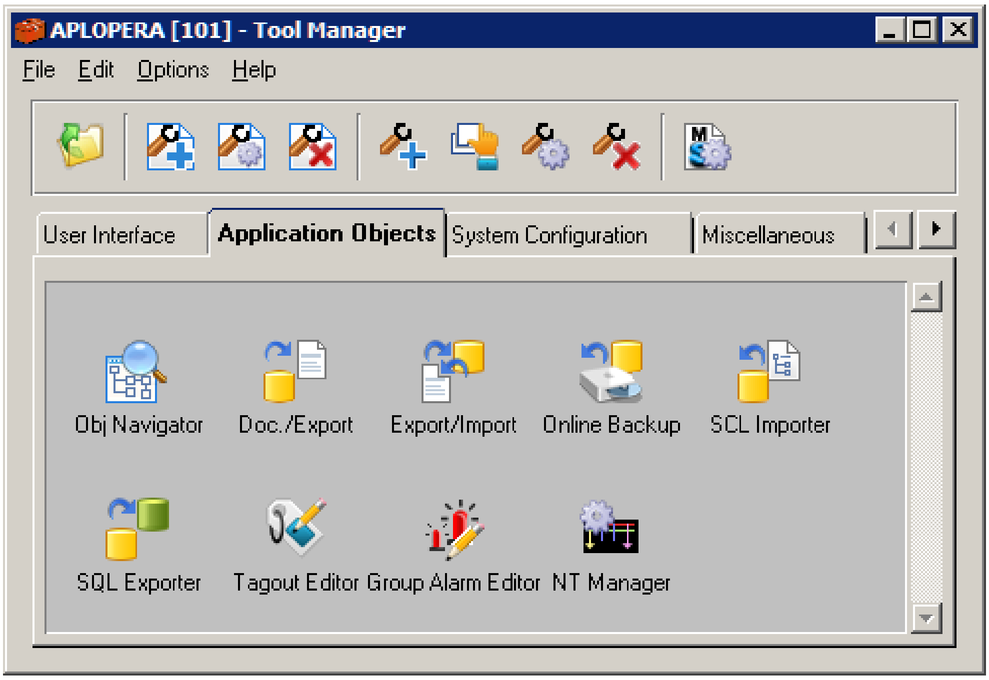


Figure 260: User Interface page. Each page in the Tool Manager represents a category of tools

26.1 General

26.1.1 Tool types

The Tool Manager handles three kinds of tools: services, system tools and object tools. Services are automatically started when the Tool Manager is started. Services provide other tools with functionality when they are used. System tools are used when the SYS environment and its applications are being maintained. Object tools use services provided by the project database management service, which is not shown in the standard version. It is part of the CAP package.

26.1.2 The toolbar

The toolbar functions are listed in Figure 261.

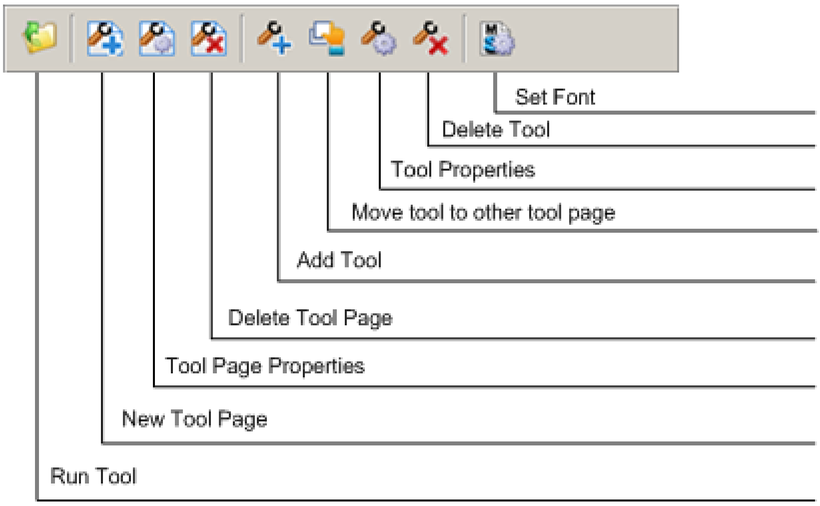


Figure 261: Tool Manager toolbar

26.1.3 Tool pages

Tool pages can be added to the Tool Manager by choosing **Insert Tool Page** from the Edit menu. Tool pages cannot be deleted before all tools are removed from the page. The tool page title can be changed by choosing **Tool Page Properties** from the Edit menu.

26.2 Tools

This section describes how to manage tools using Visual SCIL. The pictures are managed the same way, except adding pictures is done differently.

26.2.1 Adding tools

To add tools:

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

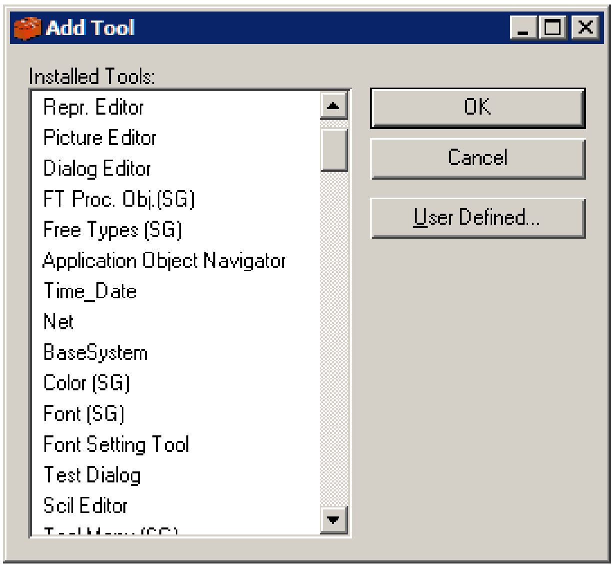


Figure 262: The tool to be added to the Tool Manager can be selected from the Installed Tools List

2. Select the tool that should be added from the list. The list is retrieved from the tool registry. Add unregistered tools by clicking the **User Defined** button.
3. Click **OK**.

26.2.2 Adding user defined tools

Adding a user defined tool is done in the dialog box shown in Figure 263.

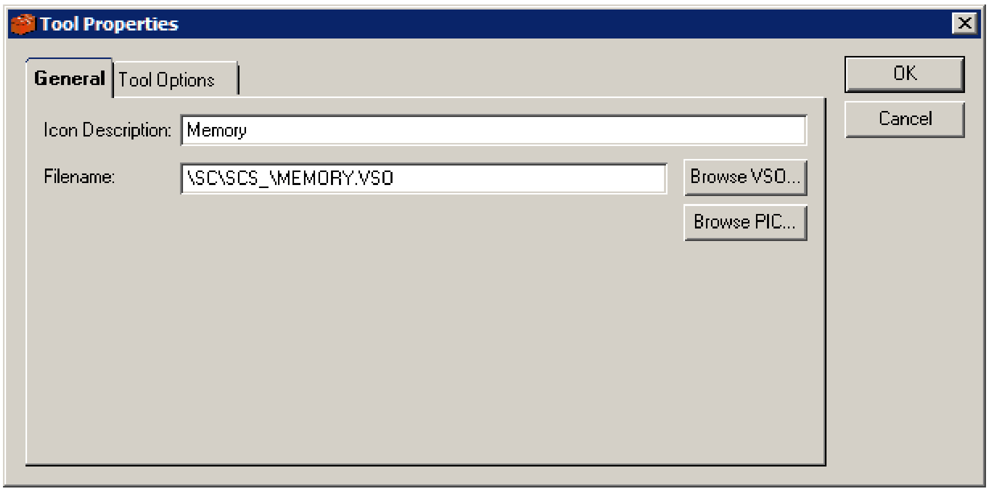


Figure 263: The General page

To add user defined tools:

1. In the **General** page, enter the tool title and its associated Visual SCIL Object resource filename. Use the browse facility if needed.
   1.1. Following the same procedure you can add an icon referring to a picture anywhere in the file system. Enter picture title and complete filename or use the **Browse PIC** button to locate the picture file.
2. In the **Tool Options** page, type additional information like the Tool Type (service, sys or obj, default=sys), Dialog name (default=MAIN), Method to run when loading the dialog

(default=none), Exclusive state (0 or 1, default=0) and the Default Path for the tool. If the VSO contains a VS_IMAGE named PRODUCT_ICON_M, it is shown as the icon by default.

- The specified method can also have parameters, for example:

```
MyMethod("PAGEVIEW", 5)
```

- Here the method is MyMethod and it passes two parameters: "PAGEVIEW" as argument 1 and number 5 as argument 2.
- The Exclusive state is used to control whether the tool allows other tools to be active simultaneously in the same Tool Manager session. If this is set to 1 (TRUE), the tool does not allow other tools to run at the same time. The default path must be specified as a logical path.

3. Click **OK**.

26.2.3 Moving tools to another tool page

Tools can be moved from one tool page to another. To move a tool:

1. Select the tool.
2. Choose **Move Tool** from the Edit menu. The Move Tool dialog appears.

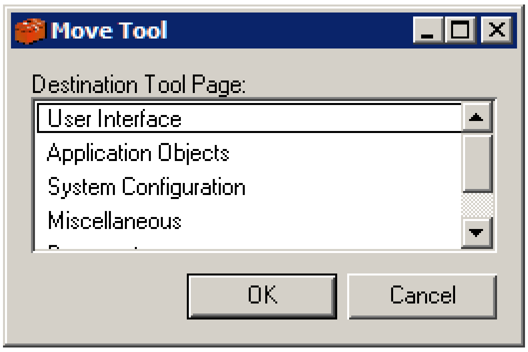


Figure 264: Dialog for moving a tool icon from one page to another.

3. Select the destination page.
4. Click **OK**.

26.2.4 Tool properties

The tool properties can be edited. To edit:

1. Choose **Tool Properties** from the Edit menu. The Tool Properties dialog box appears, containing three pages.

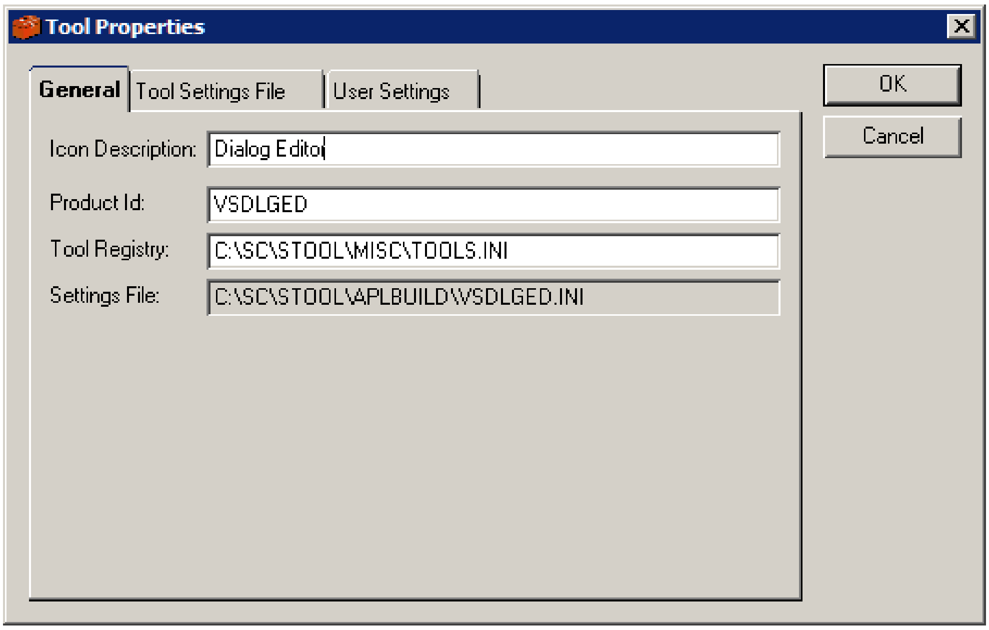


Figure 265: General Page where you determine the tools to be inserted and its properties

2. Type Product ID and associated Tool Registry file for the tool in the **General** page. Tool Manager locates tools by their product IDs from either the Sys_Tool or Cmd_Tool logical paths. The settings file contains detailed information about the tool.
   - The **Tool Settings File** page shows information from the tool's settings file. Figure 266. These settings cannot be changed from the Tool Manager.

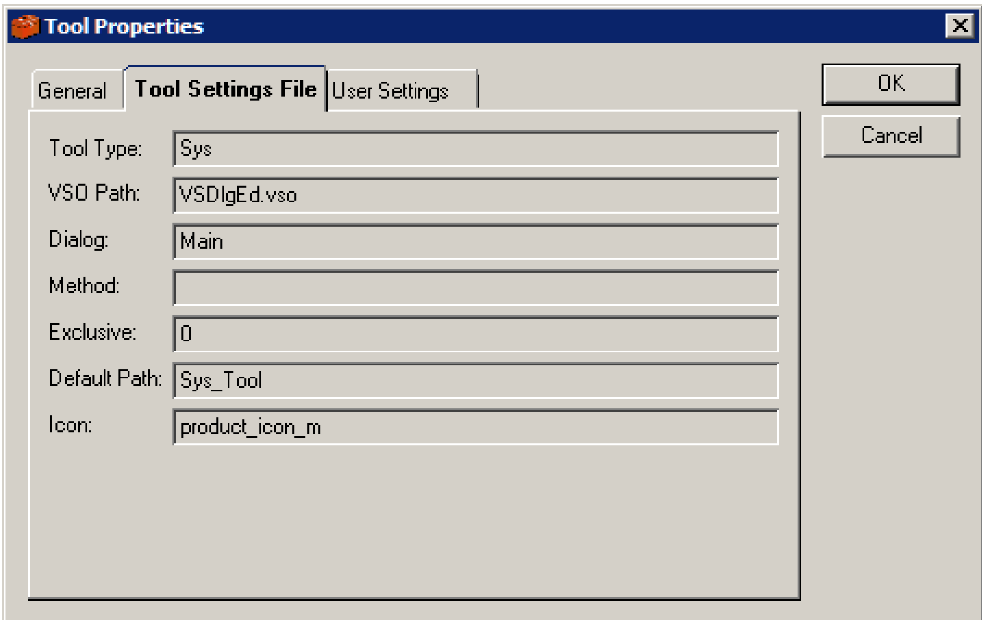


Figure 266: The Tools Settings File Page

   - The **User Settings** page has the same text boxes as in the **Insert Tool** dialog box, see Figure 267. In the **User Settings** page of this dialog, it is possible to override the settings in the tool's settings file. For example, it is possible to override the Exclusive state by setting it to its opposite value. Change the settings if needed.

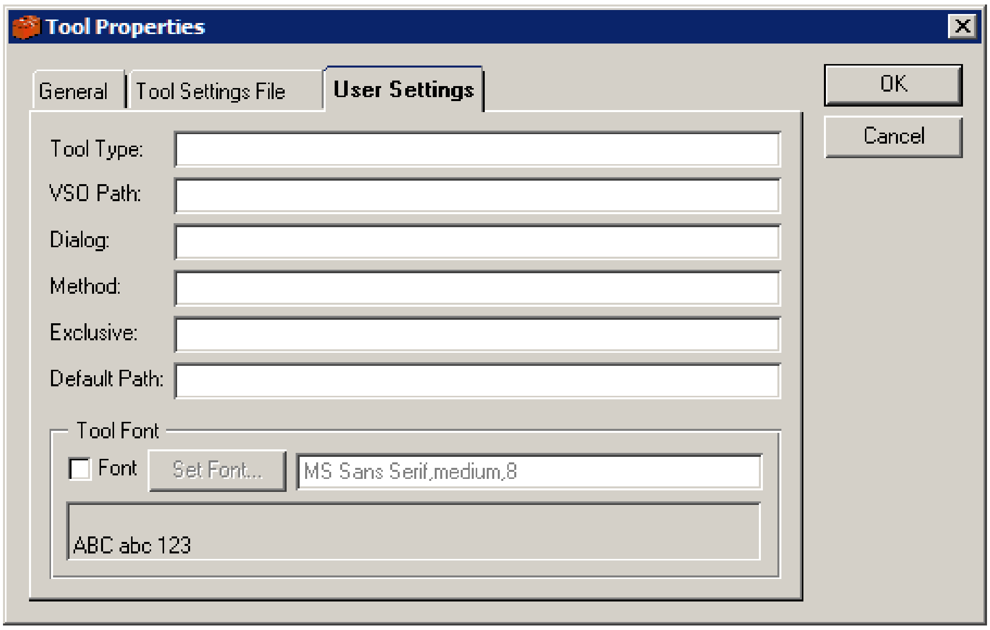


Figure 267: The User Setting Page

3. Click **OK**.

26.2.5 Deleting a tool

Tools can be deleted by choosing **Delete Tool** from the Edit menu. Deleting a tool does not affect the Tool Manager registry of installed tools, only the active user's tool collection.

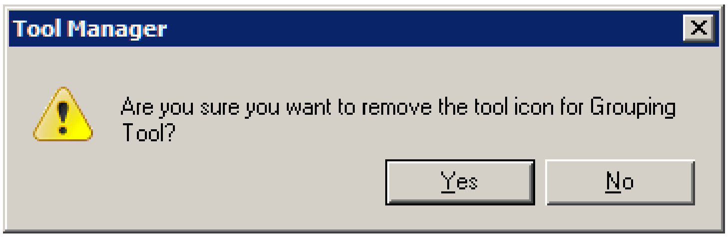


Figure 268: The Delete Tool confirmation dialog box

26.3 Setting Global Font

The default font used by VSCIL tools is MS Sans Serif, medium, 8.

The font can be changed using the **Set Font** option in the Tool Manager. Launch the Font Properties dialog by either selecting **Set Font** from the Options menu or click the **Set Font** button in the tool bar.

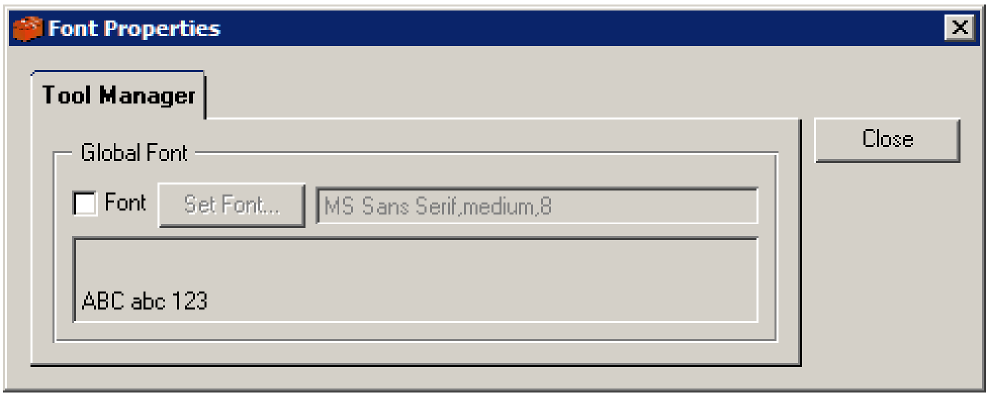


Figure 269: The Font Properties dialog which allows user to set Global Font

Check the Font check box to enable the **Set Font** button, which allows to choose any of the listed fonts.

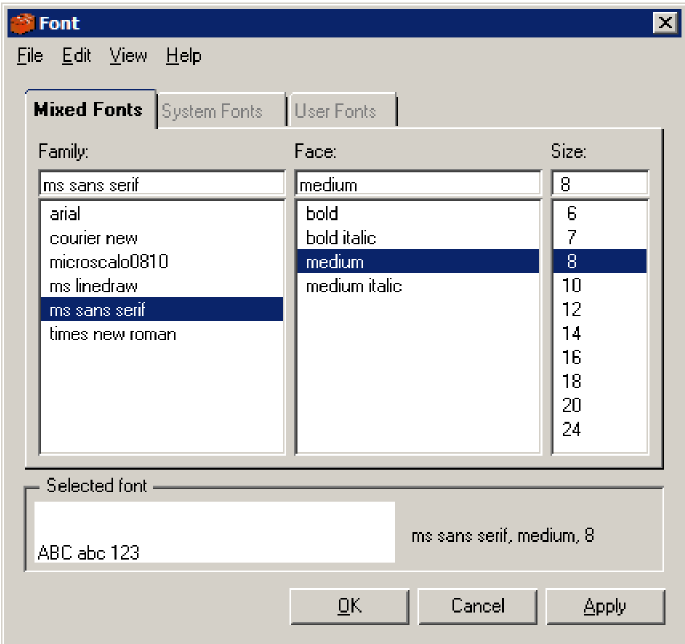


Figure 270: The Font Chooser dialog that allows user to select Fonts

When the font is set, the following tools will be launched with the selected font.

- Object Navigator
- NT Manager
- Picture Editor
- Import/Export Tool
- Search Tool
- SCIL Editor
- Test dialog
- License Tool.
- DMT

26.4 Setting Tool specific fonts

To set font for individual tools, see Figure 267:

1. Select the tool and select the **User Settings** tab in the **Tool Properties** dialog.
2. Select the Font checkbox and click the **Set Font** button.

For the above mentioned tools when the Font is set globally and if fonts are not set specifically for each of these tools, then it will be launched with the Global Font that is set.

26.5 Viewing and terminating currently loaded tools

A dialog box for viewing the currently loaded tools can be opened in the Tool Manager. From this dialog box, it is possible to terminate tools not responding to user interaction.

To view active tools loaded in current Tool Manager session:

1. Choose **Active Tools...** from the File menu.
2. Currently active tools are listed in a list box.
3. To terminate a tool, select the tool in the list box and click **Terminate**.
4. The Active Tools dialog box is closed by clicking **Close**.

26.6 Pictures

Adding a picture icon into the Tool Manager:

1. Choose **Insert Tool** from the Edit menu.
2. Click **User Defined** button.
3. Click **Browse PIC**.
4. Select a file from the list of picture files.
5. Click **OK** in the Tool Properties dialog.

Otherwise, pictures are handled in the Tool Manager the same way as tools made using Visual SCIL.

Section 27 SNMP OPC server

Integrated SYS600 SNMP OPC Server provides an OPC Gateway for the SNMP protocol. The server can be configured with the help of IET600 and IET Data Loader. In IET600 the SNMP IEDs are created with the process database containing the SNMP OIDs to be mapped to their respective process objects.

SNMP OPC Server can also be configured with Communication Engineering Tool, but in this case the process object creation has to be handled separately.

27.1 Creating the SNMP IED configuration with IET600

Refer to the IET600 documentation on exact steps needed to create the SNMP IED configuration and how to map process objects with the specific SNMP object identifiers (OID).

The process object types, which can be mapped to SNMP items are:

- Binary input and output objects. SNMP Source can be any numeric SNMP data type with an additional bit mask value.
- Analog input and output objects. SNMP source can be any numeric SNMP data type.
- Bit Stream objects for text data. Bit stream objects can be mapped to an OID or to a static text when a static attribute is used in IET600.
- The other process object types are ignored and not added to SNMP OPC server configuration.

Process objects should be one of the mentioned types and contain a valid UN. The IN attribute is not needed in the export file and it preferably should be left empty, as it will be overwritten to match the OPC SNMP configuration.

27.2 Importing SNMP OPC server configuration with IET Data Loader

If the IET600 export file contains an SNMP OPC configuration for SYS600 Integrated SNMP OPC server, it is imported automatically by IET Data Loader. OPC Data Access client configuration is also created for the server. See the SYS600 IEC 61850 System Design manual for the configuration of Data Access client with IET Data Loader, if the default values need to be changed.

SNMP OPC Server specific configuration attributes can be set in IET600 or during the IET Data Loader import.

Table 249: SNMP IED configuration attributes

Category	Attribute	Value Default [type, range]	Description
[10] Basic	Simulation Mode	False [Boolean]	Whether the SNMP IED is used in simulation mode.
	In Use	In use[one of: in use, not in use]	Specifies whether IED is in use or not.
	SNMP Version	SNMP v2c [one of: SNMP v1, SNMP v2c, SNMP v3]	Version of SNMP used for outgoing requests.
[20] Addresses	IP address of the IED	127.0.0.1 [String, IP Address]	IP Address of the IED.
	Local Host IP	127.0.0.1 [String, IP Address]	IP address of the local host.
[30] Community	Community Name	Public [String]	Read community name defined in the network device.
	Write Community Name	Private [String]	Write Community name defined in the network device.
[40] Polling	Polling Rate	900 [Integer, 0-3600]	Polling rate in seconds.
	Timeout	10 [Integer, 0-3600]	Timeout for polling. If timeout property is set to 0, all operations return immediately. With positive values system will wait for the operation to complete before returning.
[50] Security (Applicable to SNMP v3 only)	User Name	String	User name used for SNMP v3
	Authentication Protocol	Not Used [One of: Not Used, MD5 or SHA] *Note: MD5 is derived from the RSA Data Security, Inc. MD5 MessageDigest Algorithm*	The authentication protocol used for SNMPv3 packet.
	Authentication Password	String	The password used for SNMPv3 authentication.
	Encryption Algorithm	Not Used, [One of: Not Used, DES, AES, 3DES]	The encryption algorithm used for SNMPv3 packets.
	Encryption Password	String	The password used for SNMPv3 privacy.

Section 28 Terminology

Term	Description
Action tool	Used to perform an action within the application. An action tool can be a menu command or a toolbar tool. A tool can be any of the following types: button, drop-down button, textbox, combo box, label, separator.
Docking area	Monitor Pro defines four docking areas on the top, bottom, right and left edges of the control's dialog. It also defines an area for free position (floating).
Context menu	The context menu appears when you right-click for example a selection, a toolbar, or a taskbar button. It lists commands pertaining only to that screen region or selection.

Section 29 Abbreviations

Abbreviation	Description
CSV	Comma delimited text file
HSI	Human-System Interface
NCC	Network Control Center
DLL	Dynamic linking Libraries
NLS	National language support
NTC	Network Topology Coloring
SA-LIB	Power Process Library
SCS	Substation Control System
SSS	System Self Supervision
VBA	Visual BASIC for Applications

Appendix A NTC migration to SYS600 9.3 FP1 or later

The architecture of NTC is changed in SYS600 9.3 FP1. From now on, the calculation is done in the base system and the result of topological state of the network is stored in the process database.

Due to this change, some manual changes regarding NTC are required when the application is migrated to SYS600 9.3 FP1 or later.

During installation, the components related to the previous NTC version are removed.

The steps in the following sections apply only partly, when the migration is done from version 9.x, where x refers to version used before migration, to SYS600 9.3 FP1 or later.

1.1 Disabling starting and stopping of Network Topology Coloring Manager

The starting and stopping of NTC Manager have to be disabled.

To disable starting and stopping of NTC Manager, comment (add the semicolon character ";" at the front of a line) or remove the following lines:

File: SC/SYS/ACTIVE/SYS_/SYS_BASCON.COM

Line:

Stat = ops_process("\sc\prog\sa_lib\ntcmanager.exe")

File: SC/PROG/EXEC/SHUTDOWN.CIN

Line:

@a=ops_process(parse_file_name("/prog/sa_lib", "ntcmanager.exe") + " /kill")

The lines mentioned will no longer exist in the file templates of versions 9.3 FP1 or later.

1.2 Defining switching device polarity

The switching device polarity needs to be defined for the application, or for each base system STA object separately. The definition is done by setting the APL'apl_nr':BSM or STA'sta nr':BSM attributes in System Configuration Tool or directly in SYS_BASCON.COM.

The rule for switching device polarity is the following:

IEC 61850: Middle = 0, Open = 1, Closed = 2, Faulty = 3

Other protocols: Middle = 0, Open = 2, Closed = 1, Faulty = 3

In practice, the polarity needs to be defined for IEC 61850 stations. For other protocols, the polarity is the same as the default value.

Example 1: Pure IEC 61850 application

```
#SET APL1:BSM = LIST(Middle = 0, Open = 1, Closed = 2, Faulty = 3)
```

Example 2: Mixed IEC 61850 application with other protocols

```
@v_IEC61850_Stations = vector(1,2,3,4,5,21,22,23,24,25)

#loop_with Sta= 1 .. length(%v_IEC61850_Stations)

@i_StaNr = %v_IEC61850_Stations(%Sta)

#set STA'i_StaNr':BSM = LIST(Middle = 0, Open = 1, Closed = 2, Faulty = 3)

#loop_end
```

1.3 Network Topology Coloring layouts

Before performing migration, investigate how the topology is built in the application. There can be one network overview containing the whole topology or different independent topologies.

If there is one network overview display, perform the migration for that display only. This applies if all objects are in the same view, or the overview display is built from subdrawings (stations) that are connected to each other by line segments.

In some applications, the overview display can be built in a way that it contains only a subset of all objects, and more detailed displays are included as subdrawings. In this case, the topology can be built by starting from the overview and then adding the detailed views to the same topology one by one.

Handle independent topologies separately and save them under unique names.

If the display was built using network objects referenced from the overview display, these displays have to be built from the start.

Migration does not apply to selfmade NTC solutions.

1.4 Rebuilding Network Topology Coloring

NTC is rebuilt in Display Builder.

To rebuild NTC:

1. In Monitor Pro, select **Tools/Engineering Tools/Display Builder**.
2. In Display Builder, open the **Actions/Manage Topologies...** dialog.
3. Click **Add New...** and define topology name.
4. Click **Add Current Display...** or **Add Display...** to select display.

5. Select an object causing messages and fix the errors.

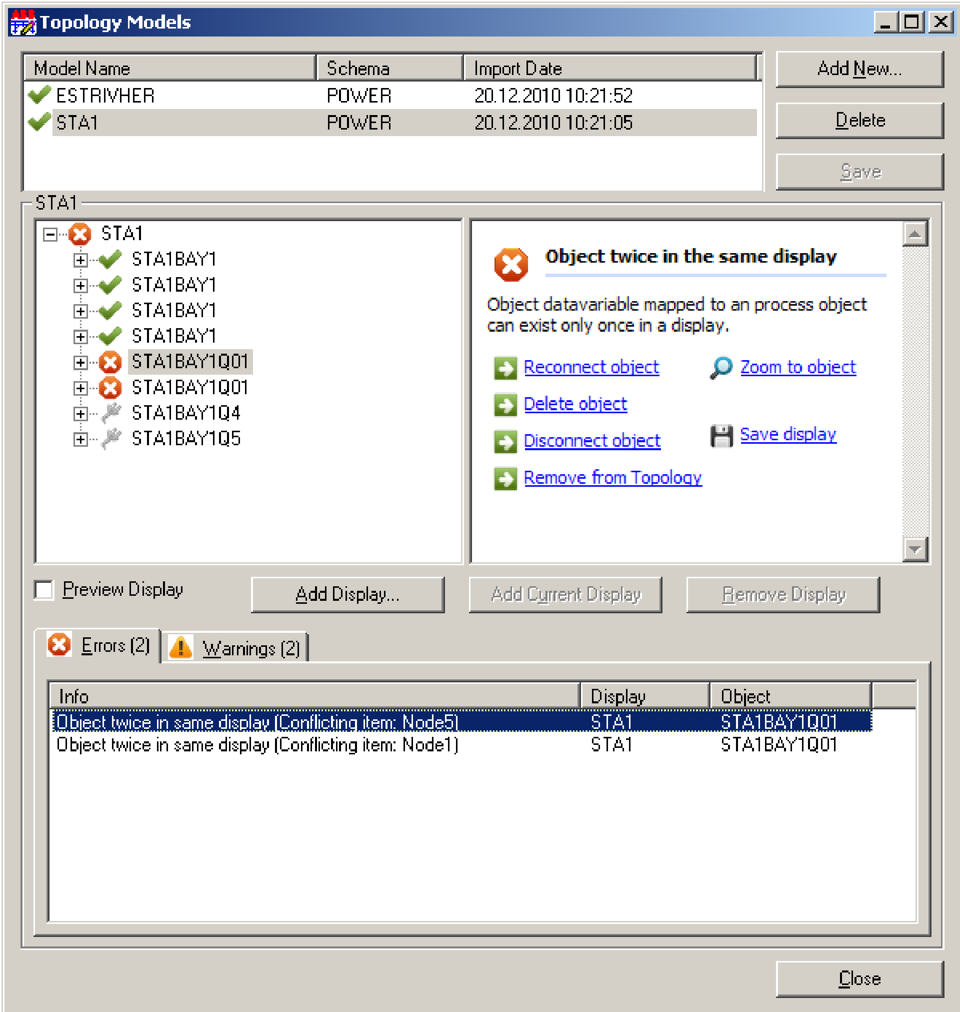


Figure 271: Error message due to switch device symbol connected twice to same process object in one display

The Topology Models dialog shows the reasons for the messages and proposes actions for fixing the errors and warnings. The user can zoom to the object causing the error or warning also by right-clicking it in the object tree or in the **Errors** or **Warnings** tab.

6. Click **Save**.
   Some errors are discovered during saving. If errors are detected at this phase, repeat steps 5 and 6.

> The rules for building NTC are stricter in 9.3 FP1 than in previous versions, which may result in a number of errors and warnings when NTC is generated.
> Fix all errors and check all warnings before saving the topology.

7. Check the logical behavior of regenerated topology in Monitor Pro.

1.5 Known issues

1.5.1 Duplicate switch devices

The new NTC algorithm allows that switch device can be in the same display only once. If there were duplicate objects in the display, remove them.

1.5.2 Breaker with truck

Draw breaker with truck so that breaker is in the middle and truck symbols are directly connected to it on both sides.

1.6 Line indicator

1.6.1 Configuration of line indicator

The line indicator can also feed Uncertain color. During the first startup after installing 9.3 FP1 or later, line indicators are turned to manual state automatically if they are configured so that the line indicator is colored by Busbar Coloring or manually.

There may be a configuration error indicating that line indicator(s) are colored by DMS600 although they should be colored by the Busbar Coloring. In such a case, the line indicator(s) (index 10) needs to be turned to the manual state (SS=1).

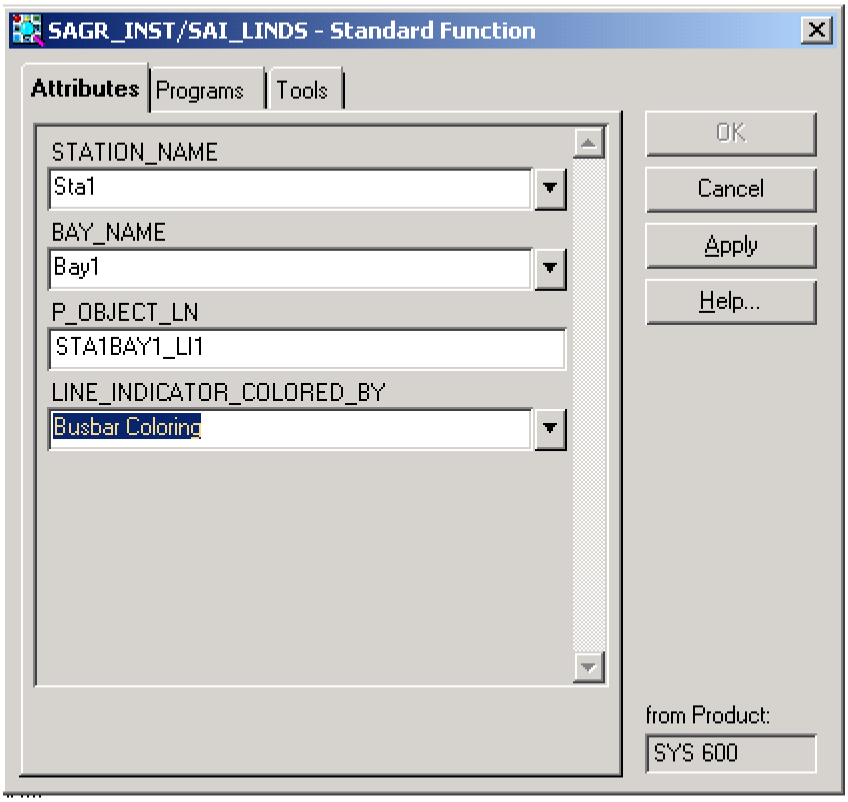


Figure 272: Configuration of line indicator

1.6.2 Topology color data variable

In SYS600 9.3 FP1 or later, the line indicator does not contain Topology color data variable anymore. Also, the topology color data variable of line segment and generator symbol is now used by the NTC. If some customer specific solution was created based on this, it needs to be recreated.

Appendix B Migrating MicroSCADA application in SYS600 and Monitor Pro

This section provides information on the migration tools needed during the application design when the existing LIB 5xx based applications are migrated to use the Monitor Pro Process Displays.

Usually, an existing application contains a set of pictures and their related process database built with LIB 5xx application library. SYS600 contains the engineering tools to migrate these applications in a straightforward way. During the migration, the picture configuration information from the existing pictures is copied to the process database. The Monitor Pro user interface uses the copied information. The connection between the existing process objects of the migrated application and the graphical symbols of the Process Displays is made by dragging and dropping the complete objects during the single line diagram engineering with Display Builder.

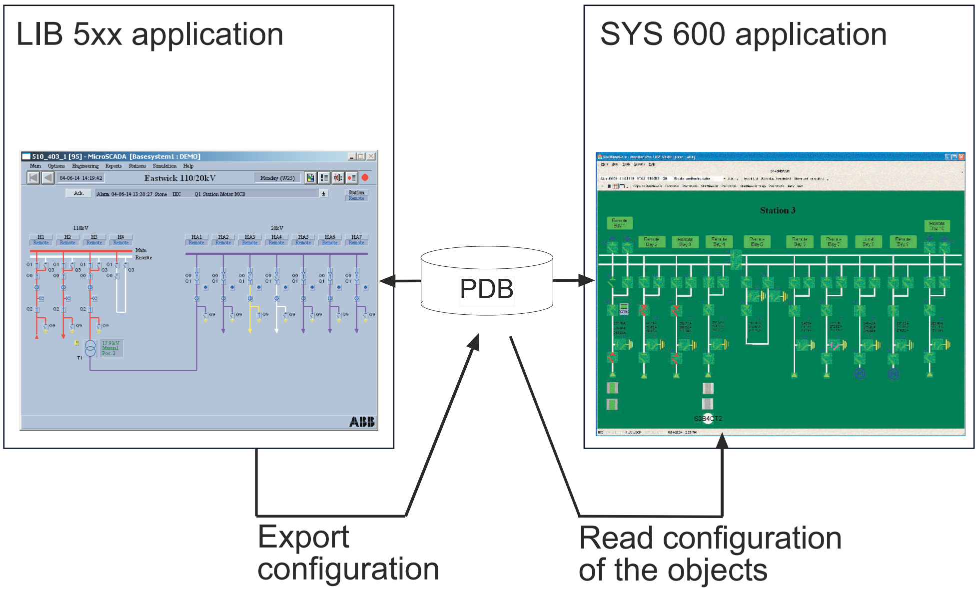


Figure 273: Migrating process

Another migration tool is needed for the event handling concept introduced in SYS600. In this concept, the Event Objects are displayed in the application's process database. Another event architecture has been used with the existing application, which database has been built with LIB5xx application library and contains the LIB 500 Event List. To provide a continuum from the existing LIB 500 Event List to the SYS600 Event List, a migration tool to migrate the existing application to the new event handling concept is introduced. The following figure describes both the event handling processes and the role of the migration tool.

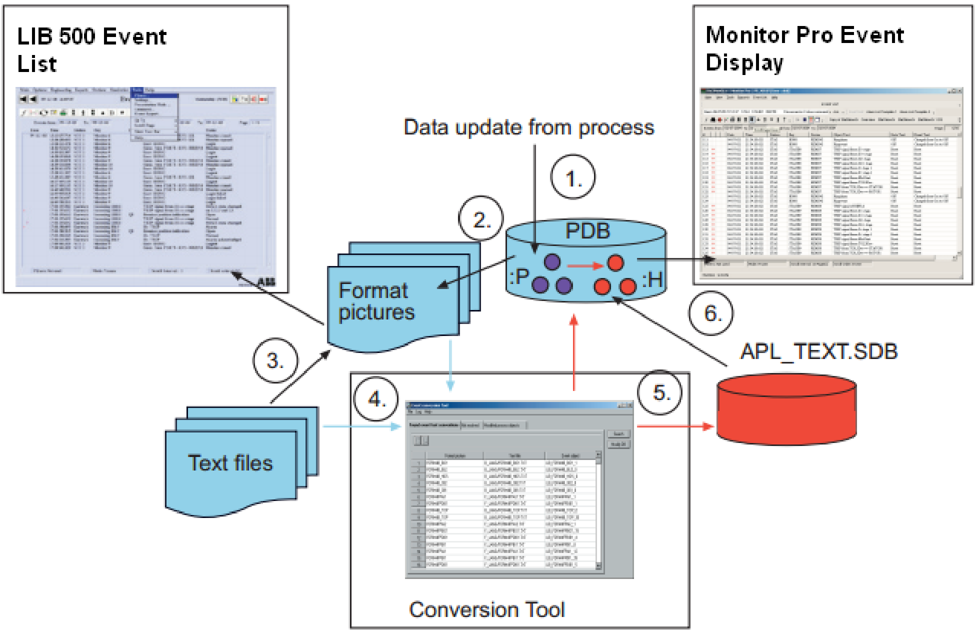


Figure 274: Event conversion process

LIB 500 Event List concept:

- The value of the process object is updated (1)
- Format picture is activated (2)
- Format picture reads the text related to the event from the text file and shows it in the Event List (3)

Event Conversion Tool:

- Reads all the format pictures and related text files found from the application (4)
- Puts texts to text database, creates Event Objects, connects Process Objects to Event Objects (5)

Monitor Pro Event Display concept:

- The value of the process object is updated (1)
- Event object is activated. Event Object contains the reference to text database, from where the textual presentation of event is fetched (6)

By default, Power Process control dialogs can connect to the configuration made with LIB 510 MV Process. However, it is possible to export data from any picture function that is designed according to LIB 5xx design rules.

1.1 Exporting information from pictures

In a LIB5xx based picture, the parameters describing the object and its behaviour is stored in the picture. When the Monitor Pro Process Displays are used with an existing process database, these parameters need to be extracted from the pictures and made available for the new control dialogs.

The Configuration Data Export tool is used for this purpose. It extracts the needed

information from the pictures and stores it to the process database, from where it is available for the Control dialogs of Power Process library. The configuration of the Measurement Reports is extracted to a separate files, to directory sc/apl/'apl name'/reports/preconf.

1.1.1 Configuration Data Export tool

The Configuration Data Export tool offers the means to reuse the pictures and process database built with LIB 5xx application library when an application is upgraded to the Monitor Pro user interface. First, Configuration Data Export can separate the configuration of the picture functions from the SCIL picture. Next, the tool copies the information to the process database. In case of Measurement Reports, tool creates the external configuration data files that are saved in apl\apl name\REPORTS\PRECONF. The Process Displays and measurement report displays in Monitor Pro can utilize the exported configuration data. For more information on the picture engineering, see SYS600 Process Display Design.

Starting Configuration Data Export tool

To start the Configuration Data Export tool, select **Miscellaneous/DAT Export**.

Figure 275: Configuration Data Export tool icon

Main view

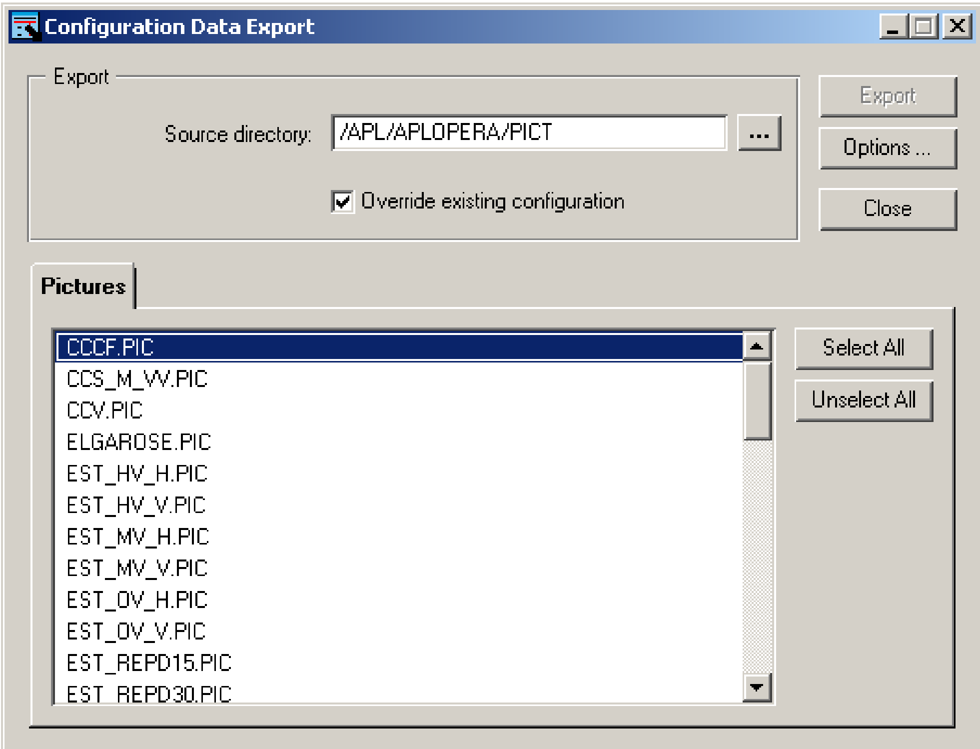


Figure 276: Main view of the tool

The Source directory shows where the SCIL pictures are located. The destination directory is always the /POBJCONF directory under the application. When the Override existing configuration option is checked, the configuration data is recreated, if it already exists. The SCIL pictures found from the source directory are listed in the Pictures tab.

Options

The purpose of the Export Options is to filter out the picture functions that are not related to the process graphics. These kinds of functions are, for example, components that are used for building the network topology coloring into the LIB 500 single line diagrams.

In this way, the amount of the data files can be reduced. Also, by having this kind of filter, the creation of unnecessary configuration data will be prevented.

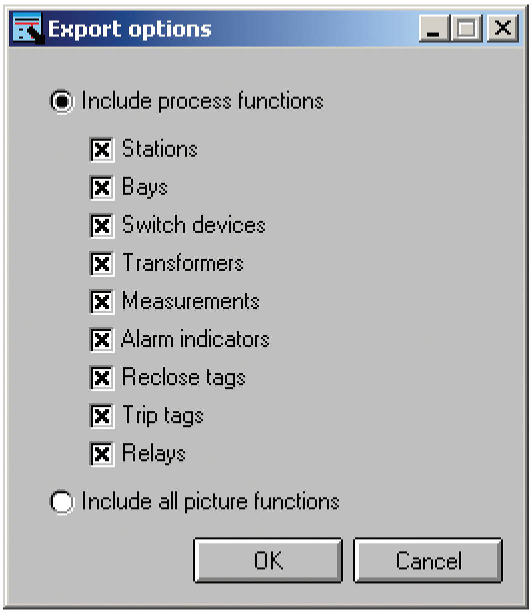


Figure 277: Export Options

Exporting picture data

To export configuration data from a picture:

1. Select the picture(s) in the Pictures list from which data should be exported.
2. Select filter from the Export Options dialog.
3. Click the **Export** button.

For more information on how to finalize the display engineering, see SYS600 Process Display Design.

1.2 Event conversion

In LIB 500 Event List, the format pictures are used for showing the dynamic part of the event text. However, the event text presentation in SYS600 Event List is based on the Event Objects. When SYS600 Event List is used with an existing process database, the format picture mechanism must be converted into the new event handling concept.

The SYS600 Event Conversion Tool offers the means to convert the used format picture concept into a new event handling concept. In practice, this means that each process object that has a known LIB 5xx format picture is connected to a comparable Event Object. When the event conversion is done, the same event texts are shown in the SYS600 Event List and in the LIB 500 Event List.

1.2.1 Event Conversion Tool

The purpose of Event Conversion Tool is:

- To locate the used, known LIB 5xx format pictures from the process objects of the application.
- To create the related Event Objects that the new event handling concept needs.
- To separate event texts from the text files, and write them to a text database file (APL_TEXT.SDB) of the running application.
- To connect the created Event Objects to the text identifiers that are located from the text database file.
- To connect process objects to event objects.

Format pictures are still needed to print the events to the Event Printer. Due to this, the format picture information in process database is not modified during the conversion.

Starting Event Conversion Tool

Start Event Conversion Tool from Tool Manager. Select the **Miscellaneous** tab and click **Event Conversion**.

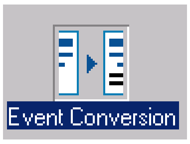


Figure 278: Icon for opening Event Conversion Tool

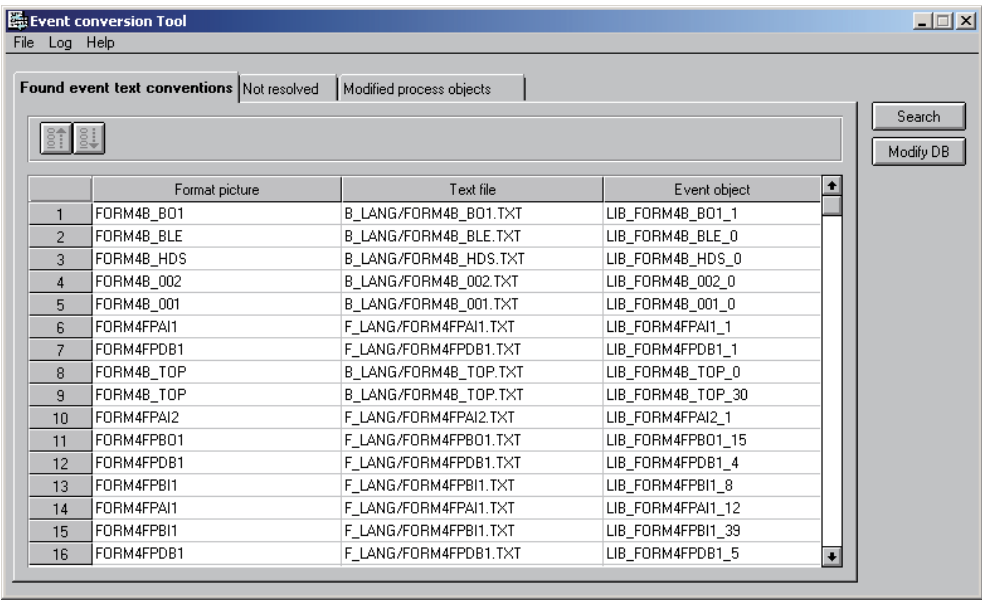


Figure 279: Main view of Event Conversion Tool

The Event Conversion Tool contains three tabs:

- **Found Event Text Conventions**
- **Not Resolved**
- **Modified Process Objects**.

Found Event Text Conventions tab shows all the supported format picture conventions found from the application. The conventions within the application can be examined by clicking **Search**.

The Format picture field shows the name of the format picture (PF attribute of process object). The Text file field shows the language text file related to the format picture. Sometimes the event text does not come from the text file, but the value of the process data is shown directly in the Event List (OV-attribute). In this kind of cases a text "Value based form" is shown in the field. The Event object field shows the name of the event object or objects where the process object is connected.

When the modification of the process database is done, the **Not Resolved** tab shows all process objects that are not connected to the event object. Possible reasons for the process objects not to connect are:

- Not supported software package. The process objects are not recognized because they are not compatible with LIB4.
- Not supported form convention. Some format pictures have internal logic that is not possible to convert by the Event Conversion Tool.
- Language text file not found. It is not possible to read the language text file.
- Event text could not be resolved. Event Conversion Tool does not recognize the text file format and therefore cannot resolve the text.

Modified Process Objects tab shows the list of all process objects that are successfully connected to event objects.

Making conversion

The conversion can be done by clicking the **Search** button, which looks for the known conventions within the application. Clicking the **Modify DB** button makes the changes to the process database. If the user does not want to examine the event conventions beforehand, the **Modify DB** button can be clicked directly.

When the conversion is ready, the result can be saved to a log file. The log file contains all the data found from three tabs in the Event Conversion Tool.

It is recommended to take a backup from the application (process database) before making the event conversion.

Appendix C Object type example

In this example, the following definitions are made in object type for switch device.

- Indexes for controlling are changed to 210 (Select), 211 (Execute), 212 (Cancel)
- Control methods (METHOD_OPEN_SELECT ...) are used for device specific control actions
- METHOD_INIT_SIGNAL is utilized for creating process objects for controlling
- METHOD_CUSTOMIZE_OBJECTS is used for changing certain attribute values of process objects
- METHOD_GET_INDICATIONS is used for converting the value of switch state indication, interlocking signals, bay L/R indication and synchrocheck indication.
- Interlocking and synchrocheck bypass is enabled in control dialog

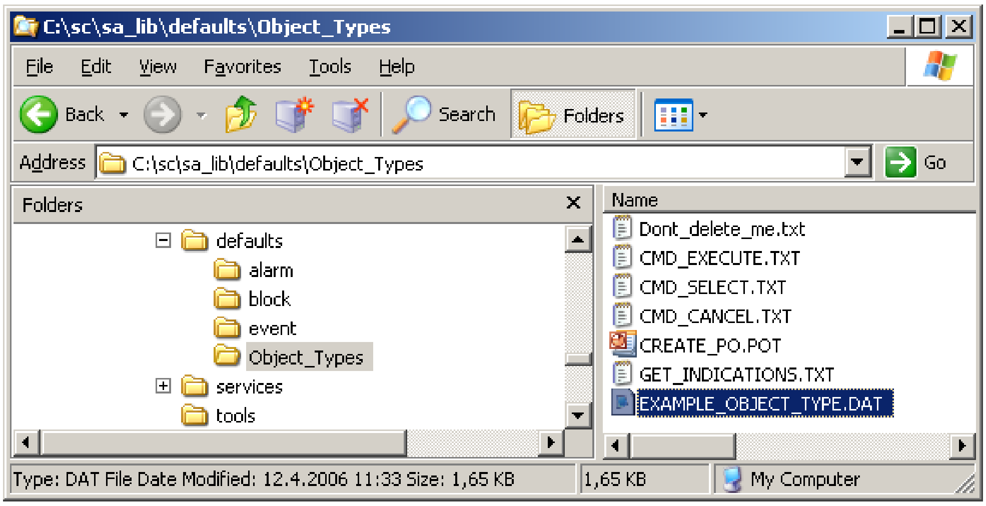


Figure 280: Object type example

1.1 Object type main file

EXAMPLE_OBJECT_TYPE.DAT

```
LIST(-

-;

-; Attributes defining index convention.

-;

INDICATION_DB_IX=10,-

OPEN_CMD_IX=210,-

CLOSE_CMD_IX=210,-

EXECUTE_CMD_IX=211,-

CANCEL_CMD_IX=212,-

EXECUTE_OPEN_CMD_IX=211,-
```

```
EXECUTE_CLOSE_CMD_IX=211,-

EXT_CNTR_BLK_IX=15,-

OPEN_ILOCK_IX=16,-

CLOSE_ILOCK_IX=17,-

ILOCK_CAUSE_IX=18,-

SEL_ON_MON_IX=19,-

CMD_EVENT_IX=20,-

AUX_PLUG_IX=21,-

SYN_IND_IX=22,-

-;

-; Attributes defining switch control methods

-;

METHOD_OPEN_SELECT=VECTOR("#DO READ_TEXT(""SAGR_OBJT/CMD_SELECT.TXT"")"),-


METHOD_CLOSE_SELECT=VECTOR("#DO READ_TEXT(""SAGR_OBJT/
CMD_SELECT.TXT"")"),-

METHOD_OPEN_EXECUTE=VECTOR("#DO READ_TEXT(""SAGR_OBJT/
CMD_EXECUTE.TXT"")"),-

METHOD_CLOSE_EXECUTE=VECTOR("#DO READ_TEXT(""SAGR_OBJT/
CMD_EXECUTE.TXT"")"),-

METHOD_CANCEL=VECTOR("#DO READ_TEXT(""SAGR_OBJT/CMD_CANCEL.TXT"")"),-

-;

-; Attributes for converting values for control dialog

-;

METHOD_GET_INDICATIONS=VECTOR("#DO READ_TEXT(""SAGR_OBJT/
GET_INDICATIONS.TXT"")"),-

-;

-; Methods Executed by POT.

-;

METHOD_INIT_SIGNALS=VECTOR("#DO READ_TEXT(""SAGR_OBJT/CREATE_PO.POT"")"),-

METHOD_CUSTOMIZE_OBJECTS=VECTOR(-

"#LOOP_WITH I = 1 .. LENGTH(%OBJECTS)",-

"@OBJECT=%OBJECTS(%I)",-

"@LN=%OBJECT.LN",-

"@IX=%OBJECT.IX",-

"#CASE %IX",-

"#WHEN 10 #SET ''LN'':PEH''IX''=""SAGR_FORM5FPDB1_17""",-
```

```
"#WHEN 16,17 #SET ''LN'':PEH''IX''=""SAGR_FORM5FPBI1_43""",-
"#CASE_END",-
"#LOOP_END"),-
-;
-; Attributes for overriding SCT selections
-;
SYNCHROCHECK_BYPASS = TRUE,-
INTERLOCKING_BYPASS = TRUE,-
SA_LIB_INDEXES = "20,19,210")
```

1.2 Process objects

CREATE_PO.POT

```
#IF 'ID'_PICTURE_FUNCTION_CONF:VcONTROL_TYPE == "OBJECT_TYPE" #THEN #BLOCK
@v_Object_Type_Process_Objects = VECTOR(-
LIST(-
IX = 210,- ;index =
PT = 11,- ;process object type = AO
DX = "X4",- ;directive text
OI = %OBJ_IDENT,- ;substation & bay name
OX = "Select",- ;object text
IU = 1,- ;in use = yes
SS = 1,- ;switch state = manual
EE = 1,- ;event enable = yes
HE = 1,- ;history enable = yes
HA = 2,- ;history activation = update
HF = 1,- ;history at first update = yes
HL = BIT_MASK(15),- ;history log number = 15
HO = 0,- ;high alarm limit = 0
LO = 0,- ;low alarm limit = 0
ST = "Command",- ;unit of value
SN = "1_1",- ;scale name
RC = 0,- ;receipt = acknowledgement not required
AB = 0,- ;alarm blocking = no
AC = 0,- ;alarm class = disabled
```

```
AD = 0,- ;alarm delay = 0
PD = 0,- ;picture device = none
PI = "",- ;alarm picture
PF = "FORM5SAGR1",- ;physical format for printout
RI = 0 ,- ;dynamic texts for printout
PA = 2,- ;printout activation = update
PU = 1,- ;picture/ printout at first update = yes
LD = BIT_MASK(1),- ;listing device = 1
RX = ""),- ;data acquisition unit and group
information
LIST(-
IX = 211,- ;index =
PT = 11,- ;process object type = AO
DX = "X4",- ;directive text
OI = %OBJ_IDENT,- ;substation & bay name
OX = "Execute",- ;object text
IU = 1,- ;in use = yes
SS = 1,- ;switch state = manual
EE = 1,- ;event enable = yes
HE = 1,- ;history enable = yes
HA = 2,- ;history activation = update
HF = 1,- ;history at first update = yes
HL = BIT_MASK(15),- ;history log number = 15
HO = 0,- ;high alarm limit = 0
LO = 0,- ;low alarm limit = 0
ST = "Command",- ;unit of value
SN = "1_1",- ;scale name
RC = 0,- ;receipt = acknowledgement not required
AB = 0,- ;alarm blocking = no
AC = 0,- ;alarm class = disabled
AD = 0,- ;alarm delay = 0
PD = 0,- ;picture device = none
PI = "",- ;alarm picture
PF = "FORM5SAGR1",- ;physical format for printout
```

```
RI = 0,- ;dynamic texts for printout

PA = 2,- ;printout activation = update

PU = 1,- ;picture/ printout at first update = yes

LD = BIT_MASK(1),- ;listing device = 1

RX = ""),- ;data acquisition unit and group

LIST(-

IX = 212,- ;index =

PT = 5,- ;process object type = BO

DX = "X1",- ;directive text

OI = %OBJ_IDENT,- ;substation & bay name

OX = "Cancel",- ;object text

IU = 1,- ;in use = yes

SS = 1,- ;switch state = manual

EE = 1,- ;event enable = yes

HE = 1,- ;history enable = yes

HA = 2,- ;history activation = update

HF = 1,- ;history at first update = yes

HL = BIT_MASK(15),- ;history log number = 15

PF = "FORM5SAGR1",- ;physical format for printout

RI = 0,- ;dynamic texts for printout

PA = 2,- ;printout activation = update

PU = 1,- ;picture/ printout at first update = yes

LD = BIT_MASK(1),- ;listing device = 1

RX = "")) ;data acquisition unit and group

#return %v_Object_Type_Process_Objects

#BLOCK_END
```

1.3 Custom commands

CMD_SELECT.TXT

```
; This is example program for Power Process Library switching device
selection service

;

; In: %SELECT_LN, text, The name of the selection object

; %SELECT_IX, int, The index of the selection object

; %SELECT_VALUE, int, The set value for selection
```

```
; %COMMAND_SELECTOR, text, OPEN_SELECT, CLOSE_SELECT

; %FORCED_OPERATION, boolean

; %FORCED_OPERATION_VALUE, int, 3 = interlocking bypass, 5 = synchrocheck
bypass, 7 = interlocking and synchrocheck bypass

; %STA_TYPE, text, The station type (SPA/ANSI/LON//RTU/RTU_E/IEC101/
IEC103/MODBUS_RTU/DNP_30/IEC61850 ...)

;

; Out: %NACK, int, The status code of the operation

;-------------------------------------------------------------------------
--------------------

@NACK = STATUS

#ERROR IGNORE

#IF %COMMAND_SELECTOR == "OPEN_SELECT" #THEN #BLOCK

@SELECT_VALUE = 0

#IF %FORCED_OPERATION #THEN #BLOCK

#CASE %FORCED_OPERATION_VALUE

#WHEN 3 @SELECT_VALUE = 10

#WHEN 5 @SELECT_VALUE = 20

#WHEN 7 @SELECT_VALUE = 30

#CASE_END

#BLOCK_END

#BLOCK_END

#ELSE #BLOCK

@SELECT_VALUE = 1

#IF %FORCED_OPERATION #THEN #BLOCK

#CASE %FORCED_OPERATION_VALUE

#WHEN 3 @SELECT_VALUE = 11

#WHEN 5 @SELECT_VALUE = 21

#WHEN 7 @SELECT_VALUE = 31

#CASE_END

#BLOCK_END

#BLOCK_END

#SET 'SELECT_LN':P'SELECT_IX' = %SELECT_VALUE

@NACK = STATUS
```

CMD_EXECUTE.TXT

```
; This is example program for Power Process Library switching device
execution service

;

; In: %EXECUTE_LN, text, The name of the selection object

; %EXECUTE_IX, int, The index of the selection object

; %EXECUTE_VALUE, int, The set value for selection

; %COMMAND_SELECTOR, text, OPEN_SELECT, CLOSE_SELECT

; %FORCED_OPERATION, boolean

; %FORCED_OPERATION_VALUE, int, 3 = interlocking bypass, 5 = synchrocheck
bypass, 7 = interlocking and synchrocheck bypass

; %STA_TYPE, text, The station type (SPA/ANSI/LON//RTU/RTU_E/IEC101/
IEC103/MODBUS_RTU/DNP_30/IEC61850 ...)

;

; Out: %NACK, int, The status code of the operation

;--------------------------------------------------------------------------
--------------------

@NACK = STATUS

#ERROR IGNORE

#IF %COMMAND_SELECTOR == "OPEN_EXECUTE" #THEN #BLOCK

@EXECUTE_VALUE = 0

#IF %FORCED_OPERATION #THEN #BLOCK

#CASE %FORCED_OPERATION_VALUE

#WHEN 3 @EXECUTE_VALUE = 10

#WHEN 5 @EXECUTE_VALUE = 20

#WHEN 7 @EXECUTE_VALUE = 30

#CASE_END

#BLOCK_END

#BLOCK_END

#ELSE #BLOCK

@EXECUTE_VALUE = 1

#IF %FORCED_OPERATION #THEN #BLOCK

#CASE %FORCED_OPERATION_VALUE

#WHEN 3 @EXECUTE_VALUE = 11

#WHEN 5 @EXECUTE_VALUE = 21

#WHEN 7 @EXECUTE_VALUE = 31

#CASE_END
```

```
#BLOCK_END

#BLOCK_END

#SET 'EXECUTE_LN':P'EXECUTE_IX' = %EXECUTE_VALUE

@NACK = STATUS
```

CMD_CANCEL.TXT

```
; This is example program for Power Process Library switching device
cancel service

;

; In: %CANCEL_LN, text, The name of the selection object

; %CANCEL_IX, int, The index of the selection object

; %STA_TYPE, text, The station type (SPA/ANSI/LON//RTU/RTU_E/IEC101/
IEC103/MODBUS_RTU/DNP_30/IEC61850 ...)

;

; Out: %NACK, int, The status code of the operation

;---------------------------------------------------------------------------
--------------------

@NACK = STATUS

#ERROR IGNORE

#SET 'CANCEL_LN':P'CANCEL_IX' = 1

@NACK = STATUS
```

1.4 Value conversion

GET_INDICATIONS.TXT

```
@RESULT=LIST()

#error ignore

;return values: 0 = intermediate, 1 = open, 2 = closed, 3 = faulty

#if 'INDICATION_DB_LN':POV'INDICATION_DB_IX' == 1 #then #modify
RESULT:V=LISt(INDICATION_OV=1)

#else #modify RESULT:V=LIST(INDICATION_OV=2)

;return values: 0 = interlocked, 1 = control allowed

#if 'OPEN_ILOCK_LN':POV'OPEN_ILOCK_IX' == 1 #then #modify
RESULT:V=LISt(OPEN_ILOCK_OV=0)

#else #modify RESULT:V=LIST(OPEN_ILOCK_OV=1)

;return values: 0 = interlocked, 1 = control allowed

#if 'CLOSE_ILOCK_LN':POV'CLOSE_ILOCK_IX' == 1 #then #modify
RESULT:V=LISt(CLOSE_ILOCK_OV=0)

#else #modify RESULT:V=LIST(CLOSE_ILOCK_OV=1)
```

```
;return values: 0 = control allowed, 1 = blocked

#if 'EXT_CNTR_BLK_LN':POV'EXT_CNTR_BLK_IX' == 1 #then #modify
RESULT:V=LISt(EXT_CNTR_BLK_OV=1)

#else #modify RESULT:V=LIST(EXT_CNTR_BLK_OV=0)

;return values: 0 = control allowed, 1 = not authorized to control the bay

#case 'RESULT_LIST:VBAY_LR_OBJECT_LN':POV'RESULT_LIST:VBAY_LR_OBJECT_IX'

#when 0 #modify RESULT:V=LISt(BAY_LR_OBJECT_OV=1)

#when 1 #modify RESULT:V=LISt(BAY_LR_OBJECT_OV=1)

#when 2 #modify RESULT:V=LISt(BAY_LR_OBJECT_OV=0)

#when 3 #modify RESULT:V=LISt(BAY_LR_OBJECT_OV=1)

#when 4 #modify RESULT:V=LISt(BAY_LR_OBJECT_OV=1)

#when 5 #modify RESULT:V=LISt(BAY_LR_OBJECT_OV=0)

#otherwise #modify RESULT:V=LISt(BAY_LR_OBJECT_OV=1)

#case_end

;return values: 0 = synchrocheck inhibits, 1 = control allowed

#if 'SYN_IND_LN':POV'SYN_IND_IX' == 1 #then #modify
RESULT:V=LISt(SYN_IND_OV=0)

#else #modify RESULT:V=LIST(SYN_IND_OV=1)

;#if 'RESULT_LIST:VSTA_LR_OBJECT_LN':POV'RESULT_LIST:VSTA_LR_OBJECT_IX'
== 1 #then #modify RESULT:V=LISt(STA_LR_OBJECT_OV=0)

;#else #modify RESULT:V=LIST(STA_LR_OBJECT_OV=2)

#return %RESULT
```

Index

U

V

Hitachi ABB Power Grids
Grid Automation Products
PL 688
65101 Vaasa, Finland

https://hitachiabb-powergrids.com/microscadax

Scan this QR code to visit our website

1MRK 511 466-UEN