

---

GRID AUTOMATION PRODUCTS

# **MicroSCADA X SYS600 10.2**

## External OPC Data Access Client







Document ID: 1MRK 511 471-UEN  
Issued: March 2021  
Revision: A  
Product version: 10.2

© 2021 Hitachi Power Grids. All rights reserved.



# Table of contents

<b>Section 1</b>	<b>Copyrights.....</b>	<b>3</b>
<b>Section 2</b>	<b>About this manual.....</b>	<b>5</b>
2.1	General.....	5
2.2	Use of symbols.....	5
2.3	Intended audience.....	6
2.4	Related documents.....	6
2.5	Document conventions.....	6
2.6	Document revisions.....	7
<b>Section 3</b>	<b>Safety information.....</b>	<b>9</b>
3.1	Backup copies.....	9
3.1.1	Taking backup copies.....	9
3.1.2	System backup.....	9
3.1.3	Application backup.....	9
3.2	Fatal errors.....	9
3.2.1	Handling.....	9
3.2.2	Status codes.....	10
<b>Section 4</b>	<b>Instructions.....</b>	<b>11</b>
4.1	Configuration.....	11
4.1.1	Base system configuration.....	11
4.1.1.1	Configuring the base system with System Configuration Tool .....	11
4.1.1.2	Configuring the SYS_BASCON.COM file.....	11
4.1.2	External OPC Data Access Client configuration.....	12
4.1.3	Starting from the command line.....	12
4.1.4	Configuration attributes.....	13
4.1.4.1	Node Attributes.....	13
4.1.5	Creating configuration.....	13
4.1.5.1	Changing External OPC Data Access Client settings.....	21
4.1.5.2	OPC item configuration.....	23
4.1.5.3	Manual mapping of process objects or custom station attributes.....	25
4.1.5.4	Adding device connection status item for system self supervision.....	35
4.1.5.5	Importing process objects from an external file prepared with OPC PO List Tool.....	36
4.1.5.6	Automated External OPC Data Access Client configuration.....	41
4.1.6	Importing and Exporting OPC Items configuration.....	43
4.1.7	OPC Process Object List tool.....	44
4.1.7.1	Introduction.....	44
4.1.7.2	Start-up.....	44
4.1.7.3	Browsing Process Objects.....	45
4.1.7.4	Setting filters.....	46

4.1.7.5	Setting a new station number.....	48
4.1.7.6	Including / excluding Process Objects.....	48
4.1.7.7	Changing application.....	48
4.1.7.8	Saving the Process Objects list.....	49
4.1.7.9	Formats for the output file.....	50
4.1.7.10	Viewing OPC signals.....	51
4.2	Starting External OPC Data Access Client.....	52
4.2.1	Starting from the command line.....	52
4.2.2	Starting from a SCIL program.....	53
4.2.3	Starting from External OPC Data Access Client Control Panel .....	55
4.3	Supervising DA Client.....	56
4.4	OPC quality flags.....	57
4.5	Quality handling with IEC 61850 OPC Server.....	60
4.6	Quality Handling, other OPC Servers.....	60
4.7	Buffered events.....	62
4.8	OPC error codes.....	63
<b>Section 5</b>	<b>Technical description.....</b>	<b>65</b>
5.1	OPC DA Client software components.....	65
5.2	Visual SCIL tools.....	65
<b>Index.....</b>		<b>67</b>

# Section 1      Copyrights

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

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

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

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

© 2021 Hitachi Power Grids. All rights reserved.

## Trademarks

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

## Guarantee

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

## Third Party Copyright Notices

List of Third Party Copyright notices are documented in "3rd party licenses.txt" and other locations mentioned in the 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      About this manual

### 2.1      General

There are two OPC Data Access Clients in SYS600. One is the internal OPC Data Access Client and the other is external. This manual describes the External OPC Data Access Client.

This manual provides thorough information on the OLE Process Control (OPC) Data Access (DA) client in SYS600. It describes how to configure the base system and the communication system to establish communication with slave devices managed by the OPC server. It also describes how to configure and use the External OPC Data Access Client software in the SYS600 application.

In addition to this configuration, the SYS600 application needs to be configured for process communication. For more information on this subject, see other manuals, for example SYS600 Application Objects and SYS600 System Objects.

The following features defined in the OPC specification are not supported by External OPC Data Access Client in SYS600:

- Public groups
- Blobs in item definitions
- Item Access Paths

These features are optional according to the OPC documents. External OPC Data Access Client is able to communicate with several OPC servers from different vendors, but several copies of the External OPC Data Access Client have to be started for it.

The following features are not supported by the External OPC Data Access Client Configuration tool:

- The list of available OPC Items is not shown, if the OPC server does not support the IOPCBrowseAddressSpace interface.
- OPC Item properties is not shown, if the OPC server does not support the IOPCItemProperties interface.

### 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 Intended audience

This manual is intended for installation personnel, administrators and skilled operators to support installation of the software.

## 2.4 Related documents

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

Name of the manual	Document ID
SYS600 10.2 System Configuration	1MRK 511 481-UEN
SYS600 10.2 System Objects	1MRK 511 482-UEN
SYS600 10.2 Application Objects	1MRK 511 467-UEN
SYS600 10.2 Status Codes	1MRK 511 480-UEN

The External OPC Data Access Client implementation is based on the following documents by the IEC Technical Committee 57:

Name of the document	Version
OPC Overview	Version 1.0. October 27, 1998
OPC Common Definitions and Interfaces	Version 1.0. October 21, 1998
OPC Data Access Custom Interface Standard	Version 2.04. September 5, 2000

## 2.5 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

## 2.6 Document revisions

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



# Section 3 Safety information

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

## 3.1 Backup copies

### 3.1.1 Taking backup copies

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

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

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

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

### 3.1.2 System backup

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

### 3.1.3 Application backup

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

## 3.2 Fatal errors

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

### 3.2.1 Handling

In case of a fatal error:

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



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

3. The data kept in the main memory at the moment of a fatal error is placed in the drwtsn32.log file with Windows 2003 Server, Windows XP and earlier. By default it is placed under %SYSTEMDRIVE%\Documents And Settings\All Users\Application Data\Microsoft\Dr Watson. Log and dump file paths can be checked with the drwtsn32 application. (Start -> run -> drwtsn32.exe). Analyze and copy the data in these files. Starting with Windows Server 2008 and Windows 7 the crash handling has changed. The location of the dump files can be read from the registry under the key HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps. The DumpFolder value tells the location of the dump files. Collect the data from this location.
4. Restart the system.

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

### 3.2.2 Status codes

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

# Section 4 Instructions

This section describes how to configure and start External OPC Data Access Client in SYS600.

## 4.1 Configuration

### 4.1.1 Base system configuration

The base system objects are defined either with System Configuration Tool or manually with SCIL commands in the SYS\_BASCON.COM file. It is recommended to use the configuration tool.

After making the changes either with the tool or by manually editing the file, the base system should be restarted to have the changes reflected in system. With a few limitations, the base system objects can also be defined and modified at any time when SYS600 is running.

#### 4.1.1.1 Configuring the base system with System Configuration Tool

To configure with the tool:

1. In the Tool Manager, select **System Configuration** and double-click **System Conf.**
2. In the tree view, right-click the **MicroSCADA Configuration** object and select **New**.
3. Select **LAN Link** and click **Insert**
4. Enter **1** in the **New Link Object Number** field, and click **OK**.
5. Right-click **Link 1** and select **New/Insert**.
6. Enter **2** as the IEC 61850 Node object number, and click **OK**.
7. Right-click **Node 2**, and select **New/Insert**.
8. Enter **2** as the new IEC 61850 Station object number, and click **OK**.
9. Save the configuration by selecting **Configuration/Save Active**.
10. Restart the base system.

#### 4.1.1.2 Configuring the SYS\_BASCON.COM file

To configure SYS\_BASCON.COM:

1. Define a base system object (SYS:B).
2. Define a link. The link type is TCP/IP.
3. Define a node.



Each OPC client instance is represented as a separate node. If several OPC client instances are used simultaneously, the same number of NOD objects must be defined in the SYS\_BASCON.COM file.

4. Define the SPA stations.

The definitions are made in the example below. For more information on the system objects, see SYS600 System Objects.

#### Example

The following is an example of the part of the SYS\_BASCON.COM file for communication via External OPC Data Access Client.

```
; ****
;
; COMMUNICATION LINK
#CREATE LIN:V = LIST(-
LT = "LAN"- ; TCP/IP connection
)
#CREATE LIN2:B = %LIN

; ****
;
; COMMUNICATION NODE
#CREATE NOD:V = LIST(-
LI = 2,-
SA = 203)
#CREATE NOD2:B = %NOD

; ****
;
; STATIONS
#CREATE STA:V = LIST(-
TT = "EXTERNAL",-  

ST = "SPA",-  

ND = 2,-  

TN = 2)
#CREATE STA2:B = %STA

; ****
```

## 4.1.2 External OPC Data Access Client configuration

This section describes External OPC Data Access Client configuration in accordance with the system, link, node and station numbers defined in SYS\_BASCON.COM.

For IEC 61850 systems the automatic External OPC DA Client configuration creation function of MicroSCADA SCL Importer tool should be used. The SCL Import tool can automatically create the full configuration based on the IEC 61850 system configuration during the configuration import to MicroSCADA.

OPC DA Client Configuration Tool (DAOCT) is used to create and modify the parameter file which is used by External OPC Data Access Client (DAOPCCL).

## 4.1.3 Starting from the command line

The External OPC Data Access Client configuration tool can be started from the command line using any of the following commands.

```
Dauct : Opens an empty file
dauct | "conf_file_name" : Opens the specified .ini file
dauct | "conf_file_name" | -hide
      -hide : Validation of the items occur without the UI and non-validated
              items are captured in the trace file(.trc)
```

```
C:\sc\prog\OPC_Client\DA_Config_Tool>dauct
C:\sc\prog\OPC_Client\DA_Config_Tool>dauct C:\sc\sys\active\sys_NT1.ini
C:\sc\prog\OPC_Client\DA_Config_Tool>dauct C:\sc\sys\active\sys_NT1.ini -hide
C:\sc\prog\OPC_Client\DA_Config_Tool>
```

*Figure 1: Starting External OPC Data Access Client configuration tool from the command line*

## 4.1.4 Configuration attributes

### 4.1.4.1 Node Attributes

#### RM Running mode

This attribute consists of a set of flags which control the behavior and functionality of External OPC DA Client instance. Each flag is one bit of this attribute. The bits are the following:

Bit 0:	When this bit is 0 and the used OPC Server reports that the quality of an item is bad, the Object Status (OS) of the corresponding item behaves as described in <a href="#">Table 6</a> . This is the default behavior. When this bit is 1, and the used OPC Server reports that the quality of an item is bad the Object Status behaves as described in <a href="#">Table 5</a> . The value of the RM is read from the configuration file, section [CPI], parameter RunningMode or it may be given as a command line argument.
Bit 1:	When this bit is 0 and the used OPC Server is <u>not</u> SYS600 IEC61850 OPC Server, process object attribute QL (Qualifier) is not updated. When this bit is 1, QL (Qualifier) is updated with the OPC Quality of the updated item when OPC Server is other than SYS600 IEC 61850. See chapter 'Quality handling, other OPC Servers' for more details.
Data type:	Integer
Value:	0..65535
Index range:	no index
Default value:	0
Access:	Read, Write

## 4.1.5 Creating configuration

The manual configuration steps described next in this manual are not required for IEC 61850 systems when MicroSCADA SCL Importer automatic configuration is used. However the Configuration Tool can be used to verify and modify the automatically created configuration.

When configuring manually the following actions should be executed via the Configuration Tool to create the configuration file:

1. Locate OPC servers installed in the system and allow the user to select the desired server, which provides access to OPC-enabled devices. The **Server Properties** dialog is used for this purpose (main menu item **File/New**).
2. Define the parameters for communication External OPC Data Access Client with SYS600 base system via the CPI interface, in the **CPI Node Properties** dialog (main menu item **File/New/Next**) and should be filled as follows:

Own Node Number	Should be the same as the value of the created node attribute of the communication NODE object (NOD: V). See the SYS-BASCON.COM file. For example: #CREATE NOD2:B = %NOD The communication node created is NOD2, therefore the own node number is 2.
Own Station Number	Should be the same as the value of the SA attribute of COMMUNICATION NODE object (NOD:V). See SYS_BASCON.COM.
System Messages Enabled (SE)	Should be 4 in new IEC 61850 systems. In IEC 61850 systems made with MicroSCADA 9.2SP2 or earlier, or with OPC servers other than the MicroSCADA IEC 61850 OPC server, the value should be 1.
Running Mode (RM)	Should be 1 (bit 0 = 1) in new systems. If a backward compatible quality handling is needed in an existing system, value 0 (bit 0 = 0) for RM may be used. Section <a href="#">Section 4.4</a> describes the differences between these values.
Base Node Number	Should be the same as the value of the ND attribute of BASE SYSTEM OBJECT (SYS:B). See SYS_BASCON.COM.
Base Station Number	Should be the same as the value of the SA attribute of BASE SYSTEM OBJECT (SYS:B). See SYS_BASCON.COM.
Base Application Number	Application Number in SYS600, where the connection is created.
Base IP Address	SYS600 TCP/IP address.

3. Define parameters for communication between External OPC Data Access Client and another SYS600 base system (secondary), if the connection with the primary base system has been lost. These parameters are optional. External OPC Data Access Client has to be reconnected to the primary SYS600 base system, if the list of the secondary system parameters is not defined. These parameters are defined in the **CPI Node Properties** dialog of the configuration tool and should be used as follows:

Back Node Number	Should be the same as the value of the ND attribute of BASE SYSTEM OBJECT (SYS:B). See SYS_BASCON.COM of the secondary SYS600 base system.
Back Station Number	Should be the same as the value of the SA attribute of BASE SYSTEM OBJECT (SYS:B). See SYS_BASCON.COM of the secondary SYS600 base system.
Back Application Number	Application Number in secondary SYS600, where connection is created.
Back IP Address	Secondary SYS600 TCP/IP address.
MS Max Reconnections	The number of reconnection attempts to the primary SYS600. After these attempts are expired, External OPC Data Access Client is connected to the secondary SYS600 base system.

4. Define SPA stations, which are supported via External OPC Data Access Client. The SPA station is defined in the Add STA dialog (main menu item **Edit/Add STA**) and the appropriate fields should be filled as follows:

Node Name	The name of the station to be used in configuration.
Unit Number	Specifies the unit number of the station. Should be the same as the value of the TN attribute of SPA STATION OBJECT (STA:V).
Read/Write Timeout	Timeout (ms) to be defined for all read/write operations to/from the OPC server for each STA unit. When this parameter is 0, these operations never time out.
In Use	Specifies whether the station is in use or not.

### Special parameters

There are parameters which are not supported by the configuration tool but which can be added to generated configuration file manually if needed. These parameters become effective when configuration is loaded. Saving the configuration from configuration tool will overwrite the settings made manually.

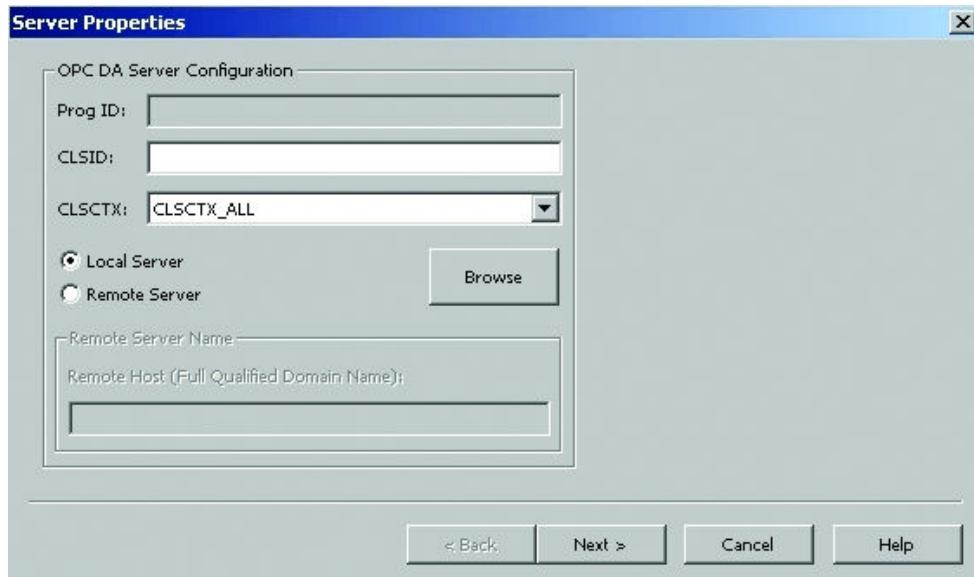
Section in .ini file	Parameter	Description
[CPI]	CpiTimeout	If present, it defines the response timeout to data updates to MS application in seconds. If not present, timeout is 50 seconds.
[OPC]	HistoricalEventsAsBuffered	If set to 'Yes', all data updates OPC will be sent to MS Application as non-buffered and this lead to duplicate event. If set to 'No' or is not defined, event handling is as described in chapter 'Buffered events'

### Example

The following is an example of External OPC Data Access Client configuration for SYS600. Before starting this example, the SYS600 base system should be configured according to the example in [Section 4.1.1](#). The Matrikon OPC Simulation server should be installed on the computer. The OPC server can be obtained from <http://www.matrikon.com/opc>.

When the Matrikon OPC Simulation Server has been installed the following actions should be executed:

1. Start the configuration tool using Windows Explorer, command line or a desktop short cut to the executable file. The file name of the program is DAOCT.EXE.
2. To open the **Server Properties** dialog, select **File/New** from the menu.



*Figure 2: Server Properties dialog*

3. To open the **Available Servers** dialog, click the **Browse** button.

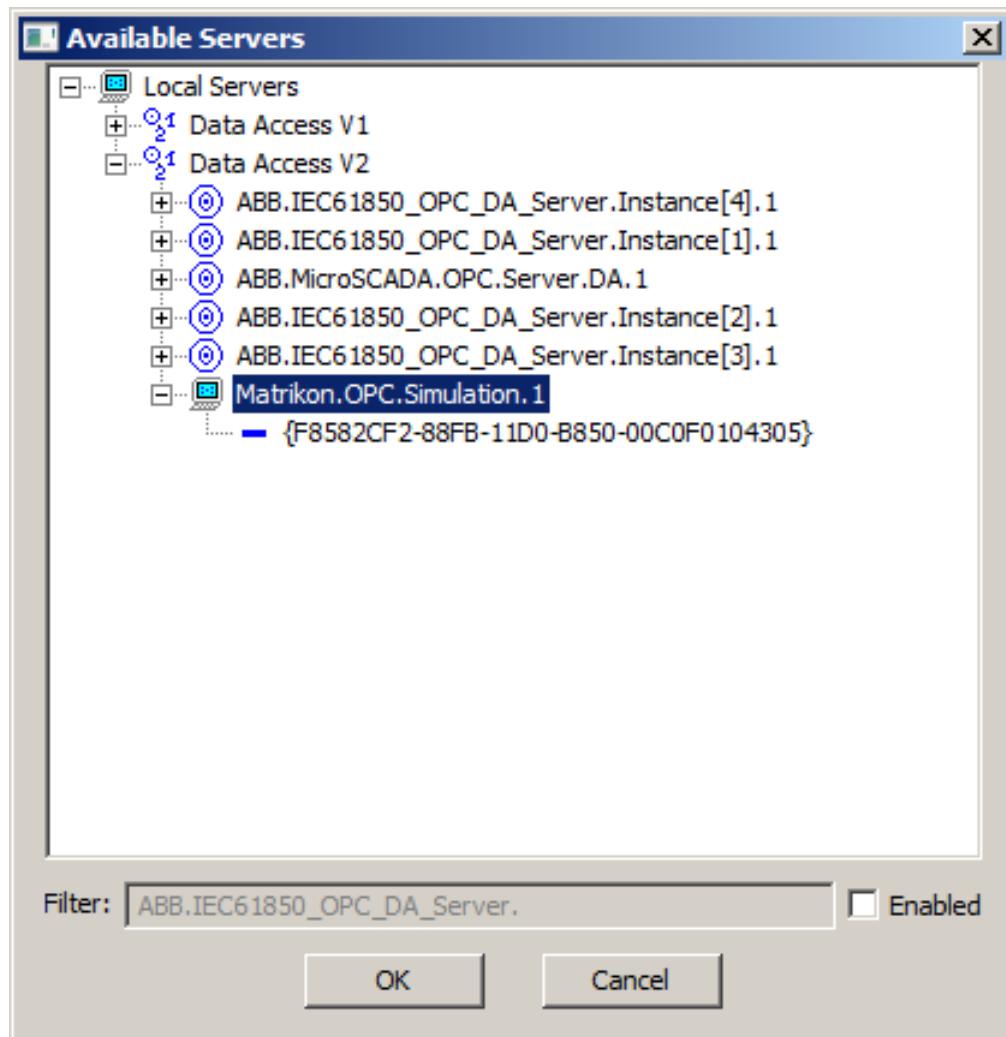
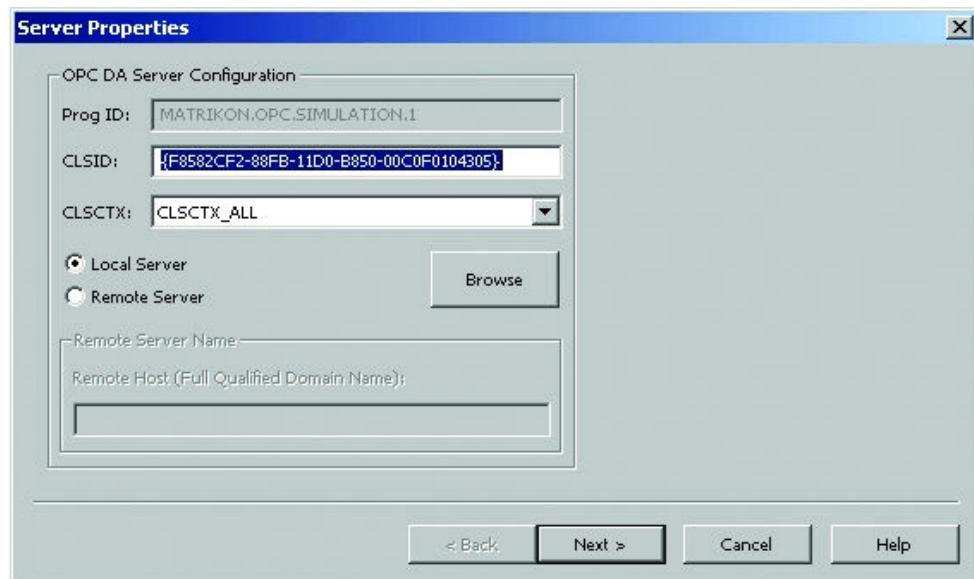


Figure 3: Available Servers dialog



A filtering string (case sensitive) in the Filter dialog can be used to filter the displayed OPC DA server ProgIDs. Enable or disable the filter using the Enabled check box. The filter settings are stored.

4. Double-click on **Matrikon.OPC.Simulation.1**.  
The following information is displayed in the **Server Properties** dialog:

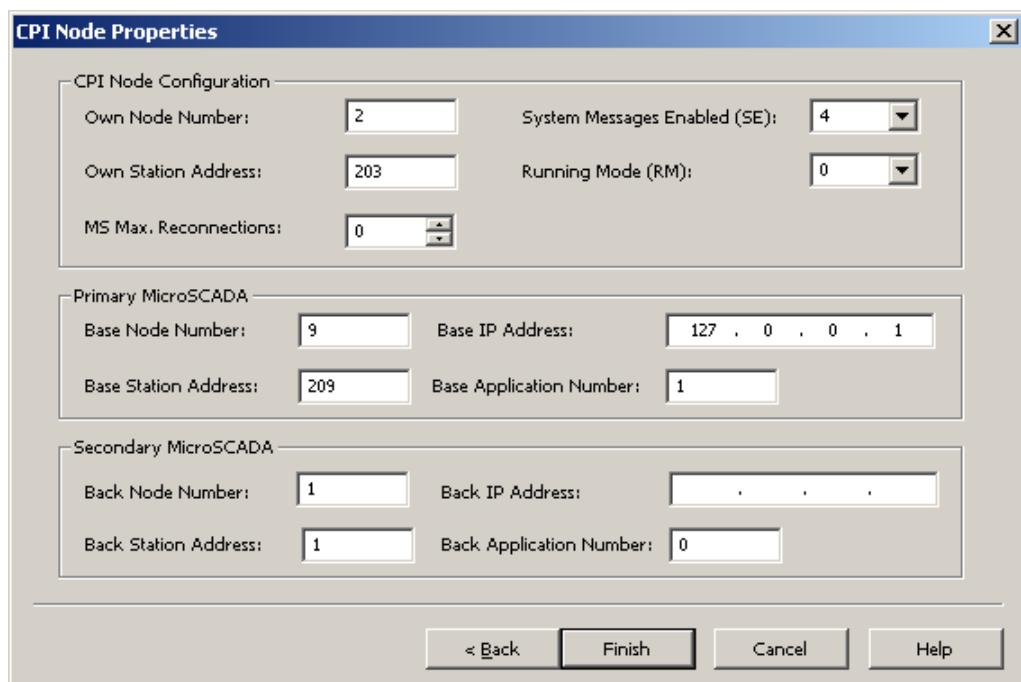
*Figure 4: Server properties dialog*

When the Prog ID & CLSID fields are filled in, the OPC server is configured.

5. Click **Next** to open the **CPI Node Properties** dialog.
6. Fill in the dialog fields as shown in [Figure 5](#)



If the External OPC Data Access Client and the OPC server are running on the same computer, the Base IP address of the CPI node should be 127.0.0.1. Otherwise, the communication between SYS600 and the OPC server may not work properly.

*Figure 5: CPI Node Properties dialog*

The SYS600 communication is now configured.

7. Click **Finish** to open the main window.

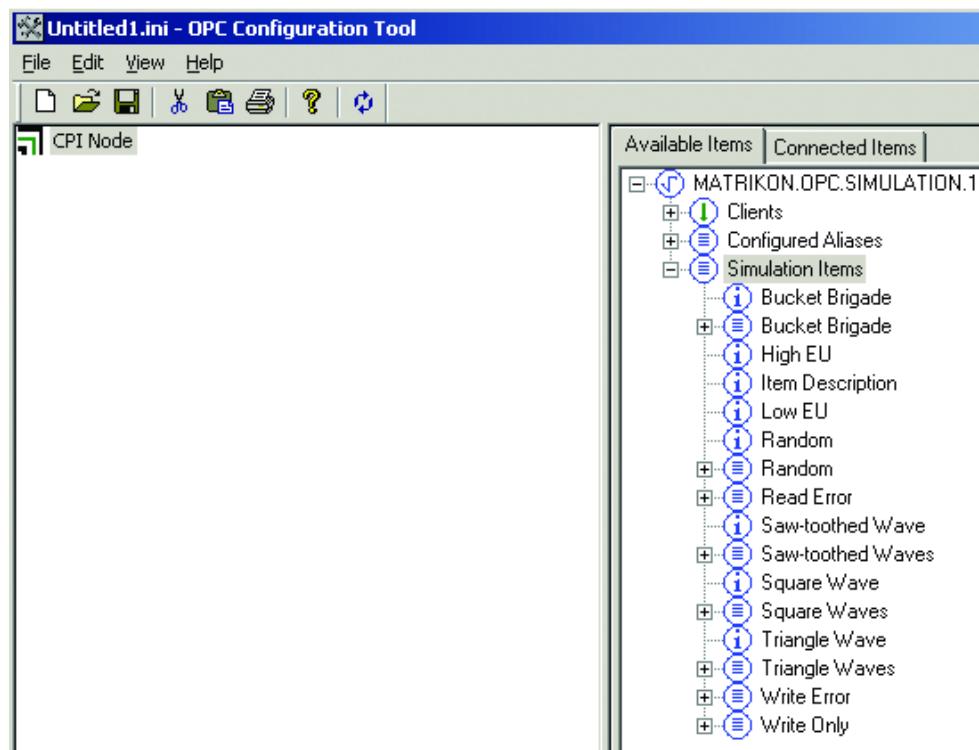
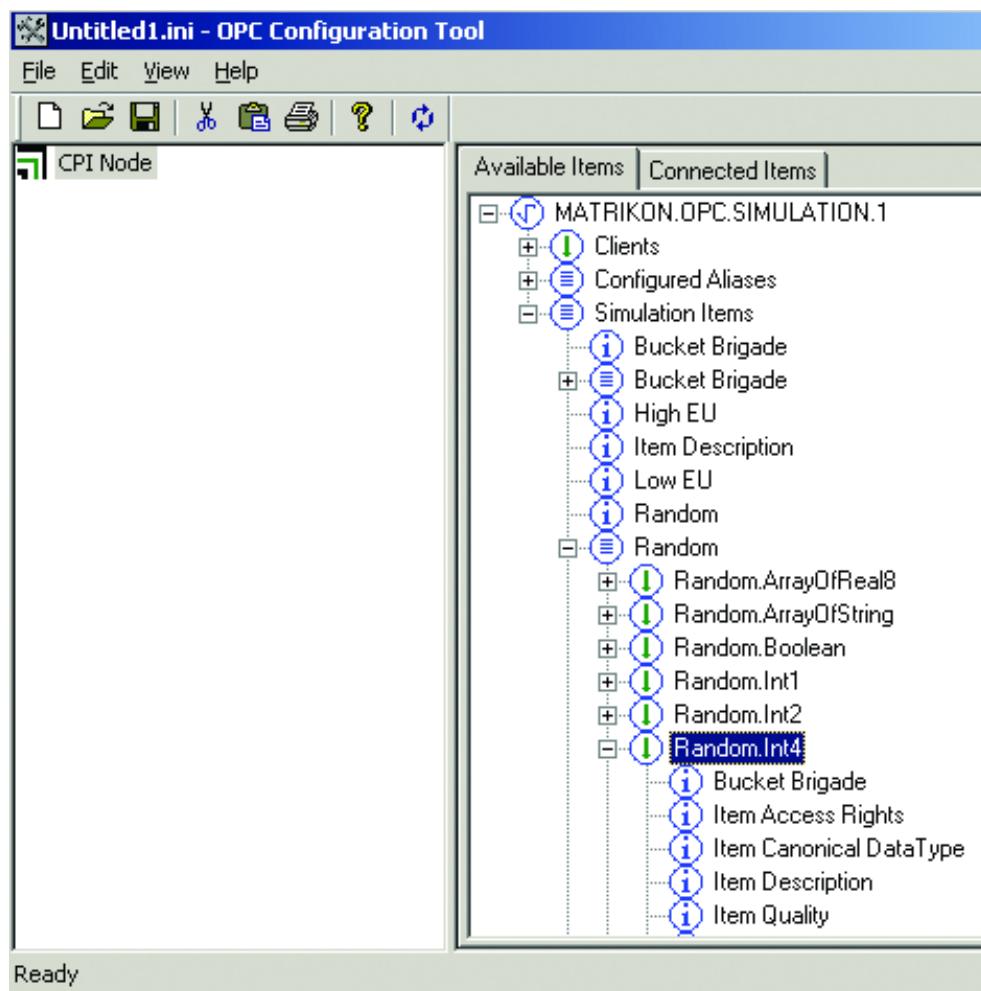


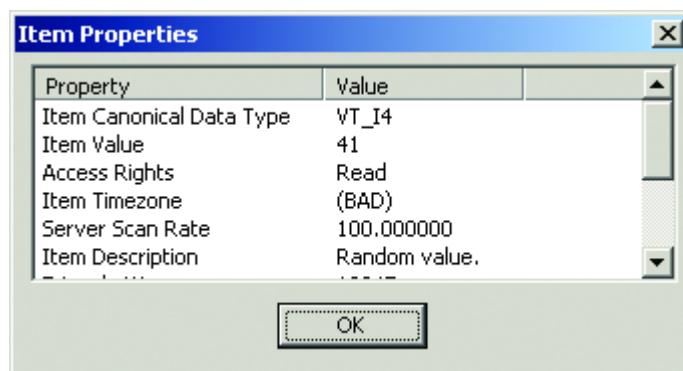
Figure 6: OPC Configuration Tool main window

Expand the **Simulated Items** branch in the tree on the right side of the main window. When this is done, the other branches and properties are shown in the tree. Expand the branch named **Random**, and any **OPC Item** that is shown under the branch. The main window should look like this:



*Figure 7: OPC Configuration Tool main window after expanding the branches*

Random.Int4 is an OPC Item in the window above. The OPC Items are represented on the Available Items page by various icons. A green downward arrow means that the OPC Item has read access right, whereas write access right is denoted by a red upward arrow. The tree leaves under Random.Int4 are OPC Item properties. To see the values of the item properties available at the moment, right-click the **Random.Int4** item and select **Properties** in the pop-up menu to open the **Item Properties** dialog.



*Figure 8: Item Properties dialog*

To close the dialog, click **OK**.

8. Select **Edit/Add STA** in the menu bar to open the Add STA dialog.

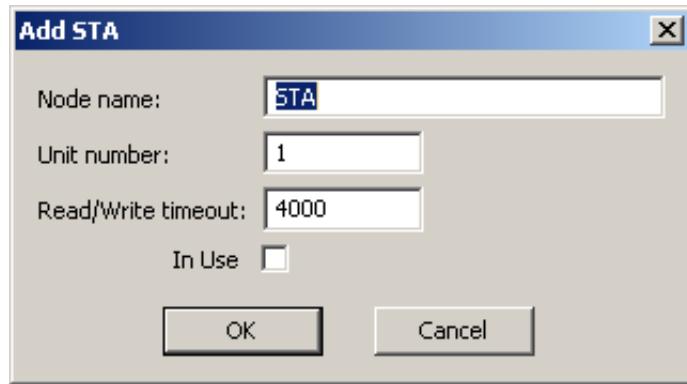


Figure 9: Add STA dialog

9. Change the Unit number value to 2 and click the In Use check box.
10. Click OK to open the main window.

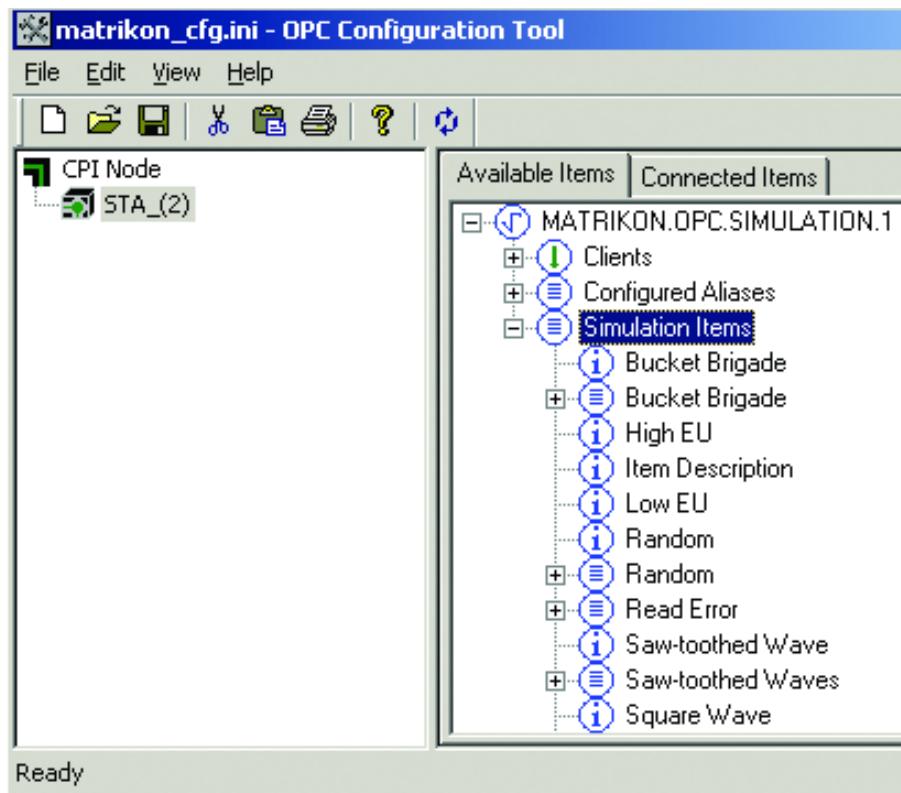


Figure 10: OPC Configuration Tool window after adding STA

The External OPC Data Access Client has now been configured for communication with Matrikon OPC Simulation Server and with the SYS600 base system.



Save the current configuration in the file matricon\_cfg.ini (Menu **File/Save**). It is used in a later example for OPC Items configuration.



The STAs are arranged according to the STA numbers after the configuration is saved.



If the filter options **Show valid item** and **Show invalid items** are chosen, the **Save** icon remains disabled. The **Save** icon is enabled only if the **Show All** filter option is selected.

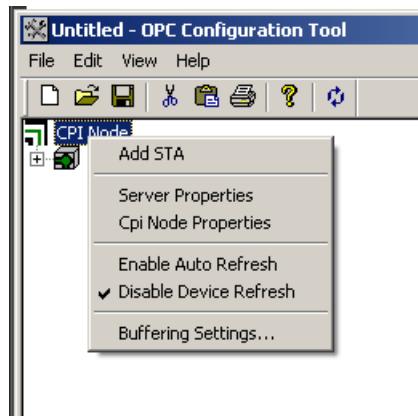


If Notification Window message  
0020 ? TCPIP New Socket For Node (SA: 208, from IP: 10.10.10.120, to IP:  
127.0.0.1)  
is displayed, it indicates a configuration error. This occurs when more than one External OPC DA Client instances with the same Node Number and Station Addresses are connecting to the same base system or the External OPC DA Client is running in the same host as MicroSCADA and the Node Name of the OPC DA client node is not empty.  
To prevent this, make sure that the Node number and Station Address for each OPC DA client instance are different when defining the configuration and these settings are consistent with SYS\_BASCON.COM configuration for base system. If the OPC DA client instance is running on the same host as MicroSCADA, check that the node name of the OPC DA Client node is empty (NOD'n':BNN = "") and the Base IP address in the CPI node properties dialog should be the localhost address 127.0.0.1 (recommended, default) or the same as the IP address of the host computer.

#### 4.1.5.1 Changing External OPC Data Access Client settings

When the OPC server and the CPI node properties have been configured, they can be changed in the same session, or in the future sessions of the OPC DA Client Configuration Tool.

To open the pop-up menu for changing the Server and CPI properties, right-click the **CPI Node** (root of the tree) on the left side of the main window.



*Figure 11: Changing the Server and CPI properties*

Here new STA objects can be added and the Server and CPI node properties can be changed by selecting the corresponding item in the pop-up menu.

##### Enable Auto Refresh

This flag is used to configure active state of OPC items mapped to output process objects. This setting should not be selected in IEC 61850 systems when connecting to IEC 61850 OPC Server.

If this setting is selected, the OPC items are created in active state. When items are in active state all data changes are received from the OPC Server. This may cause unnecessary events in some configurations.

If this setting is not selected, the OPC items are created in inactive state and will not receive data updates. This is the default configuration as data updates to output process objects are discarded by SYS600.



At run-time, it is possible to force External OPC Data Access Client to refresh items from OPC server by setting the UP attribute of the corresponding STA unit to 1, for example #SET STA2:SUP=1.

### Disable Device Refresh

This flag is used to configure OPC item data refresh functionality of the External OPC Data Access Client.

If this setting is selected, the refresh requests from the External OPC Data Access Client to the OPC Server are set to type 'cache'. Cache refresh should be used when the OPC Server provides up-to-date data. With cache refresh requests (e.g. UP attribute of the station object is written) the data will be requested from the cache of the OPC server and not from the IED. OPC server is responsible for keeping its data up-to-date in its cache.

If this setting is not selected, the refresh requests from the External OPC Data Access Client to the OPC Server are set to type 'device'. Device refresh should be used when the OPC Server may provide outdated data and needs to be updated from the IED. Device refresh is used to trigger the OPC Server to refresh the item values explicitly from the IED and update the values. This causes more processing for the IEDs and additional traffic to the IED. Depending on the IEDs this may cause additional events as there can be changes in timestamps of IED values. Device refresh can be used with 3rd party OPC Servers which require updating of the process values of the underlying devices.



This setting should always be selected in IEC 61850 systems when connecting to IEC 61850 OPC Server. Since the IEC 61850 OPC Server always has an up-to-date image of the connected IEDs process values, cache refresh can be used.

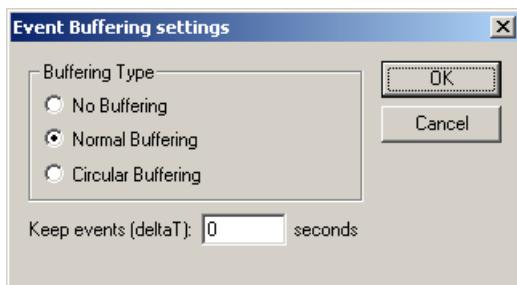


Figure 12: Event Buffering settings

#### No Buffering

When No Buffering is selected, External OPC Data Access Client does not buffer the events.

#### Normal Buffering

When Enable Normal Buffering is selected, External OPC Data Access Client buffers the events so that the latest update is always stored into the buffer independently of the process object type. Normal buffering is the default setting.

#### Circular Buffering

When Enable Circular Buffering is selected, External OPC Data Access Client buffers the events so that changes of switching device indications are always stored into the buffer as their own entries, which means that all the state transitions are stored.

With measurement updates, the latest update is always stored into the buffer by default but if setting 'QueueUpdatesOfAllDataTypes' in [OPC] section of the loaded configuration file exists and is set to 'Yes', all incoming analog and pulsecounter updates are queued.

In the case of IEC 61850 HSB systems, Circular Buffering should be used. However, all the changes of the switching device indications are stored during the communication break as long as the size of the buffer has not been exceeded. For more information, see [Section 4.1.5.1.7](#).

#### Keep events (deltaT)

This parameter determines the maximum time for keeping events in the buffer. When this time has elapsed and the event has still not been sent to SYS600, the event is removed from the buffer. By default, deltaT is equal to zero, which denotes infinity.



External OPC Data Access Client changes the date of all coming events from the OPC server dated earlier than 01.01.1980 12:00:00. Such events are sent to SYS600 with BAD\_TIME status.

#### Maximum buffering size and behavior at run-time

The maximum buffer size for the Normal and Circular buffering is the same. It is calculated as the number of all indications (number of configured process objects of Single and Double Indication types) multiplied by 20 plus the number of all items. The formula is the following:

$$(\text{number of indications} \times 20) + \text{number of items}$$

If the number of indications \* 20 is less than 1000, the default buffer size 1000 is used instead.

As soon as the connection with the OPC server is established, External OPC Data Access Client starts buffering all coming events from the server. Circular buffering is used only for Single and Double Indication types if setting 'QueueUpdatesOfAllDataTypes' is not used. All coming events of these types are pushed into the tail of the buffer and when it is possible, External OPC Data Access Client sends them to SYS600 application starting from the head of the buffer (oldest events first). An event is not removed from the buffer until an acknowledgement is received from SYS600. When the length of the buffer becomes greater than the maximum allowed size, the oldest event is removed from the head of the buffer. The buffering is active even if there is no actual connection with SYS600. For all other types of process objects, External OPC Data Access Client only keeps the latest value.

In MicroSCADA version 9.3 or newer, if the main application is in COLD state and the phase of the shadowing is HOT\_RECEIVE, the data from the External OPC DA client is buffered in the base system. In practice, this means that in a takeover situation where the shadowing starts quickly, the period of buffering in the OPC DA Client is short.

For more information on buffering in the HSB System, refer to SYS600 9.3 System Objects, Section Redundancy, Subsection *HSB Systems*.

### 4.1.5.2 OPC item configuration

External OPC Data Access Client is able to represent OPC Items with the following SYS600 object types depending on the OPC Item access rights:

1. Input process objects

*Table 1: Input process objects for the OPC Items with read-only or read-write access rights*

Station	Object type	Station	Object type
SPA	Single Indication	IEC 61850	Single Indication
SPA	Double Indication	IEC 61850	Double Indication
SPA	Digital Value	IEC 61850	Digital Value
SPA	Analog Value	IEC 61850	Analog Value
SPA	Pulse Counter	IEC 61850	Pulse Counter
SPA	Bit Stream	IEC 61850	Bit Stream

When Bit Stream process objects are used to receive the OPC items of array data type from the OPC server, additional handling in the SYS600 application is required.

The following example demonstrates how the original array of numeric values, separated by commas, could be extracted from the value of the Bit Stream process object:

```
%csv_text = TYPE_CAST( 'LN':POV'IX', "TEXT" )
@text_vector = SEPARATE(%csv_text, ",")
@analog_vector = DEC_SCAN(%text_vector)
```

## 2. Output process objects

*Table 2: Output process objects for the OPC Items with write-only or read-write access rights*

Station	Object type
SPA	Object Command
SPA	Digital Setpoint
SPA	Analog Setpoint
IEC 61850	Object Command
IEC 61850	Digital Setpoint
IEC 61850	Analog Setpoint

OPC servers use standard OLE data types for OPC items. There is no strict correspondence between OPC server data types and types of process objects, because all data coming from the OPC server is converted by External OPC Data Access Client into the data type requested by SYS600 for a certain kind of process object. SYS600 process object types are represented by the following OLE data types:

- Single Indication, Double Indication, Block of Indications, Digital Input and Digital Setpoint are mapped to VT\_UI2
- Analog Input and Analog Setpoint are mapped to VT\_R4
- Pulse Counter is mapped to VT\_UI4
- Binary Command is mapped to VT\_BOOL
- Bit Stream is mapped to VT\_ARRAY of the source type.
- Req. type field contains a type which is requested for the selected type of process object.
- Canonical type field contains data type of the selected OPC item.  
This information can be found in the **Item Editor** dialog of the External OPC Data Access Client Configuration tool. It is not necessary that Req. type is equal to Canonical type, but it should be possible to make a conversion between these types.

## 3. Custom STA object attributes (STA:S) can represent the OPC Items with read-write access rights.

The OPC data items with read-write access rights can be represented by two SYS600 objects: input process object for reading data and output process object for writing data. Whereas, the read only and write only OPC Items can just be mapped in one input or output SYS600 object.

The configuration tool checks for the rules when creating or updating the configuration (parameter file).

In addition, OPC data items can be used to represent alarm flags of SYS600 process objects of Digital Input and Analog Input types, which have already been connected to OPC Items. These alarm flags indicate that one of the alarm limits is exceeded. There are four alarm limit types in SYS600: Low Alarm, Low Warning, High Warning and High Alarm. Separate alarm flags are used for each of them. There is a one-to-one correspondence between OPC Items and alarm flags. OPC Items cannot be assigned to just some alarm flags of a certain SYS600 input process object and not to others. Only a complete set of the four alarm flags can be used.

This section describes the mechanism of mapping OPC Items to STA:S attributes and process objects. In the configuration tool, there are two methods of mapping:

- Manual mapping of process objects or custom attributes (STA:S) on OPC Items. The **OPC Item Editor** dialog should be used for this purpose (main menu item **Edit/Add Item**).
- Importing process objects from the list file provided by the OPC PO List tool. The **Import Process Objects** dialog should be used for this purpose (main menu item **File/Import Proc List**).

Before mapping SYS600 process objects or custom attributes, the appropriate OPC Groups should be created. An OPC group is a logical unit, which provides a way for the users to organize data. An OPC Item must be included in an OPC group before it can be accessed by the OPC clients. All access to OPC Items occurs via groups. The **Add Group** dialog is used to create the OPC group (main menu item **Edit/Add group**). The following fields of this dialog should be filled in:

<b>Update Rate</b>	The minimum period of time (in ms), in which the OPC server sends the updated OPC Item values of the items included in the group to the client. In addition, it is the maximum period of time, in which the OPC server updates its internal cache of the item values from the process devices. In SYS600, the process objects mapped to the items in the group cannot be updated faster than this parameter defines. The signal changes shorter than specified value can get lost. For the measurements (analog signals). It is typical to use an update rate of 1000 ms, but for the state indications an update rate of 0 ms should be always used. The default value is 0 ms.
<b>Time bias</b>	The initial time bias (in minutes) for the group. The purpose of this parameter is to indicate the time zone, in which the data (process object changes) was collected. The time bias is the difference between Coordinated Universal Time (UTC) and local time. The timestamp parameter for SYS600 is set according to the time zone specified in this parameter, i.e. timestamp = UTC + Time Bias.
<b>Percent deadband</b>	Another parameter defined for each OPC group. It should be in the range from 0 to 100. This parameter defines how sensitive the updating items process on the OPC server is to small fluctuations of the item value. The value 0 tells the OPC server to update its catch and to send notifications to the client when any small modification of the item value occurs. Higher values of this parameter allow avoiding unnecessary data sending after small fluctuations of item values. Not all OPC servers support this feature. If an OPC server supports this feature, deadband applies only to those items in the group that have dwEUType attribute equal to 1—"Analog". For other items, and if the OPC server does not support this feature, the percent deadband parameter is ignored. See the "OPC DA 2.0x Specification" document on the OPC foundation Web site: <a href="http://www.opcfoundation.org">http://www.opcfoundation.org</a> .

#### 4.1.5.3 Manual mapping of process objects or custom station attributes

The **Item Editor** dialog of the configuration tool is used for mapping the SYS600 objects to the OPC data items, see [Figure 13](#). When the appropriate OPC groups are created, mapping of the SYS600 process objects or custom attributes to the OPC Items can be started. Select an appropriate group on the left side of the configuration tool's main window and select **Edit/Add Item** in the main menu to open the **Item Editor** dialog.

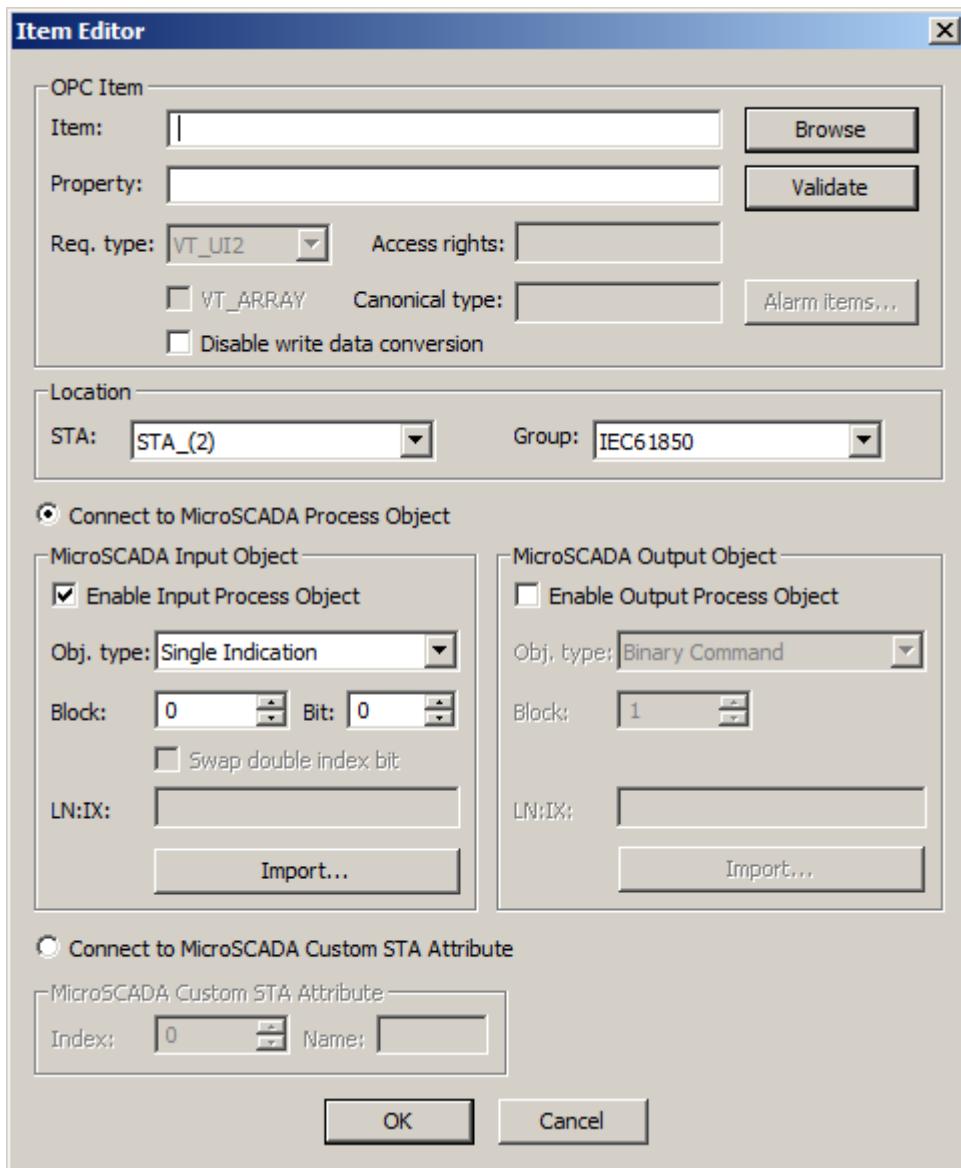


Figure 13: Item Editor dialog

An OPC Item or item property can be mapped to SYS600 Process Objects or to an STA object attribute.

The following fields of this dialog should be filled in before the mapping can be done:

<b>Item</b>	The name of the OPC Item. This field may be typed in manually or selected from the OPC server address space by the Browse button. The Browse feature is accessed, if the appropriate OPC server supports IOPCBrowseServerAddressSpace interface.
<b>Property</b>	The name of the OPC Items property. This field may be typed in manually or filled in automatically by selecting an OPC Item property in the Browse dialog. This field is empty when an OPC Item, not an OPC Item property, is being mapped.
<b>Req.Type</b>	The requested type of appropriate SYS600 object value. The value of the requested type is sent to SYS600. This field may only be changed for STA:S attributes, that is when radio button "Connect to SYS600 Custom STA Attribute" is selected. For process objects Req. type, is predefined by the dependence of the process object type (the PT attribute of the process object).
<b>Disable write data conversion</b>	Option for disabling local data conversion prior to writing data over OPC DA. This option is separately configurable for all items.

When Connect to SYS600 Process Object is selected:

<b>Enable Input Process Object</b>	This option should be checked in order to map the OPC Item to the SYS600 input process object. This check box is dimmed for OPC Items without read-access right.
<b>Obj. type</b>	Is the type of SYS600 input process object.
<b>Block</b>	Should be the same as the value of an OA attribute of the SYS600 input process object.
<b>Bit</b>	Should be the same as the value of an OB attribute of the SYS600 input process object. This field only has a meaning for SPA/ Single Indication and SPA/ Double Indication types.
<b>Enable Output Process Object</b>	This option should be checked in order to map an OPC Item to the SYS600 output process object. This check box is dimmed for OPC Items without write-access right.
<b>Block</b>	Should be the same as the value of an OA attribute of the SYS600 output process object.

When Connect to SYS600 Custom STA Attribute is selected:

<b>Name</b>	The name of the STA:S attribute. For example, if Name = AA, the user can address this attribute from SCIL as STA2:SAA.
<b>Index</b>	The STA:S attribute index. For example, if Name = AA and Index = 1, the user can address this attribute from SCIL as STA2:SAA1. The index range is 0..65535. In case the attribute is accessed from SCIL without index value, a default index = 0 is given by the base system.



In case the OPC item name connected to a custom STA attribute contains a substring 'Attributes\Transparent XSAT', a read operation to the same attribute must be performed before a new write operation is accepted. Otherwise a specific error code 13260 SPAP\_ATTRIBUTE\_IS\_WAITING\_FOR\_READ is returned to SCIL. This feature becomes visible with the IEC 61850 OPC Server.

SYS600 process objects can be imported from an external file produced by OPC PO List Tool by clicking the **Import** button. If the process objects have never been imported before, the File Chooser dialog opens where the appropriate file is selected and opened (click **Open**). Otherwise the process objects are read from the previously used file.

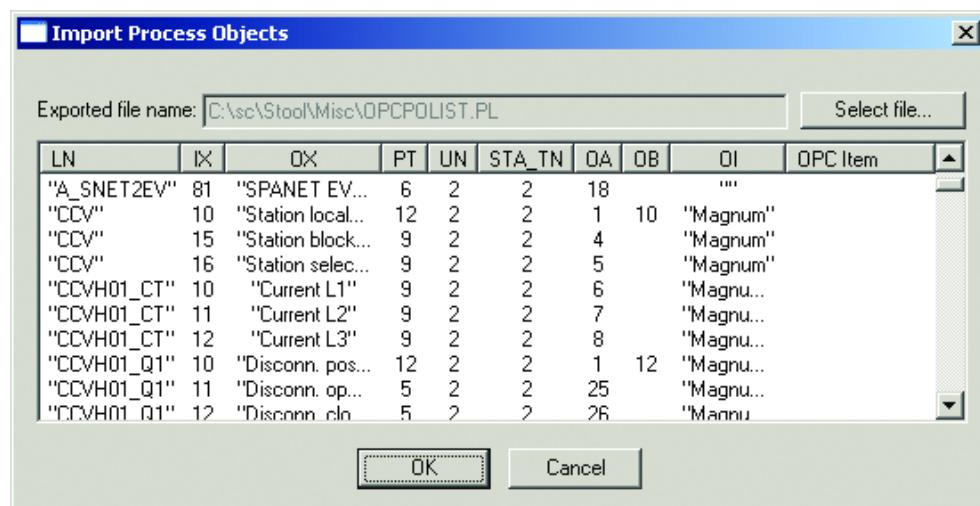
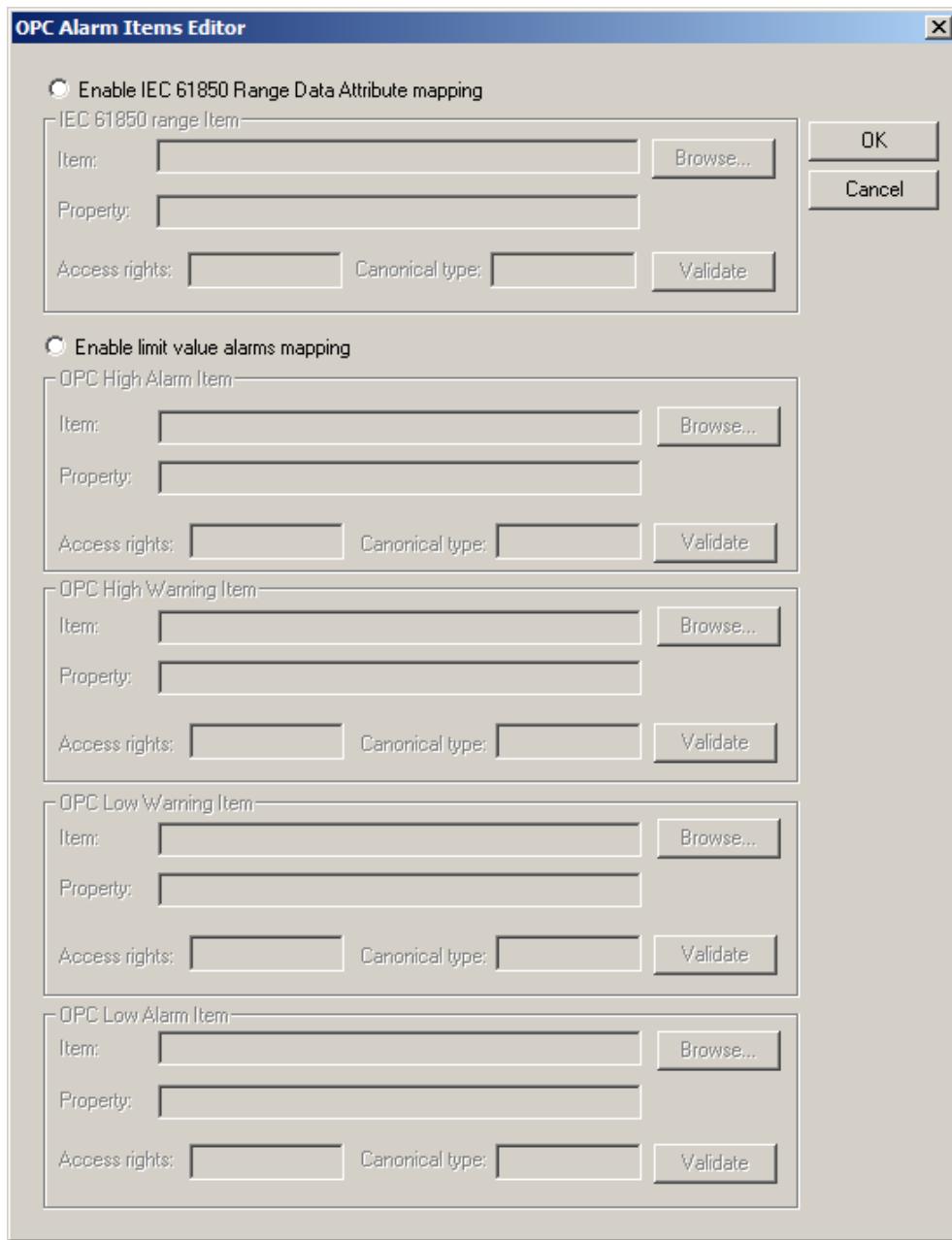


Figure 14: Import Process Objects dialog

When an object to be imported is selected and **OK** is clicked, the appropriate fields of the **Item Editor** dialog are filled in automatically. For more information on importing SYS600 process objects, see [Section 4.1.5.4](#).

When an OPC Item is connected to a SYS600 input process object of the Analog Input or Digital Input type, the **Alarm Items** button is enabled. Now, the alarm flags of this object can be mapped to the other OPC Items or to IEC 61850 Range Item.

Click the **Alarm Items** button to open the **OPC Alarm Items Editor** dialog, see [Figure 15](#).



*Figure 15: OPC Alarm Items Editor dialog*

To enable the mapping to IEC 61850 Range Item, the **Enable IEC61850 Range Data Attribute Mapping** radio button should be enabled.

To enable the mapping of alarm flags, the Enable limit value alarms mapping radio button should be enabled. The meaning of the dialog field is identical to the Item and Property fields of the **OPC Item Editor** dialog, and they can be selected from the OPC server address space by clicking one of the **Browse** buttons. It is not possible to define less than four alarm flags. After all the Alarm Items are filled in, click **OK**.



The limit value alarms do not correspond to IEC 61850 limits. For IEC 61850 systems using IED measurement supervision with range reporting the **Enable IEC 61850 Range Data Attribute Mapping** option should be used.

When all the fields of the **Item Editor** dialog have been filled in, the mapping is completed and **OK** can be clicked. The newly created and mapped item is shown under the appropriate STA object, under the appropriate OPC Group.



Some of the OPC Item properties cannot be validated in the OPC server. If an item property has been selected in the **Item** field, after clicking **OK**, a warning dialog "The OPC Item or property is NOT valid. Continue?" may appear. If this occurs, the item property can still be mapped to a custom STA attribute. Mapping it to a process object causes an error on this item property in the OPC client instance, which was started by this configuration.

### Example

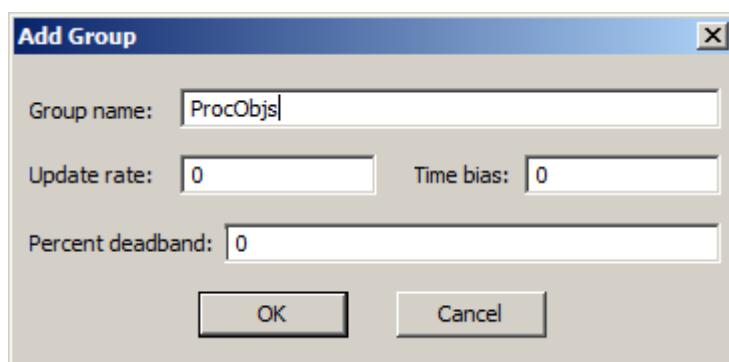
The following is an example of mapping IEC 61850 measurement data with IED limit checking to SYS600 process object through SYS600 IEC 61850 OPC Server.

Before starting this example, the following requirements should be fulfilled:

- IEC 61850 OPC Server is configured and connected to an IED supporting measurement limit supervision.
- The SYS600 base system should be configured as in the example in [Section 4.1.1](#).
- In the SYS600 Object Navigator, create an analog input process object with block address 45.

Once these actions are done, the following can be done:

1. Start the configuration tool (DAOCT.EXE).
2. Create a configuration file from the main menu **File/New**. Configure server properties for an IEC 61850 OPC Server (for example ABB.IEC61850\_OPc\_DA\_Server.Instance[1].1).
3. Add STA 2.
4. Right-click the STA\_(2) object and select **Add group** to open the **Add Group** dialog.



*Figure 16: Add Group dialog*

5. Set the group name to ProcObjs and click **OK**.

Now the ProcObjs object is added under the STA (2) as shown in [Figure 17](#).

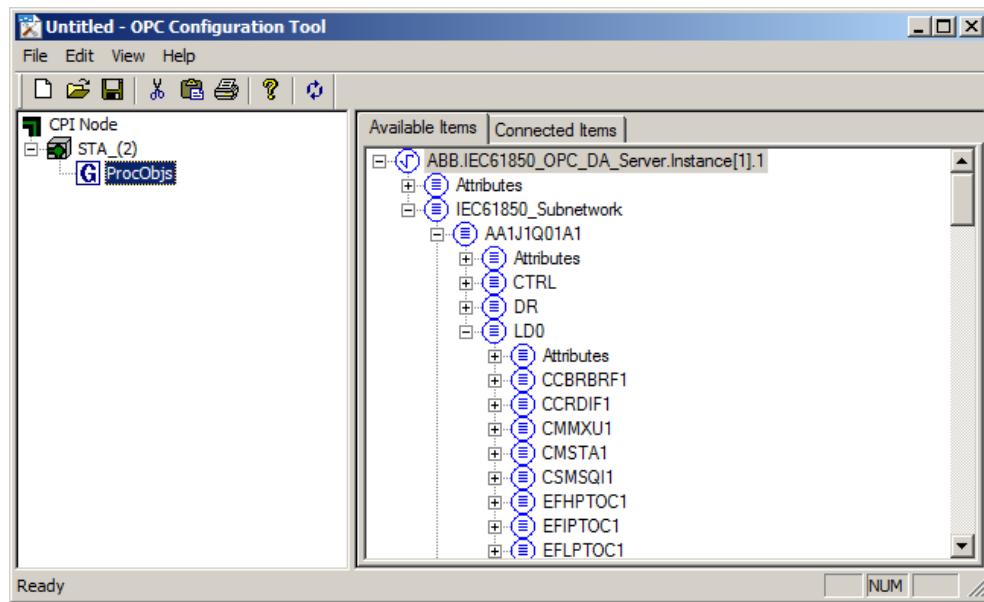


Figure 17: The main window after adding ProcObjs

6. To open the **Item Editor** dialog, activate the **ProcObjs** object on the left side of the main window and select **Edit/Add** from the main menu, see [Figure 18](#).

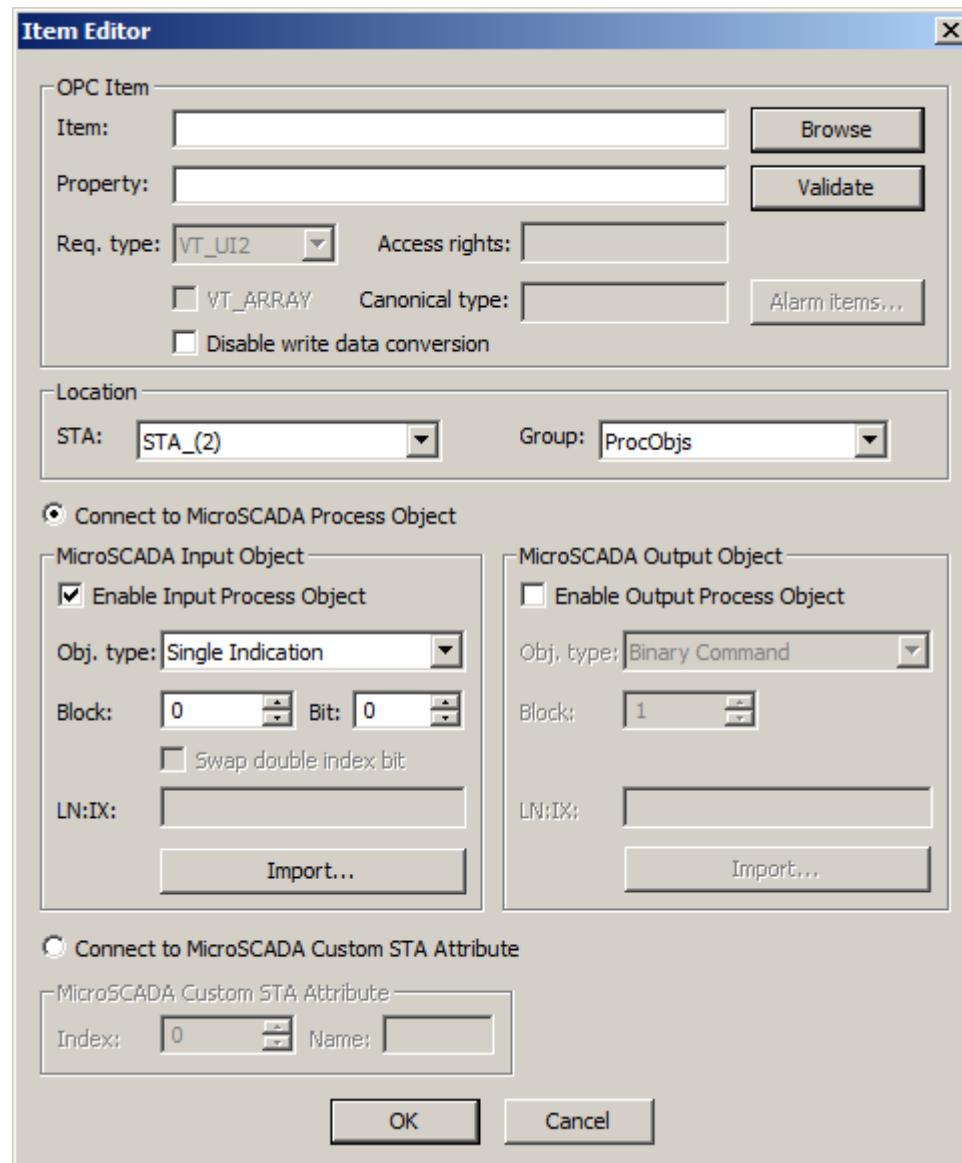


Figure 18: Item Editor dialog

7. Click **Browse**. Select a measurement in the **Available Items** dialog and click **OK**.

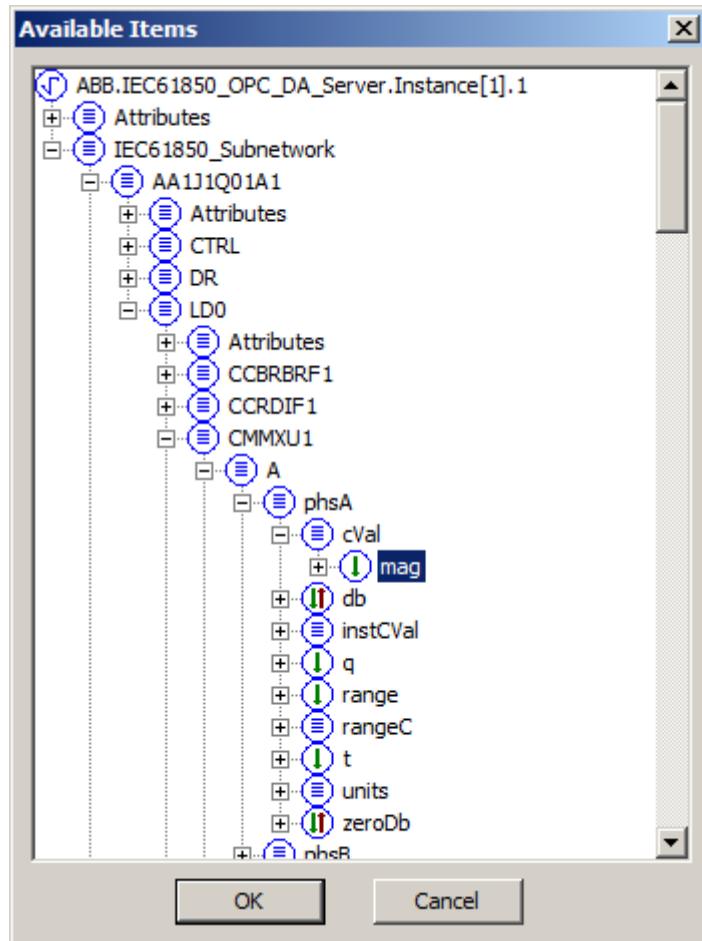
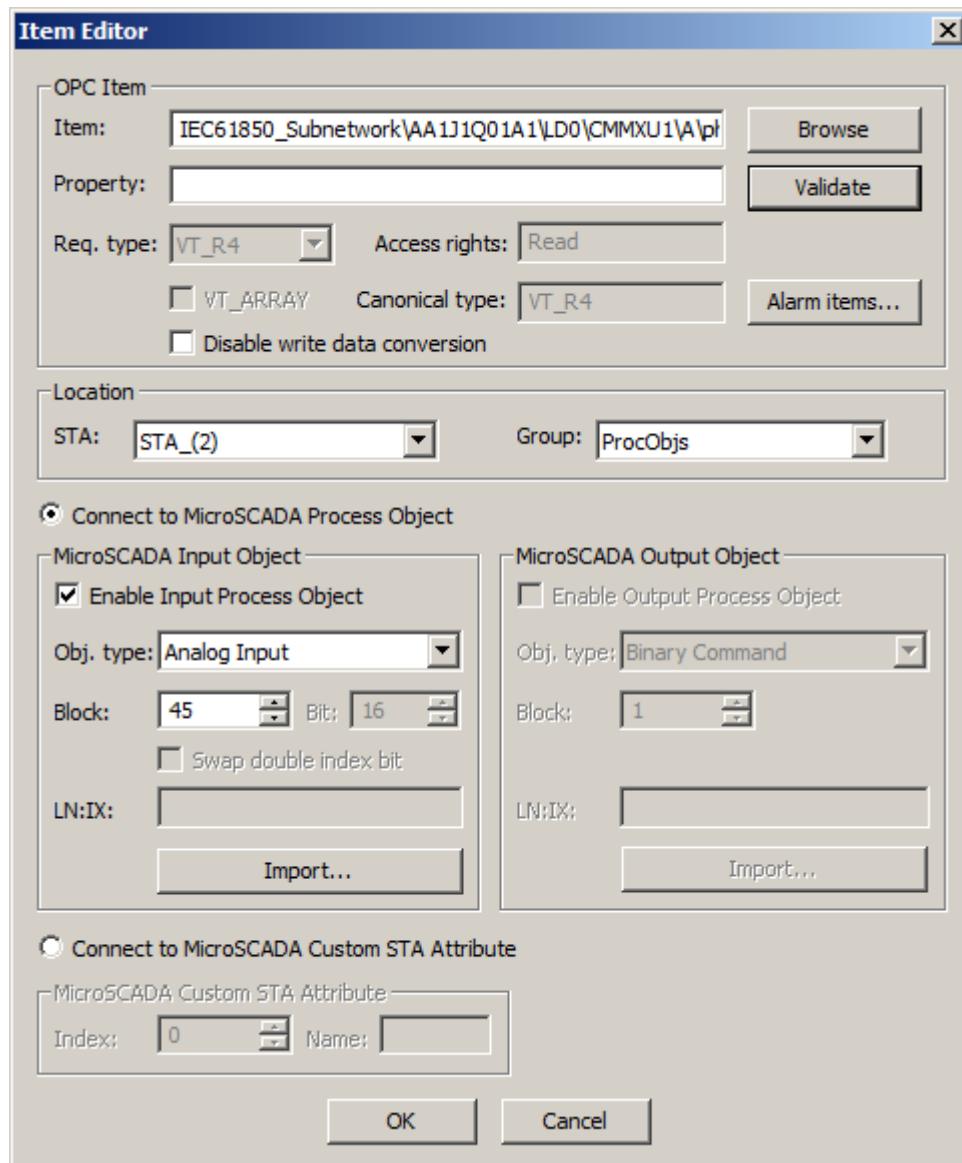


Figure 19: Available Items dialog

8. After clicking **OK**, the measurement is shown on the **Item** field in the **Item Editor** dialog, see [Figure 20](#).
9. Select the **Connect to MicroSCADA Process Object** radio button.
10. Check the **Enable Input Process Object** check box.
11. Select **Analog Input** in **Obj. type** drop-down list.
12. Fill in the other fields related to input process object.



*Figure 20: Filling in the Item Editor dialog fields*

13. Click the **Alarm Items** button to open the Alarm Items Editor dialog, see [Figure 21](#).

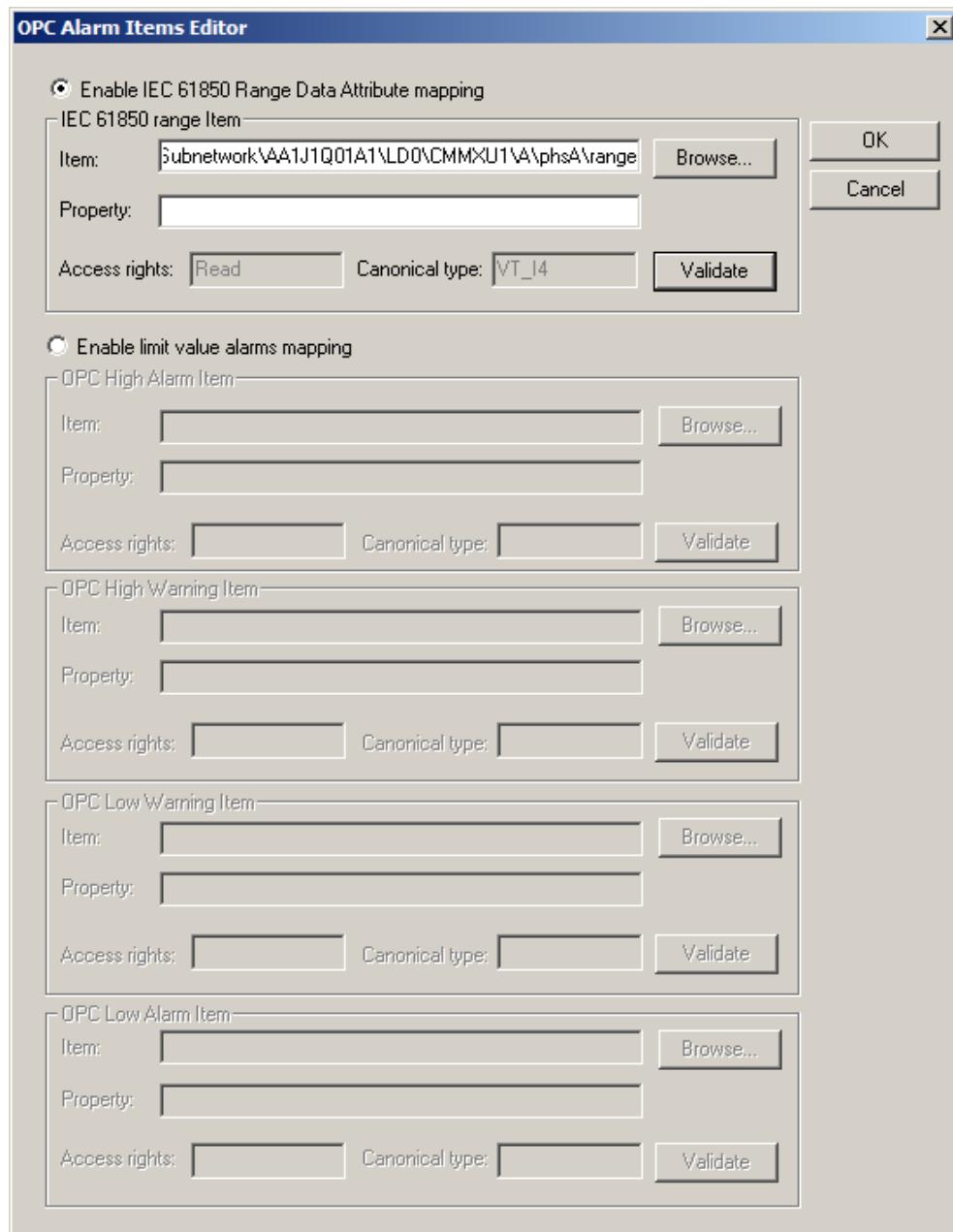
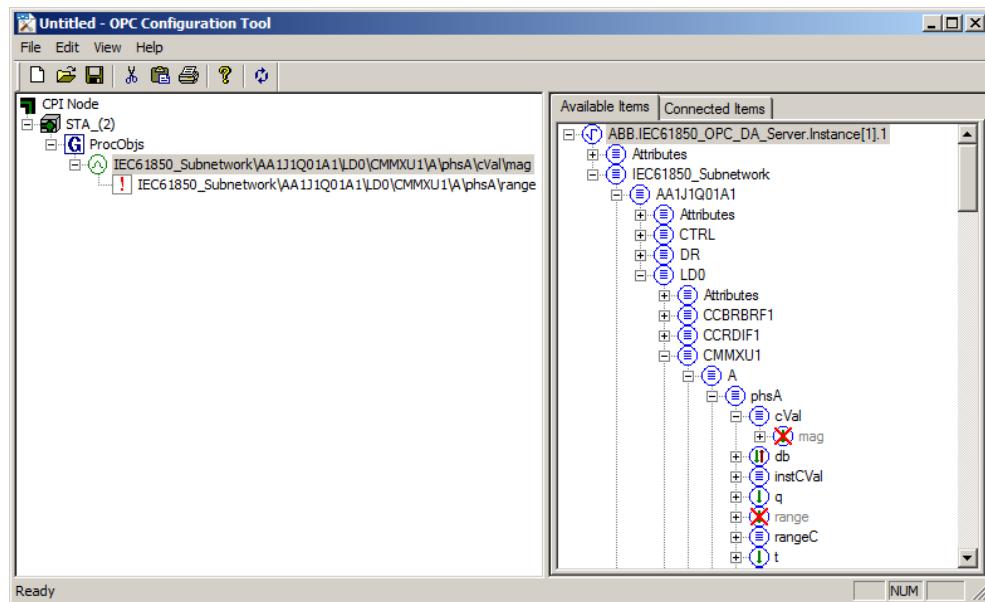


Figure 21: Filling in the Alarm Item Editor dialog fields

14. In **Alarm Item Editor** dialog select **Enable IEC 61850 Range Data Attribute mapping** and browse the appropriate measurement limit indication item in the **Browse** dialog.
15. Click **OK** in the **OPC Alarm Items Editor** dialog and then click **OK** again in the **Item Editor** dialog.



*Figure 22: Main window after adding items*

Mapped items can be seen on the left side of the main window, under ProcObjs object.

#### 4.1.5.4 Adding device connection status item for system self supervision

In case the System Messages Enabled (SE) definition made in the **CPI Node Properties** dialog is set to 4, the device connection status items must be added.

In case the System Messages Enabled (SE) definition made in the **CPI Node Properties** dialog is set to 1, adding the device connection status items is not required. When SE=1, the OPC Item Attributes\Device connection status has no special meaning and adding this item is not mandatory.

For more information about the differences between the SE=1 and SE=4 configurations, see [Section 4.2 Starting from the command line](#).



Adding of the device connection status item is necessary for the correct operation of the system self supervision (SSS).

If the status of the STA is not reported to be RUNNING in SE=4 configuration, the data is not updated in the process database.

The item Attributes\Device connection status is provided by the IEC 61850 OPC Server and it is used to report whether the OPC Server has a connection to the specified IED or not.

System events RUNNING and SUSPENDED are based on the updating of this item in the OPC Server. If SE=4, each configured STA object must have a device connection status item configured. Otherwise, the data points defined for the STA object are not updated to the process database. See SYS600 Application Objects manual for more information about the predefined event channel APL\_EVENT and events RUNNING and SUSPENDED.

To add the device connection status item when creating a new configuration:

1. When creating the configuration manually, add an STA for each IED to be configured. The number of STAs should equal the number of IEDs configured in the OPC Server. Each STA must have at least one configured group.
2. Add the IED's device connection status item in a group of the IED's STA with the lowest update rate.

This can be done manually before saving the configuration. The device connection status item for the IED in that STA is added automatically when saving the configuration. It is added to the group with the lowest update rate. In case there are items from multiple IEDs under one STA, the device connection status item is added to the IED with the maximum number of configured items in that STA.

A message box is shown for the user to accept the adding of the device connection status item.



*Figure 23: Confirming the adding of the device connection status item*

3. Confirm the adding of the device connection status item by clicking **Yes** for the indicated STA, or **Yes to All** for all STAs.  
If an existing configuration file with SE=4 is opened, the device connection status items are added if necessary when the file is saved.
4. Verify the attributes for the device connection status item.
  - Indication type is "I"
  - Object type is single binary "SB"
  - Address is allocated automatically. It is the maximum block address occupied by any item in an STA +1. The block number is present, but it is meaningless if the SE value is 4, since the process object is not updated.

#### **4.1.5.5 Importing process objects from an external file prepared with OPC PO List Tool**

Before using this feature, prepare the process objects list file which contains information about the process objects to be imported. This is a plain text file in which the process objects' attributes are stored in comma-separated format. There are 8 columns for different attributes:

*Table 3: PO list file format*

“STA_UN”	UN attribute
“STA_TN”	meaning of STA’UN’:BTN
“LN”	object name
“IX”	object index
“OX”	Signal text
“PT”	object type
“OA”	block address
“OB”	bit address
“OI”	Object Identifier
“CX”	Comment text

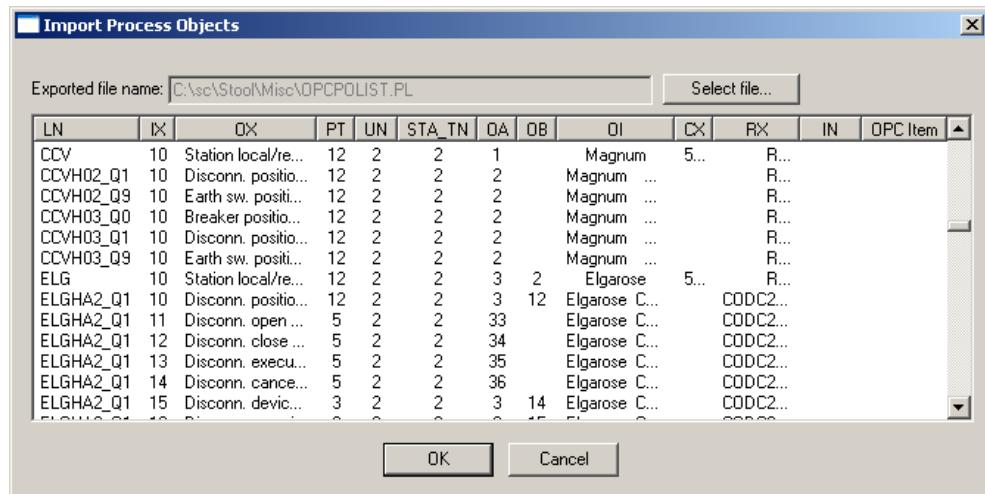
Table continues on next page

“RX”	Filter text
“IN”	OPC item name
“ATTR_NAME:VAL”	User defined attribute name and its value (optional)

The OPC PO List tool can be used to create such a file. See [Section 4.1.7](#) for details.

The **Import Process Objects** dialog of the OPC DA Client Configuration tool is used for mapping SYS600 objects to OPC data items.

When the OPC groups have been created, the SYS600 process objects or custom attributes can be mapped to OPC Items. Select the appropriate group on the left side of the configuration tool main window and select **File/Import Proc List** from the main menu to open the **Import** dialog. In this dialog, select the file which has been prepared previously to see the list of available process objects, see [Figure 24](#).



*Figure 24: List of available process objects*

If this dialog is started for the first time, select the file which has been prepared previously. Otherwise, process objects are read from the last used file. Another file can be opened by clicking the **Select File** button. When the process objects are loaded, select the process object(s) from the list to be mapped to the OPC Items. Multiple selection is available.

When certain objects have been selected and **OK** is clicked, the following [Figure 25](#) opens.

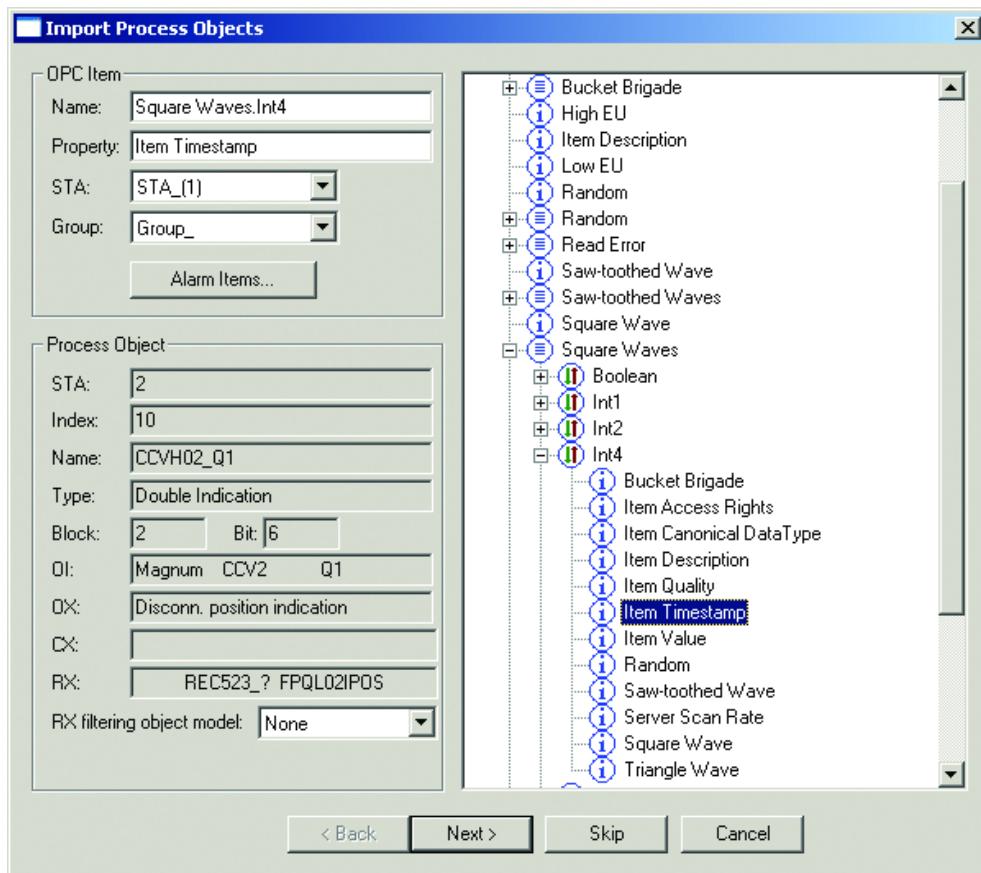


Figure 25: Import Process Objects dialog

The selected process objects are shown in the Process Object group. The address space of the OPC server is shown on the right side of the dialog (tree view). It is possible to filter the OPC items displayed in this tree view by using the RX attribute, if it is not empty.

There are two filtering modes that are allowed. The LIB 5xx mode is filtering by using the first 30 positions from the RX attribute and the IEC 61850 mode is filtering by using 31-63 positions of the RX attribute. When the filtering is enabled, the left tree contains only those items that satisfy the selected condition. The RX string can consist of characters and wildcards "\*". For mapping the appropriate process object to an OPC Item or item property, double-click the item on the right side of the dialog. After that, the **Name** field is filled and the **Add** button is activated.

If the imported SYS600 process object is of Analog Input or Digital Input type, OPC Items can be assigned to the alarm flags of this object.

1. Click the **Alarm Items** button in the same manner as in the **Item Editor** dialog.
2. Add the mapped object by clicking the **Next** button, or return to the previously mapped object by clicking the **Back** button, or skip the current object by clicking the **Skip** button.
3. When the process objects are mapped or skipped, the next selected process object is shown in this dialog, and the mapping procedure can be continued.
  - The whole mapping operation can be stopped by clicking the **Cancel** button. If some objects have already been mapped, the following dialog is shown, see [Figure 26](#).



*Figure 26: Save or discard the mapping results dialog*

Click the **Yes** or **No** button, to save or discard the mapping results. To continue the mapping of the process objects, click the **Cancel** button on this dialog.

**Example:**

The following is an example of importing SYS600 process objects and mapping them to OPC Items of Matrikon OPC Simulation Server.

Before starting this example, the following requirements should be fulfilled:

- Matrikon.OPC.Simulation server should be installed. See the example in [Section 4.1.5](#).
- The configuration file (matrikon\_cfg.ini) should be prepared. See the example in [Section 4.1.5](#).
- The SYS600 base system should be configured as in the example in [Section 4.1.1](#).

When using the SYS600 Object Navigator, there are two process objects that should be created:

- SPA/Analog Input process object with the following attributes: PT=SPA/Analog Input, OA=801.
- SPA/Single Indication process objects with the following attributes: PT=SPA/Single Indicat., OA=802, OB=2.
- Export the created process objects from SYS600 to the PO list file (ObjPOList.pl) by using the OPC PO List tool. See the description of PO List tool in [Section 4.1.7](#).

When these procedures are done, the following actions can be taken:

1. Start the configuration tool (DAOCT.EXE).
2. Open the configuration file (matrikon\_cfg.ini) from the main menu **File/Open**, see [Figure 27](#).
3. Select the STA\_(2) object on the left side of the main window. Then select **Add group** from the main menu to open the **Add Group** dialog.
4. Set the group name to ProcObjs in the dialog and click **OK**.

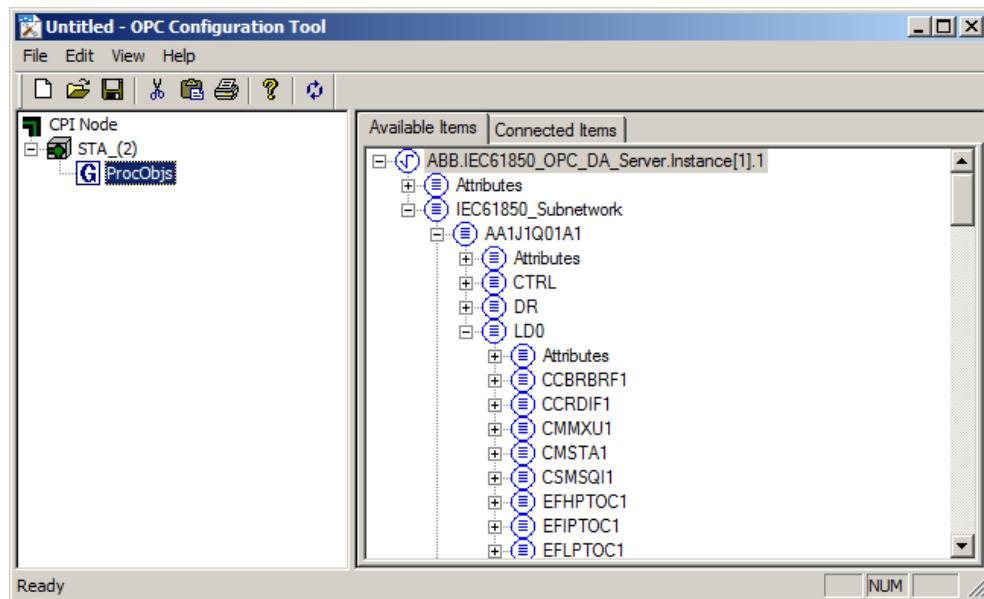
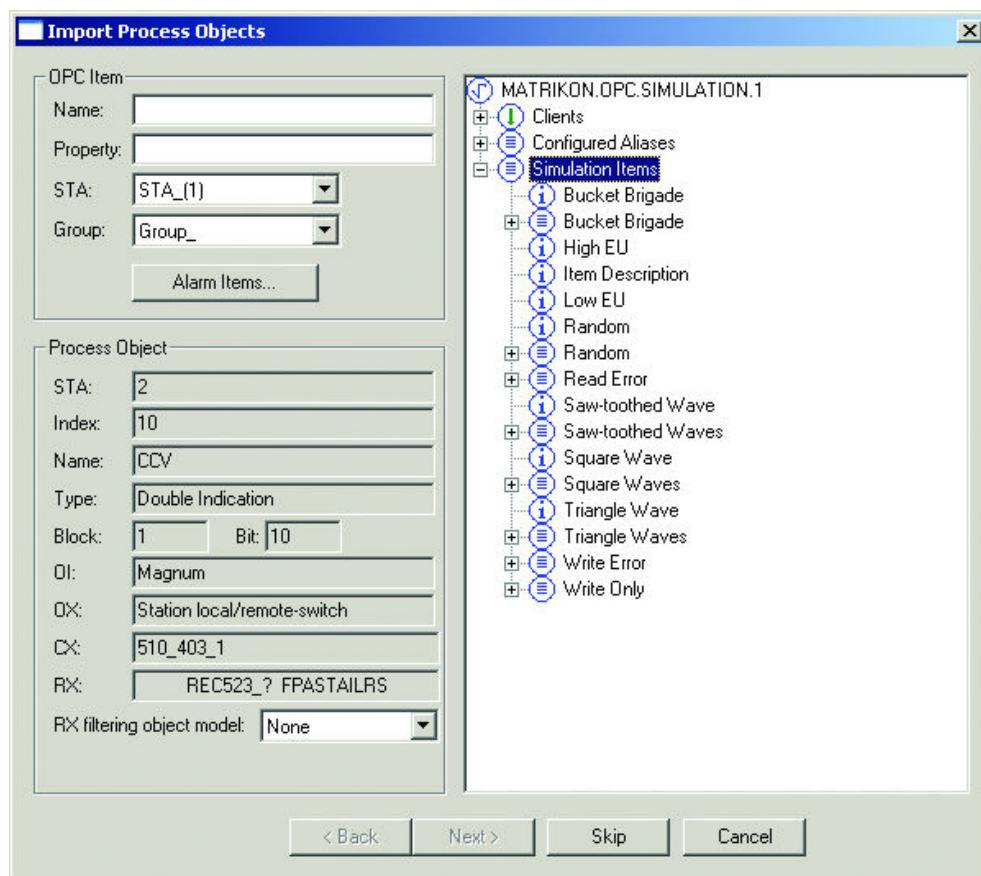


Figure 27: Main window with one empty group

5. Activate ProcObjs object on the left side of the main window and select **File/Import Proc List** from the main menu. Select the file ObjPOList.pl in the **Import** dialog to open a dialog containing a list of the available SYS600 objects, see [Figure 24](#). This dialog contains the two process objects (SPA/Analog Input and SPA/Single Indication) that have been created by using the Object Navigator.
6. Select the two objects from the list and click **OK**.  
The **Import Process Objects** dialog for the first object is opened, see [Figure 28](#).



*Figure 28: Import Process Objects dialog*

7. Expand the Simulation Items and Random tree node and double-click the OPC Item Random.Int2 on the right side of the dialog to fill in the **Name** field on the dialog and to activate the **Add** button.
8. Click **Next**.  
The first object (SPA/Analog Input) is mapped to the OPC Item Random.Int2. After that, the dialog is refreshed automatically and the new imported process object (SPA/Single Indication) is shown in the Process Object section. At the same time, Random.Int2 is marked as not allowed (already used) on the right side of the window.
9. Double-click the OPC Item Random.UInt2 on the right side of the dialog to show the Random.UInt2 item in the **Name** field and to activate the Next button. Click **Next**.
10. When the SPA/Single Indication object has been mapped to the OPC Item Random.UInt2, the importing is complete.  
On the left side of the main window, under ProcObjs group, there are now two mapped objects for the ProcObjs group: Random.Int2 and Random.UInt2.

#### 4.1.5.6 Automated External OPC Data Access Client configuration

This function is useful only in IEC 61850 based SYS600 systems. Similarly to importing process objects, see [Section 4.1.5.4](#), a process objects list file should first be prepared by using the OPC PO List tool. The auto configure function can be applied to process objects that contain a reference to an OPC item name in their IN attribute. This attribute is fulfilled by the SA-LIB objects when the IEC 61850 type process objects are created or imported to the SYS600 process database by using the SCL Importer tool.

When a connection to the IEC 61850 server is established, the automated configuration can be started by selecting **File/Auto Configure** from the main menu to open the **Auto Configure** dialog, see [Figure 29](#).

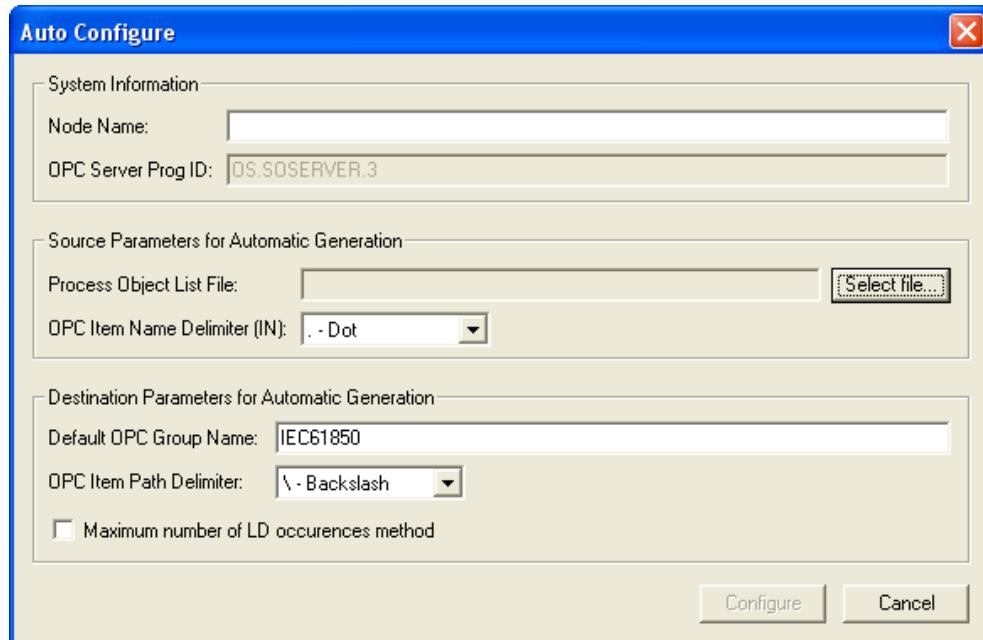


Figure 29: Auto Configure dialog

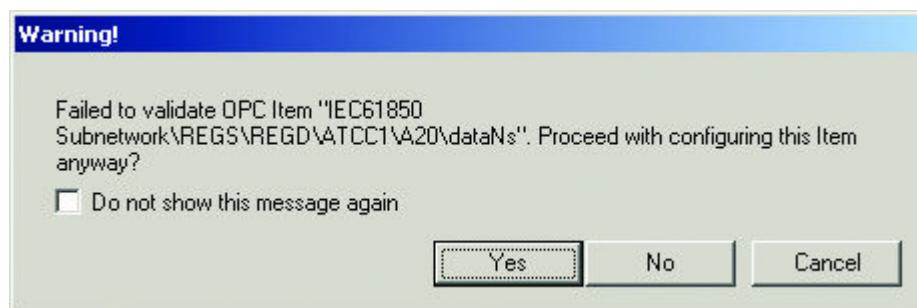
All the fields in this dialog are filled in with the default values, but it is also possible to redefine them manually.

<b>Node Name</b>	The computer identifier in the network (name or TCP/IP address).
<b>OPC Server Prog ID</b>	By default, this is the name of the OPC server instance.
<b>Process Object List File</b>	The name of the file containing a list of exported process objects. By default, this is the last used file name. This is a read-only field. To change this, click the <b>Select file...</b> button.
<b>OPC Item Name Delimiter (IN)</b>	The character used as a path delimiter in the IN attribute of process objects.
<b>Default OPC Group Name</b>	The name of OPC Group which contains the configured items. Such a group is created for all the Unit Numbers found in the Process Objects List file.
<b>OPC Item Path Delimiter</b>	The character used by an OPC server as a path delimiter.
<b>Maximum number of LD occurrences Method</b>	This flag indicates the maximum number of logical device occurrences under one STA.

During the auto configuration process, the tool reads process objects from the PO List file and maps an OPC Item to each of them. The OPC Item name consists of Node Name, OPC Server Prog ID and the IN attribute of the imported process object. The newly configured OPC Items are placed in the Default OPC Group Name, which is located under the STA number corresponding to the Unit Number of process object.

When all the fields of the **Auto Configure** dialog are completed and verified, click the **Configure** button to start the automatic configuration process. The status of the configuration process and its progress information is shown in the opening dialog.

A warning dialog may be displayed during the auto configuration process, see [Figure 30](#)



*Figure 30: Auto Configure Confirmation dialog*

Such a dialog with an appropriate message is displayed in the following cases:

- The OPC item name cannot be verified because the OPC server configuration does not contain such an item or there is no connection to the OPC server.
- The OPC item has already been mapped to another process object.
- The process object or SM STA attribute has already been mapped to another OPC item.

The operation is confirmed by clicking **Yes** and canceled by clicking **No**, and the auto configuration process is canceled by clicking **Cancel**. To apply the choice to all the similar situations, check the **Do not show this message again** check box.

After the auto configuration process is finished, all the configured items are shown in the tree structure on the left side of the main window.

#### 4.1.6 Importing and Exporting OPC Items configuration

OPC DA Client Configuration Tool can be used to import the whole item configuration from a comma-separated (CSV) file (main menu **File/Import CSV file**). The file can be prepared in any text editor or Excel, or created in the same or previous session of the configuration tool.

The file has the following format:

Each record contains 7 or 8 fields, separated by commas. The fields should not contain quotation marks. A record for an item mapped to a custom STA attribute has 8 fields:

1. Fully qualified OPC Item name.
2. The character **I**, **P** or **R** indicating whether the record describes **Item**, **Item Property** or **IEC 61850 Range Data Attribute**.
3. Required data type (for example **VT\_BSTR**).
4. Object type as it is seen in the item editor dialog (Single Indication, Analog Input, Custom STA attribute and so on).
5. "STA" followed by the STA number concatenated by an underscore ("\_"), as it is seen on the left side of the tool main window (for example **STA\_(2)**).
6. Custom STA attribute in the form "STA:Sxx", where xx is the name of the attribute.
7. Address of the process object or STA attribute index (0 if the attribute is not indexed).
8. Name of the group to which the OPC Item belongs.

It is also possible to import a .csv file which contain UTF-8 BOM (Byte Order Mark) header, but the actual contents of the file should contain only ASCII characters.

A record describing an item mapped to a process object has the same format as above, except field 6, which is not included. Therefore, the record has only 7 fields. The address field can be in the form **BLOCK:BIT** for Single or Double indication objects or **BLOCK** for other Process Objects. The following is an example of the CSV file rows:

```
Bucket Brigade.UInt2,I,VT_UI2,Custom STA
attribute,STA_(2),STA:SIM,0,Group_1
Bucket Brigade.Int4, I,VT_UI2, Single indication, STA_(2), 1700:0, Group_2
```

The file can be prepared by using the configuration tool from the open configuration (main menu **File/Export CSV file**). In both operations (Export or Import), the CSV file can be selected in the standard Windows file chooser dialog. Both operations are enabled only if an existing configuration is open or a new one is created in the tool session. Export file is saved as ASCII file, i.e. no UTF-8 header is included.

## 4.1.7 OPC Process Object List tool

The OPC Process Object List tool (OPC PO List tool) can be used to make it easier to create mapping between the OPC Items and SYS600 addresses (process objects). The tool produces a file (OPCPOLIST.PL) in which available process objects are listed in comma-separated (CSV) format. This file is then used by the OPC Configuration tool to select a process object and map it to an OPC Item.

### 4.1.7.1 Introduction

The SYS600 External OPC Data Access Client uses its own parameter file. Along with other data, this file contains mapping information on OPC Items and points in SYS600. These points are block and bit addresses of the SPA type process objects in one of the SYS600 applications. The OPC DA Client Configuration tool allows the manual customizing of these values for an OPC Item. Another possibility is to import the list of available addresses from an external file, then select a row in the imported table and assign an OPC Item to the block and bit addresses selected. The OPC PO List tool makes it easy to create such an external file. The format of the file is described in [Section 4.1.7.9](#).

The following actions can be done in the OPC PO List tool:

- Display the process objects filtered by the OX and RX attributes or by using the custom filter
- Browse process objects
- Set a new station number for the listed process objects
- Set filters
- Include and exclude an object into/from the output file
- Change the application from which the process objects are taken
- Save the Process Objects List
- Generate and assign new block and bit addresses for process objects that are included in an export file

### 4.1.7.2 Start-up

OPC PO List tool is a Visual SCIL tool of SYS type in SYS600. It can be started from the Tool Manager by double-clicking its icon:

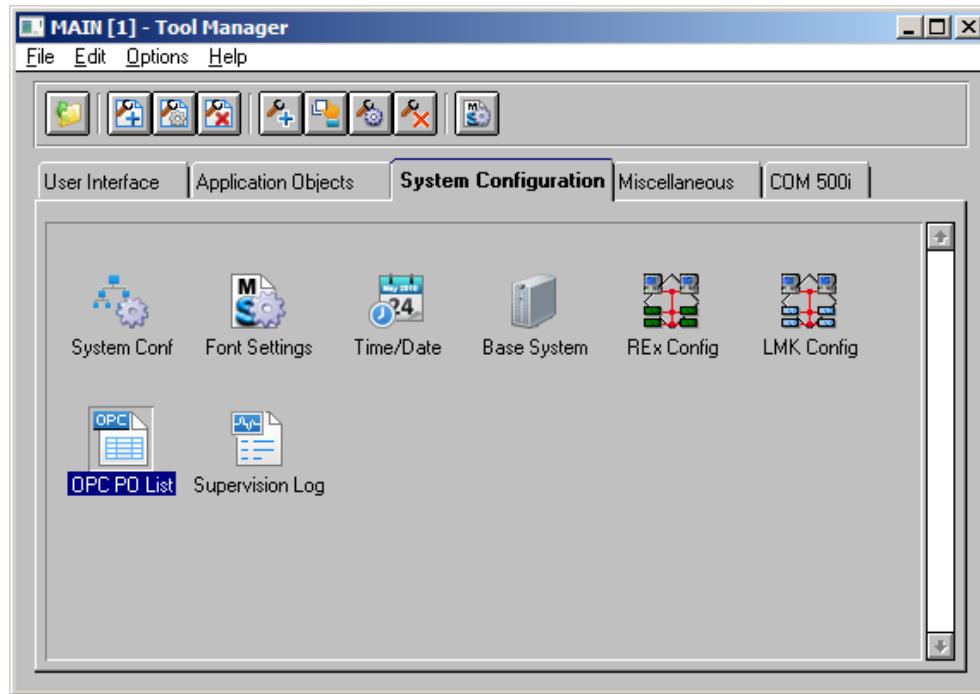


Figure 31: Starting the OPC PO List tool from the Tool Manager

#### 4.1.7.3

### Browsing Process Objects

The main dialog of the OPC PO List tool is a process objects navigator with a main table, a toolbar and a menu, see [Figure 32](#). At start-up, the tool reads the current application and shows the available SPA type process objects.

There are five standard columns that are always shown: Process object (in 'LN':P'IX' format), the Object Identifier of the process objects, signal text (OX), Block/Bit address and station (value of STA'UN':BTN). There is also one user-defined column (IX), where another process object attribute can be listed.

OPC Process Objects List Tool						
No Filter(s) in Use						User-defined attribute: None
Object	Object Identifier	Signal Text	Block/Bit addr.	Station	IN	
S2B2Q0-P10	STA2 STA2B2	Breaker position indication	1/2	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSW1.Pos.stVal	
S2B2Q0-P11	STA2 STA2B2	Breaker open select command	5	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSW1.Pos.ctsSelOff	
S2B2Q0-P12	STA2 STA2B2	Breaker close select command	6	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSW1.Pos.ctsSelOn	
S2B2Q0-P13	STA2 STA2B2	Breaker open execute command	7	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSW1.Pos.ctlOperOff	
S2B2Q0-P14	STA2 STA2B2	Breaker close execute command	8	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSW1.Pos.ctlOperOn	
S2B2Q0-P15	STA2 STA2B2	Breaker device control block	8	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSW1.Beh.stVal	
S2B2Q0-P16	STA2 STA2B2	Breaker open interlocked	0/16	41		
S2B2Q0-P17	STA2 STA2B2	Breaker close interlocked	0/16	41		
S2B2Q0-P18	STA2 STA2B2	Cause of interlocking	0	41		
S2B2Q0-P19	STA2 STA2B2	Breaker selection on monitor	0	41		
S2B2Q0-P20	STA2 STA2B2	Breaker command event	0/16	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSW1.Pos.Seld	
S2B2Q0-P25	STA2 STA2B2	Breaker cancel command	9	41	IEC61850 Subnetwork.REF542_41.LD1.Q0CSW1.Pos.ctlCan	
S2B2Q1-P10	STA2 STA2B2	Disconn. position indication	1/4	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSW2.Pos.stVal	
S2B2Q1-P11	STA2 STA2B2	Disconn. open select command	50	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSW2.Pos.ctsSelOff	
S2B2Q1-P12	STA2 STA2B2	Disconn. close select command	51	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSW2.Pos.ctsSelOn	
S2B2Q1-P13	STA2 STA2B2	Disconn. open execute command	52	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSW2.Pos.ctlOperOff	
S2B2Q1-P14	STA2 STA2B2	Disconn. close execute command	53	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSW2.Pos.ctlOperOn	
S2B2Q1-P15	STA2 STA2B2	Disconn. device control block	79	41	IEC61850 Subnetwork.REF542_41.LD1.Q1CSW2.Beh.stVal	

Figure 32: The main dialog of the PO List tool

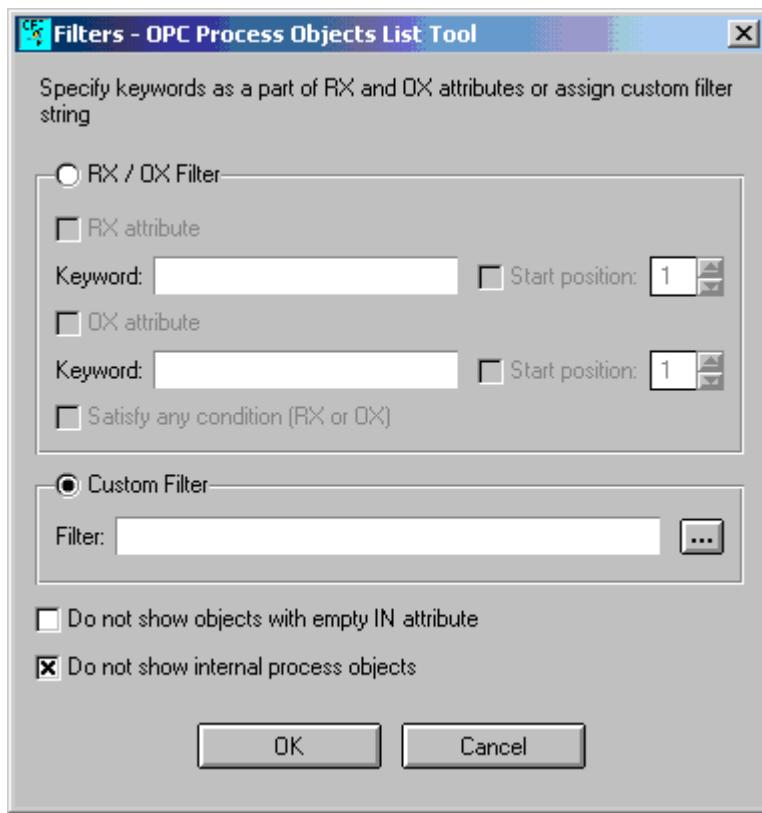
The main table has a limited row count. If the number of possible process objects is more than this value, then the full objects list is shown page by page. The full list can be navigated by

using the toolbar buttons **First**, **Previous**, **Next** and **Last**. The maximum row count is specified by the Table\_PageSize parameter of the [PO\_List] section in the OPCPOLIST.INI file in the user parameters directory. The default page size is 100. The left part of the status bar at the bottom of the table shows information about the number of the objects found and the currently displayed object numbers. The application from which the process objects are taken is shown in the right part of the status bar.

It is possible to refresh the object list by reading objects from the current application or another application by means of the **File/Open** menu command.

#### 4.1.7.4 Setting filters

The process objects listed in the main table can be filtered by using the RX and OX attributes. The **Filters** dialog is accessible from the toolbar button or from menu **Edit/Filter**, the [Figure 33](#) is shown when an option is selected.



*Figure 33: Setting filters*

The conditions for the OX and/or RX attributes can be specified in this dialog. It is possible to select objects which satisfy any or both of the specified conditions.

By default, all the objects shown in the main dialog are included in the output file. If the file OPCPOLIST.PL already exists in the tool home directory, only those objects that are present in the file, are marked as "included". The not included objects are dimmed.

By selecting the option "Do not show objects with empty IN attribute", the tool hides those objects that do not have a value in IN (OPC Item Name) attribute. This setting is useful in the IEC 61850 based systems, because it hides unnecessary objects during signal engineering.

All the filters are saved in the OPCPOLIST.INI file in the user parameter directory.

By checking the **Custom Filter** check box the user can add a custom filter string to set various conditions for the filtering process objects. This string uses the common syntax for a logical expression, for example UN>1 AND LN == "TEST".

To define a Custom filter condition, click the browse button (Button with three dots) in the **Filters** dialog > **Custom Filter** container.

The **Custom Filter** dialog opens.

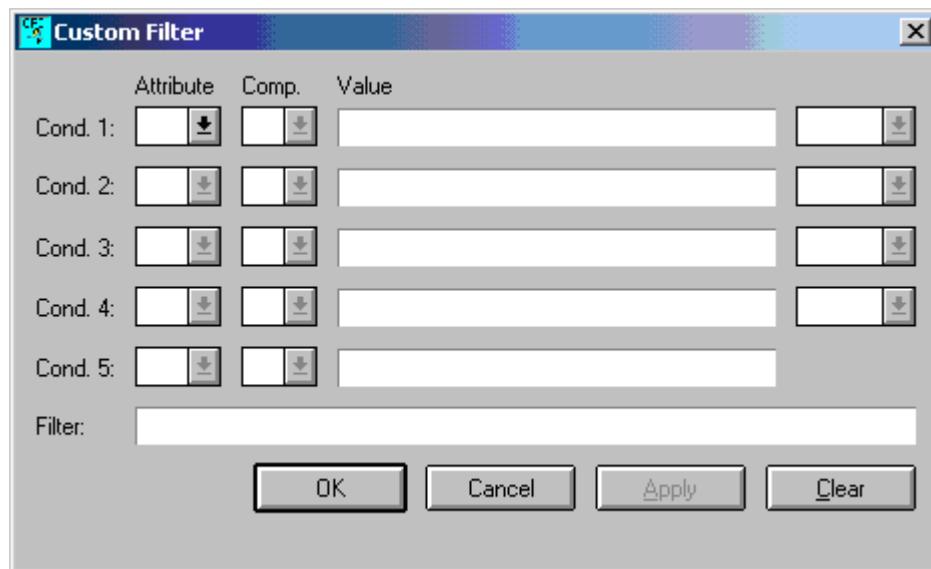


Figure 34: Setting Custom Filter

In this dialog it is possible to define filter conditions by using attributes found in the process database and by using a set of comparison operators and condition values. It is also possible to type in the filter condition directly in the **Filter** text box.

When the filter dialog is opened, only the first drop-down list is enabled. It is possible to select any attribute from the **Attribute** drop-down list. After the attribute is selected, the next drop-down list becomes enabled. From this list, it is possible to select the comparison signs: < (smaller than), <= (smaller than or equal to), = (equal to), >= (bigger than or equal to), > (bigger than) or <> (unequal). In the text box, it is possible to type any text.

To enter the next filter condition AND or OR has to be selected from the last drop-down list.

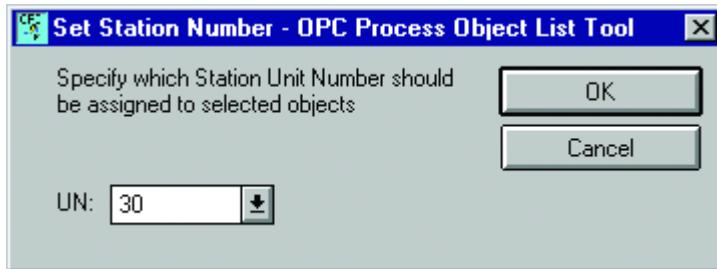
The chosen filter is added to the **Filter** text box when **OK** or **Apply** is clicked. Below are the explanations for all buttons:

- **OK** adds the contents of all fields and combo boxes into the **Filter** text box and closes the dialog.
- **Apply** adds the contents of all fields and combo boxes to the **Filter** text box, but does not close the dialog. Another filter condition can be selected in the combo boxes and fields to append to the existing filter conditions present in the **Filter** text box.
- **Clear** clears all fields and combo boxes.
- **Cancel** closes the dialog after confirming the update of the selected filter condition in combos.

When clicking **Cancel**, a message dialog opens to confirm whether the filter condition selected in combo boxes and fields must be updated. If **Yes** is clicked, the Filter Text is updated with the selected condition in combo boxes and fields. If **No** is clicked, the dialog closes without updating.

#### 4.1.7.5 Setting a new station number

The station number of the process objects listed in the Station column in the tool window can be modified. Select **Edit /Set station number** from the main menu, or click the right mouse button and select Set Station Number... from the pop-up menu to open the dialog shown in [Figure 35](#), in which a station number can be chosen.



*Figure 35: Set Station Number dialog*

The selected station number is assigned only for those objects marked to be included in the export file.

#### 4.1.7.6 Including / excluding Process Objects

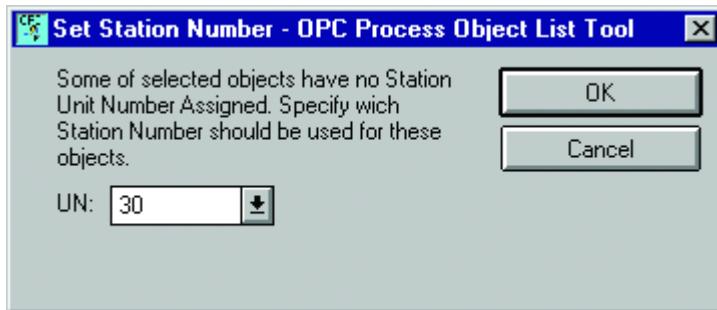
One or more process objects can be simultaneously included in or excluded from the output file.

To include process objects, select the table row(s) with the object(s) to include. Choose **Edit/Include Selected Objects** in the menu or click the toolbar button. Or click the right mouse button and select Exclude Selected Objects in the pop-up menu.

To exclude process objects, select the row(s) with the object(s) to exclude. Then select **Edit/Exclude Selected Objects** menu option, click the toolbar button, or select the corresponding item in the pop-up menu. All the excluded process objects are dimmed.

It is also possible to include or exclude all process objects simultaneously by selecting **Edit/Include All** and **Edit/Exclude All** in the menu or in the corresponding pop-up menu.

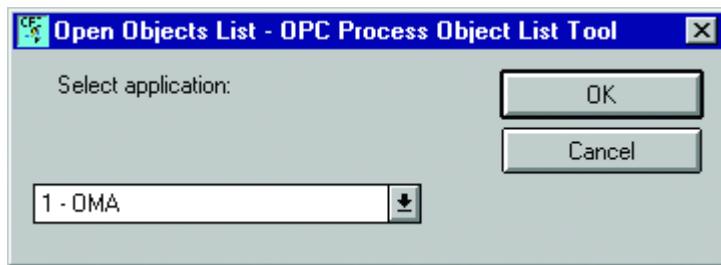
If some of the process objects included do not have a valid station number, then the **Set Station Number** dialog opens, see [Figure 36](#).



*Figure 36: Set Station Number dialog*

#### 4.1.7.7 Changing application

When the tool is started, it lists the objects for the current application. Later, the process object list can be retrieved from another application by using the menu item **File /Open** or the toolbar button to open the **Open Objects List** dialog, see [Figure 37](#).

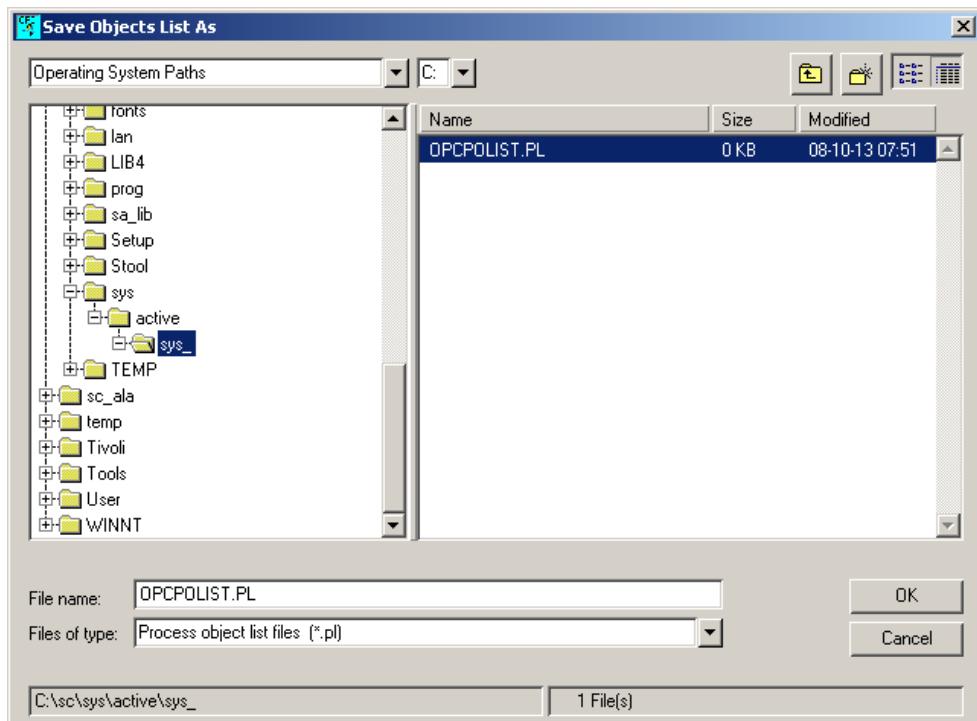


*Figure 37: Open Objects List dialog*

The drop-down list displays the currently selected application. Another application can be selected from the list.

#### 4.1.7.8 Saving the Process Objects list

The Process Objects list can be saved to the output file by selecting **File /Save** or by using the toolbar button. It is also possible to select **File /Save As** to open the **Save Objects List As** dialog. (See [Figure 38](#))



*Figure 38: Save Process Objects list as dialog*

The name and location of the output file is defined in this dialog.

When the output file has been defined, click **OK** or select **File /Save** to save the Process Objects list. The [Figure 39](#) opens.

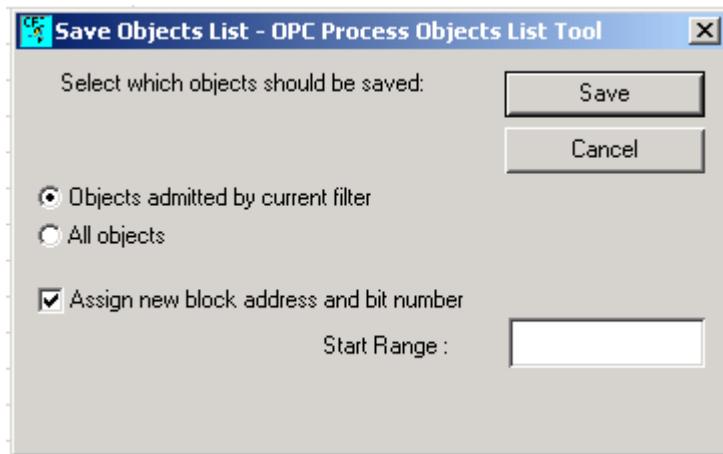


Figure 39: Save objects list dialog

The following options can be checked in the dialog:

- Save all objects, ignore the custom filtering
- Save only objects shown by using the current filters and marked as "included"

It is also possible to assign new block addresses and bit numbers for the process objects to be saved by checking the **Assign new block address and bit number** check box in the Save dialog. The tool generates them automatically.

Start range for the block can be assigned from 1 to 4094, depending on the number of the process object in the list. If the assign range goes beyond 4095, a notification message is given. The default start range is 1.

If this option is not selected, the tool validates the block and bit numbers of the process objects and the [Figure 40](#) is shown for each object with an invalid address:



Figure 40: Invalid address dialog

It is possible to exclude objects with invalid addresses from the export file by clicking the **Skip** button, or automatically generate the new address by clicking **OK**.

During the save operation the tool produces an output file named OPCPOLIST.PL to the folder sc\sys\active\sys\_. The format of the output file is described in [Section 4.1.7.9](#). File is saved as ASCII file, i.e. no UTF-8 header is included.

#### 4.1.7.9 Formats for the output file

The OPCPOLIST.PL file is a plain text file, in which process object attributes are stored in a comma separated format. It is also possible to import a .pl file which contain UTF-8 BOM (Byte Order Mark) header, but the actual contents of the file should contain only ASCII characters.

There are 10 columns for different attributes:

"STA_UN"	UN attribute
"STA_TN"	Meaning of STA'UN':BTN
"LN"	Object name
"IX"	Object index
"OX"	Signal text
"PT"	Object type <sup>1)</sup>
"OA"	Block address
"OB"	Bit address
"OI"	Object Identifier
"CX"	Comment text
"RX"	Filter text
"IN"	OPC item name
"ATTR_NAME:VAL"	User Defined attribute name and its value (optional)

- 1) This attribute does not depend on the SPA or other protocols. The SPA-type objects are filtered using the DX attribute.

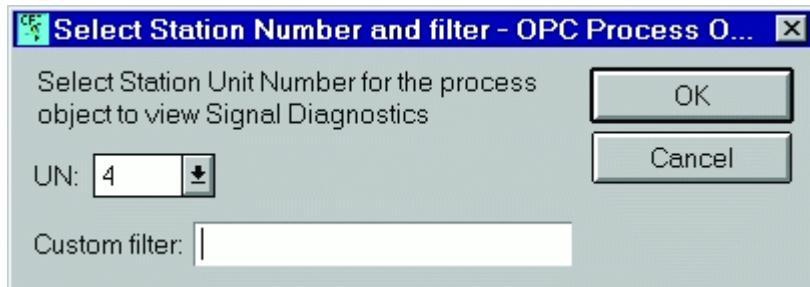
### Examples:

The following is an example of the file contents:

```
2,2,"RIVH03_R1Z",135,"Output TS2",3,23,1,
    "Rivers Winchester","",,"SPCJ4D61 FRFMFUIOUT","",,AT:"04-12-28 17:48:02"
2,2,"RIVH03_R1Z",137,"Output TS3",3,23,2,"Rivers
    Winchester","",,"SPCJ4D61 FRFMFUIOUT","",,AT:"04-12-28 17:48:02"
2,2,"RIVH03_R2A",2,"Tripping of stage Io>",3,23,3,"Rivers
    Winchester","",,"REJ511SB-AREJ511_1 FRFMFUITRP","",,AT:"04-12-28
17:48:02"
2,2,"RIVH03_R2A",4,"Tripping of stage Io>>",3,23,4,"Rivers
    Winchester","",,"REJ511SB-AREJ511_1 FRFMFUITRP","",,AT:"04-12-28
17:48:02"
2,2,"TEST111",11,"Breaker open select command",5,0,16,"Rivers
    IEC Q9","",,"FPQB01COBC CSWI1.Pos.ctlSelOff","",,AT:"04-12-28 17:48:02"
2,2,"TEST111",12,"Breaker close select command",5,1,16,"Rivers
    IEC Q9","",,"FPQB01COBC CSWI1.Pos.ctlSelOn","",,AT:"04-12-28 17:48:02"
2,2,"TEST111",13,"Breaker open execute command",5,2,16,"Rivers
    IEC Q9","",,"FPQB01COBC CSWI1.Pos.ctlOperOff","",,AT:"04-12-28
17:48:02"
2,2,"TEST111",14,"Breaker close execute command",5,5,16,"Rivers
    IEC Q9","",,"FPQB01COBC CSWI1.Pos.ctlOperOn","",,AT:"04-12-28 17:48:02"
```

### 4.1.7.10 Viewing OPC signals

If one or more OPC client instances are running, it is possible to see the process object states for the objects involved in running OPC client configurations. To do this, select **Tools/Signal Diagnostics** from the menu bar to open the **Select Station Number and filter** dialog, see [Figure 41](#).



*Figure 41: Select Station Number and filter dialog*

In this dialog, select the STA number of the process objects to be seen and set a custom filter to narrow down the list to the process objects which have this station number, and click **OK**. The Signal Diagnostics system tool is started and the process objects are loaded into this tool. The objects are loaded from the application that is open in the PO List tool. The main window of the Signal Diagnostics tool is shown below.

The dialog box has a title bar 'Signal Diagnostics'. It contains a large table with columns: LN:IX, OI, DX, PT, DV, RT.RM, OS, UN, DA, DB, SS. The table lists various process objects with their details. At the bottom are buttons for 'Set Filter...', 'User-defined:', and navigation icons (left, right, up, down). Below the table is a status bar with 'Signal 1/83 selected' and buttons for 'Enable history in dialog', 'History', 'Refresh', and 'Close'.

LN:IX	OI	DX	PT	DV	RT.RM	OS	UN	DA	DB	SS
S3B1Q0:P14	STA3 BAY1 REF43	Breaker cancel command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7230	16	2 - Automatic	
S3B1Q1:P10	STA3 BAY1 REF43	Disconn. position indication	12 - DB	2	2004-01-08 15:55:41.812	0 - Ok status: 4	31912	2	2 - Automatic	
S3B1Q1:P11	STA3 BAY1 REF43	Disconn. open select command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7237	16	2 - Automatic	
S3B1Q1:P12	STA3 BAY1 REF43	Disconn. close select command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7238	16	2 - Automatic	
S3B1Q1:P13	STA3 BAY1 REF43	Disconn. execute command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7239	16	2 - Automatic	
S3B1Q1:P14	STA3 BAY1 REF43	Disconn. cancel command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7240	16	2 - Automatic	
S3B1Q2:P10	STA3 BAY1 REF43	Disconn. position indication	12 - DB	2	2004-01-08 15:55:42.608	0 - Ok status: 4	31922	2	2 - Automatic	
S3B1Q2:P11	STA3 BAY1 REF43	Disconn. open select command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7247	16	2 - Automatic	
S3B1Q2:P12	STA3 BAY1 REF43	Disconn. close select command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7248	16	2 - Automatic	
S3B1Q2:P13	STA3 BAY1 REF43	Disconn. execute command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7249	16	2 - Automatic	
S3B1Q2:P14	STA3 BAY1 REF43	Disconn. cancel command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7250	16	2 - Automatic	
S3B1Q5:P10	STA3 BAY1 REF43	Earth sw. position indication	12 - DB	2	2004-01-08 15:55:43.436	0 - Ok status: 4	31842	2	2 - Automatic	
S3B1Q5:P11	STA3 BAY1 REF43	Earth sw. open select command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7267	16	2 - Automatic	
S3B1Q5:P12	STA3 BAY1 REF43	Earth sw. close select command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7268	16	2 - Automatic	
S3B1Q5:P13	STA3 BAY1 REF43	Earth sw. execute command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7269	16	2 - Automatic	
S3B1Q5:P14	STA3 BAY1 REF43	Earth sw. cancel command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7270	16	2 - Automatic	
S3B1Q9:P10	STA3 BAY1 REF43	Disconn. position indication	12 - DB	2	2004-01-08 15:55:44.217	0 - Ok status: 4	31932	2	2 - Automatic	
S3B1Q9:P11	STA3 BAY1 REF43	Disconn. open select command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7257	16	2 - Automatic	
S3B1Q9:P12	STA3 BAY1 REF43	Disconn. close select command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7258	16	2 - Automatic	
S3B1Q9:P13	STA3 BAY1 REF43	Disconn. execute command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7259	16	2 - Automatic	
S3B1Q9:P14	STA3 BAY1 REF43	Disconn. cancel command	5 - BO	0	2004-01-08 14:07:42.281	2 - Obsolete status: 4	7260	16	2 - Automatic	
S3B1RINGEVENT:P1	STA3 BAY2 REF43 (LON)	Ringevent	3 - BI	0	2004-01-08 15:55:52.607	0 - Ok status: 4	31792	1	2 - Automatic	
S3B2A0:P1	STA3 BAY2 REF43 (LON)	Analog Output	11 - AD	660.000000	2004-01-08 14:07:42.281	2 - Obsolete status: 4	19584	16	2 - Automatic	
S3B2M1:P10	STA3 BAY2 REF43 (LON)	Current L1	9 - AI	52.059155	2004-01-09 05:53:09.152	0 - Ok status: 4	27787	16	2 - Automatic	
S3B2M1:P11	STA3 BAY2 REF43 (LON)	Current L2	9 - AI	52.88239	2004-01-09 05:53:09.112	0 - Ok status: 4	27788	16	2 - Automatic	
S3B2M1:P12	STA3 BAY2 REF43 (LON)	Current L3	9 - AI	53.045849	2004-01-09 05:53:09.072	0 - Ok status: 4	27789	16	2 - Automatic	

*Figure 42: Signal Diagnostics tool dialog*

The process object values, states and time stamps are updated as soon as new events are received from the OPC server.

## 4.2 Starting External OPC Data Access Client

There are three methods of starting the External OPC Data Access Client: starting from the system command line, starting from a SCIL program and starting from the External OPC Data Access Client Control Panel.

### 4.2.1 Starting from the command line

The following commands can be used in the External OPC Data Access Client type:

```
DAOPCCL -show | -id "id_name" { -start "conf_file_name" [-trace [level] | -stop | -restart | -trace [on|off|high] }
    -show : show information about all active instances
    -id "id_name" : unique identifier of client instance (according to NT file name conventions)
```

```

-start "conf_file_name" : start instance with given config file
(according to NT file name conventions)
-stop [timeout] : stop instance. For forcing stop, timeout may be
specified (in sec.). If timeout is 0,infinite waiting
-restart : Restarts the running instance with id "id_name"
-trace [ normal | off | high | extreme] : switch tracing between
NORMAL , OFF (disabled level -default ), HIGH and EXTREME levels.
-se [1|4] :
-rm [0|1] :

```



Starting of the External OPC DA Client instance from the command line will make it run in the context of current user. When this user logs out, the execution of the instance will stop. See section 'Starting from a SCIL program' how to start the instance in the context of the other SYS600 processes.



The status handling of the IEC 61850 devices is divided into two different modes. The mode must be selected using a command line option "-se" that sets the node attribute value SE accordingly.

The setting of "-se" can be done using the **CPI Node properties** dialog of the OPC DA client Configuration Tool, or it can be read from the ini-file generated by the SCL Importer.

The modes are as follows:

With the setting "-se 1", the OPC item "Attributes\Device connection status" updates a process object that has been configured for the item. There is no special handling for this item. This mode of operation does not fully support the usage of System Self Supervision, since no RUNNING and SUSPENDED events are generated by the External OPC DA client. This is the default mode of operation. The functionality is the same with MicroSCADA versions prior to 9.3.

With the setting "-se 4", the OPC item "Attributes\Device connection status" does not update a process object configured for the item, but it updates special process objects used by System Self Supervision. Furthermore, if the "Device connection status" item indicates that the connection to the IED is lost or established, events RUNNING and SUSPENDED are generated to the predefined event channel APL\_EVENT in the application that is configured for the External DA client. The process objects used by SSS are of type ANSI Analog Input, UN=0, OA=1000+STA number and ANSI Binary Input, UN=0, OA=1000+1000000hex+STA number. No more than one Device connection status point should be configured for one STA object. If the STA object is reported to be SUSPENDED, no data configured for the STA object is sent to the MicroSCADA application. See SYS600 Application Objects manual for more information about the predefined event channel APL\_EVENT and events RUNNING and SUSPENDED.

The value given as a command line argument defines the value of the node attribute SE. This can be read from the application by evaluating 'NETx:SSE', where "x" is the node number of the External DA client instance. It is not recommended to modify the value of SE in runtime. If the "-se" parameter is not defined, the value of SE is 1. In case the trace option is used, the trace option must be defined before the "-se" option.



Other OPC servers than the IEC 61850 OPC Server do not usually provide the item Attributes\Device connection status. In these cases the setting "-se 1" must be used.

## 4.2.2 Starting from a SCIL program

The same commands can be executed by means of the OPS\_CALL function in SCIL, for instance:

```
@a = OPS_CALL("daopccl -id <id_name> -start <file_name> -trace <trace value> -se <se value>", 0)
```

or

```
@a = OPS_CALL("start daopccl -id <id_name> -start <file_name> -trace <trace value> -se <se value>", 0)
```



External OPC DA Client instance should be started from APL\_INIT\_1:C or using #EXEC in order to run in the context of permanent SYS600 processes. If External OPC DA Client instance is started using SCIL statement #DO e.g. from test dialog, the instance will run in the context of current user and its execution will stop when the user logs out.

Parameter -se is optional.

The SE (system message enabled) attribute can have the following values:

- 1 System message generation enabled (analog status points updated)
- 4 System message generation enabled (analog and binary status points updated)

If option "-se" is not defined and the configuration file is created with MicroSCADA version 9.2SP2 or older, value SE=1 is used as default and the functionality is as in version 9.2SP2 and older. See [Section 4.2.1](#) for more information.

This line can be included into a Visual SCIL tool method or command procedure. For example, include this command into the application initialization procedure (APL\_INIT\_... or LIB\_INIT\_...) to start the External OPC Data Access Client automatically at the SYS600 start-up.

The same command line argument presented in [Section 4.2.1](#), can be used when the External OPC client is started from the SCIL application. In case System Self Supervision is used, the option "-se 4" must be used.

To stop an External OPC Data Access Client, where "id\_name" is the id of the instance:

```
@b=OPS_CALL(" daopccl.exe -id <id_name> -stop")
```

or

```
@b=OPS_CALL("start daopccl.exe -id <id_name> -stop")
```

To restart an External OPC Data Access Client, where "id\_name" is the id of the instance:

```
@c=OPS_CALL("daopccl.exe -id <id_name> -restart")
```

or

```
@c=OPS_CALL("start daopccl.exe -id <id_name> -restart")
```

See the following examples of starting, stopping and restarting instances. In the examples the <id\_name> is IEC61850\_1 and the configuration file path is c:\sc\sys\active\sys\_\IEC61850\_1.ini

### Example 1

#### Start Instance:

```
daopccl -id IEC61850_1 -start c:\sc\sys\active\sys_\DAClients\\IEC61850_1.ini -trace high -se 1
```

The -trace switch is optional

#### Stop Instance:

```
daopccl -id IEC61850_1 -stop
```

**Restart Instance:**

```
daopccl -id IEC61850_1 -restart
```

**Example 2**

Using a SCIL command:

**Start Instance:**

```
@con=console_output(timems + " Start OPC DA Client for IEC61850")
@a=OPS_CALL("C:\sc\prog\OPC_Client\DA_Client\daopccl.exe -id IEC61850_1 -
start C:\sc\sys\active\sys_\DAClients\IEC61850_1.ini -trace normal", 0)
```

**Stop Instance:**

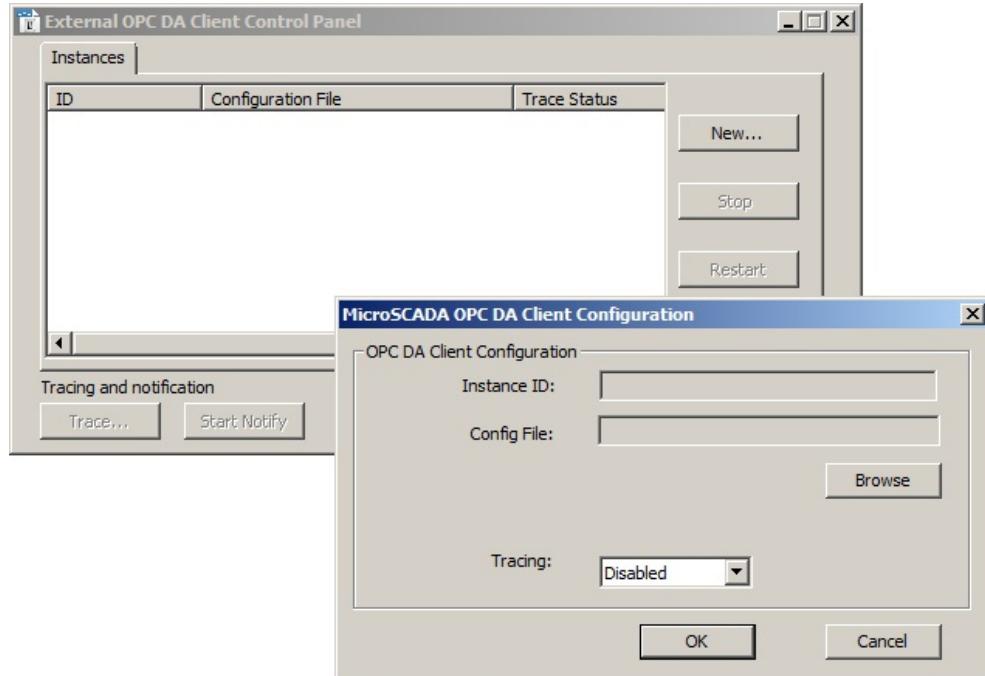
```
@a=OPS_CALL("C:\sc\prog\OPC_Client\DA_Client\daopccl.exe -id IEC61850_1 -
stop")
```

**Restart Instance:**

```
@a=OPS_CALL("C:\sc\prog\OPC_Client\DA_Client\daopccl.exe -id IEC61850_1 -
restart")
```

## 4.2.3 Starting from External OPC Data Access Client Control Panel

Run the DAOPCCP.EXE program to open the OPC DA Control Panel, see [Figure 43](#).



*Figure 43: External OPC Data Access Client Control Panel*

This dialog contains a list of the existing client instances. A new instance is started by clicking the **New Instance** button to open the dialog and fill in the following parameters:

Instance ID	The name of the created instance.
Config File	The name of configuration file, which can be selected by clicking the <b>Browse</b> button.
Tracing	<p>The level of tracing. A trace file with the same name as the Instance ID is written, if Normal, High level or Extreme is selected. The high level of tracing provides detailed information about data transfer during the run-time.</p> <p>If the Extreme level is selected, the OPC DA writes the following information into the trace log:</p> <ul style="list-style-type: none"><li>• Name of the item that received an update</li><li>• Its timestamp</li><li>• Value</li></ul> <p>The same information is written after an update has been transferred to the SYS600 application.</p>



In a normal situation, the Extreme level should not be set. It is intended for testing and debugging purposes only.

When a corresponding name (ID) is selected from the list of External OPC Data Access Client instances, the following actions are available for the existing instances:

- To change the trace level, click **Trace** and select a new level from the pop-up menu.
- To open the notification window, click **Start Notify**.
- To stop the client instance, click **Stop**.



If the External OPC DA client is started from the control panel, the value of SE is read from the ini-file. If the definition "SystemMessagesEnabled" is not present in the ini-file, the value of SE is 1.

- To restart the client instance, click **Restart**.



Use the SCIL command OPS\_CALL to start the External OPC Data Access Client permanently. See section 'Starting from a SCIL program' for details.

The External OPC Data Access Client Control Panel and the system command line are meant only for the temporary startup of client instances.

Permanent start up allows the instance to run even when the user logs off and the instance continues to run until the MicroSCADA service is running in the system.



When an External OPC DA Client instance is started with the **New** or **Restart** buttons in the Control Panel, all process objects are updated. If the HA (History Activation) attribute of the process objects is set to 2 - Update, useless events may be generated to the event list.

To avoid this, setting HA to 1 - New Value is necessary for all process objects which may generate events to the event list. All possible values for the process object attribute HA are listed in the Application Objects manual.

## 4.3 Supervising DA Client

The supervision of External OPC Data Access Client is done by reading the related Object Status (STA:SOS) attribute. In case a numerical value is received as a response to the STA:SOS request, the connection the OPC DA client is OK. The existence of the OPC DA client is also supervised by the node diagnostic of the base system and cyclically by the SSS (System Self Supervision).

When "System Messages Enabled" SE has the value 4, the STAn:SOS returns 0 when the connection to the IED is OK and the connection between the OPC DA client and the configured OPC server is OK. Otherwise, a value 13251 = SPAP\_DEVICE\_SUSPENDED is returned.

The returned value of the STAn:SOS is the same as the value which is updated to the ANSI Analog input process object described in [Section 4.2.1](#), setting "-se 4".

When an OPC DA Client instance is started, it updates and ANSI Analog Input process object UN=0, OA=6000+Node number, with value 10001 and an ANSI Binary Input process object, UN=0, OA=6000+1000000hex+Node number with the value 1. These process objects are used by the SSS. In general, the usage of SSS is recommended in the supervising of the External OPC DA Client.

When "System Messages Enabled" SE has the value 1 and the connection to the IED is lost, OPC DA Client loops through all configured items and marks the corresponding process objects to status OS=2 (Obsolete). In most cases, this happens when the OPC server has lost its connection to the IED or a failure has occurred in the OPC Server itself. SSS cannot supervise the behavior of the OPC DA client, if SE has the value 1.

## 4.4 OPC quality flags

OPC quality flags represent the quality state of an item's data value. The low 8 bits of the OPC quality flags are currently defined in the form of three bit fields:

- Quality
- Substatus
- Limit status

The 8 quality bits are arranged as follows:

QQSSSSLL

Details of the OPC standard quality bits are represented in the following tables:

*Table 4: Quality bit field*

QQ	BIT VALUE	DEFINE	SYS600 Object Status (OS)	DESCRIPTION
0	00SSSSLL	Bad	Depends on the Substatus	Value is not useful for reasons indicated by the Substatus.
1	01SSSSLL	Uncertain	Depends on the Substatus	The quality of the value is uncertain for reasons indicated by the Substatus.
2	10SSSSLL	N/A		Not used by OPC.
3	11SSSSLL	Good	0 (OK)	The Quality of the value is good.



If the quality is good for IEC 61850 OPC Server, the status returned to SYS600 depends on the time quality. This is represented by the second of the high 8 bits which are not used by other OPC servers. If this bit is set, the BAD\_TIME status is reported to SYS600. When the quality is not good, the time quality bit is ignored, since the OPC item value quality has higher priority than the quality of time.

The layout of the Substatus field depends on the value of the **Quality** field.

The used substatus values are listed below. A separate table is given for both configurations node attribute RM bit 0 = 0 and RM bit 0 = 1.

Table 5: Substatus for bad quality when node attribute RM bit 0 = 1

SSSS	BIT VALUE	DEFINE	SYS600 Process Object Status (OS)	DESCRIPTION
0	000000LL	Non-specific	1 (Faulty Value Status)	The value is bad but no specific reason is known.
1	000001LL	Configuration Error	1 (Faulty Value Status)	There is a server specific problem with the configuration. For example, the item in question has been deleted from the configuration.
2	000010LL	Not Connected	1 (Faulty Value Status)	The input is required to be logically connected to something but is not. This quality can reflect that no value is available at this time, for reasons like the value may not have been provided by the data source.
3	000011LL	Device Failure	1 (Faulty Value Status)	A device failure has been detected.
4	000100LL	Sensor Failure	1 (Faulty Value Status)	A sensor failure has been detected. The <b>Limits</b> field can provide additional diagnostic information in some situations.
5	000101LL	Last Known Value	1 (Faulty Value Status)	Communication has failed. The last known value is available.
6	000110LL	Comm Failure	1 (Faulty Value Status)	Communication has failed. No last known value is available.
7	000111LL	Out of Service	1 (Faulty Value Status)	The block is off-scan or otherwise locked. This quality is also used when the active state of the item or the group containing the item is inactive.
8-15		N/A	1 (Faulty Value Status)	Not used by OPC.

RM bit 0 = 0 provides a quality handling which is backward compatible with versions prior to SYS600 9.3.

Table 6: Substatus for bad quality when node attribute RM bit 0 = 0

SSSS	BIT VALUE	DEFINE	SYS600 Object Status (OS)	DESCRIPTION
0	000000LL	Non-specific	10 (Not Sampled Status)	The value is bad but no specific reason is known.
1	000001LL	Configuration Error	10 (Not Sampled Status)	There is a server specific problem with the configuration. For example, the item in question has been deleted from the configuration.
2	000010LL	Not Connected	10 (Not Sampled Status)	The input is required to be logically connected to something, but it is not. This quality can reflect that no value is available at this time, for a reason such as the value may have not been provided by the data source.
3	000011LL	Device Failure	10 (Not Sampled Status)	A device failure has been detected.
4	000100LL	Sensor Failure	10 (Not Sampled Status)	A sensor failure has been detected. The <b>Limits</b> field can provide additional diagnostic information in some situations.
5	000101LL	Last Known Value	1 (Faulty Value Status)	Communication has failed. The last known value is available.

Table continues on next page

SSSS	BIT VALUE	DEFINE	SYS600 Object Status (OS)	DESCRIPTION
6	000110LL	Comm Failure	10 (Not Sampled Status)	Communication has failed. The last known value is not available.
7	000111LL	Out of Service	10 (Not Sampled Status)	The block is off-scan or otherwise locked. This quality is also used when the active state of the item or the group containing the item is inactive.
8-15		N/A		Not used by OPC.

*Table 7: Substatus for Uncertain quality*

SSSS	BIT VALUE	DEFINE	SYS600 Object Status (OS)	DESCRIPTION
0	010000LL	Non-specific	2 (Obsolete Status)	There is no specific reason why the value is uncertain.
1	010001LL	Last Usable Value	2 (Obsolete Status)	This value has stopped writing. The returned value should be regarded as stale. Note that this differs from a Bad value with Substatus 5 (Last Known Value). That status is associated specifically with a detectable communication error on a fetched value. This error is associated with the failure of some external source to put something into the value within an acceptable time frame.
2-3		N/A		Not used by OPC.
4	010100LL	Sensor Not Accurate	2 (Obsolete Status)	Either the value has violated one of the sensor limits (in which case the limit field should be set to 1 or 2) or the sensor is otherwise known to be out of calibration via some form of internal diagnostics (in which case the limit field should be 0).
5	010101LL	Engineering Units Exceeded	2 (Obsolete Status)	The returned value is outside the limits defined for this parameter. Note that in this case (per the Fieldbus Specification) the <b>Limits</b> field indicates which limit has been exceeded but does not necessarily imply that the value cannot move farther out of range.
6	010110LL	Sub-Normal	2 (Obsolete Status)	The value is derived from multiple sources and has less than the required number of Good sources.
7-15		N/A		Not used by OPC.

*Table 8: Substatus for Good quality*

SSSS	BIT VALUE	DEFINE	DESCRIPTION
0	110000LL	Non-specific	The value is good. There are no special conditions.
1-5		N/A	Not used by OPC.
6	110110LL	Local Override	The value has been overridden. Typically this means that the input has been disconnected and a manually entered value has been forced.
7-15		N/A	Not used by OPC.

The Limit Field is valid regardless of the Quality and Substatus. In some cases, such as Sensor Failure, it can provide useful diagnostic information.

*Table 9: Limit bit field*

LL	BIT VALUE	DEFINE	DESCRIPTION
0	QQSSSS00	Not Limited	The value is free to move up or down.
1	QQSSSS01	Low Limited	The value has violated the lower limit.
2	QQSSSS10	High Limited	The value has violated the high limit.
3	QQSSSS11	Constant	The value is a constant and cannot move.

## 4.5 Quality handling with IEC 61850 OPC Server

When an External OPC DA client is used with the SYS600 IEC 61850 OPC server and the version is 9.3 FP2 or newer, the quality of the data item from the IEC 61850 device updates the corresponding process object attributes as follows:

*Table 10: IEC 61850 quality to Process Object Attributes*

IEC 61850 Quality bit	MicroSCADA process object attribute
detailQual.overflow	OF
detailQual.outOfRange	OR
source	SB
operatorBlocked	BL
test	TM

Values of the attributes are 0 when the corresponding quality bit is not set, and 1 when the corresponding quality bit is set.

In addition to the table above, the 13-bit IEC 61850 Quality is mapped as a whole to the MicroSCADA process object attribute QL. Attribute QL contains 16 bits and the IEC 61850 quality is shifted left by 3 bits (and padded with zeros) with most significant bit left. With versions 9.4FP2HF3 and newer, bit 4 (Inaccurate in 61850) of attribute QL is never set since it is used to indicate that the update is originated from general interrogation. The attributes OF, OR, SB, BL, TM and QL are updated in a same way with following input type process objects: single and double indications, analog value, digital value and pulse counter value. This functionality is present without any configuration and it cannot be disabled.

In case the OPC update is a result of general interrogation with IEC61850 OPC Server, the CT attribute of the process object is updated with cause of transmission value 20 (interrogated), otherwise it is updated with value 3 (spontaneous). This functionality is the same with all data types. If the update has been of type 'Buffered', the value of the CT attribute is added with 128 (bit 7).

## 4.6 Quality Handling, other OPC Servers

When an External OPC DA client is used with other OPC Servers but SYS600 IEC 61850 OPC Server, the quality handling is different, see table below. OS handling is similar in all OPC servers including SYS600 IEC 61850 OPC Server, but process object attributes SB (Substituted), OR (Out of Range) and BL (Blocked) are updated based on sub-statuses of the OPC Qualities:

OPC Quality VVVVVVVVQQSSSSLL									
VVVVVVVVendor specific byte									
QQ	Quality								
SSSS	Sub-status								
LL	Limit Status								
<b>V=vendor</b>									
Quality	Sub-Status	Value	OPC bits	OS	OS	SB	OR	BL	RM bit1=1
Good	Non-specific	0	1 1 0 0 0 0 L L	0	0				C0..C3 V
	N/A	1	1 1 0 0 0 1 L L	0	0				C4..C7 V
	N/A	2	1 1 0 0 1 0 L L	0	0				C8..CB V
	N/A	3	1 1 0 0 1 1 L L	0	0				CC..CF V
	N/A	4	1 1 0 1 0 0 L L	0	0				D0..D3 V
	N/A	5	1 1 0 1 0 1 L L	0	0				D4..D7 V
	Local Override	6	1 1 0 1 1 0 L L	0	0	1			D8..DB V
	N/A	7..15	1 1 7..15 L L	0	0				DC..FF V
Uncertain	Non-specific	0	0 1 0 0 0 0 L L	2	2				40..43 V
	Last usable	1	0 1 0 0 0 1 L L	2	2	1			44..47 V
	N/A	2	0 1 0 0 1 0 L L	2	2				48..4B V
	N/A	3	0 1 0 0 1 1 L L	2	2				4C..4F V
	Sensor Not accurate	4	0 1 0 1 0 0 L L	2	2				50..53 V
	Eng. Units Exceed	5	0 1 0 1 1 1 L L	2	2	1			54..57 V
	Sub-Normal	6	0 1 0 1 1 0 L L	2	2				58..5B V
	N/A	7..15	0 1 7..15 L L	2	2				5C..7F V
Bad	Non-specific	0	0 0 0 0 0 0 L L	10	1				0..3 V
	Configuration Error	1	0 0 0 0 0 1 L L	10	1				4..7 V
	Not Connected	2	0 0 0 0 1 0 L L	10	1				8..B V
	Device failure	3	0 0 0 0 1 1 L L	10	1				C..F V
	Sensor failure	4	0 0 0 1 0 0 L L	10	1				10..13 V
	Last known value	5	0 0 0 1 0 1 L L	1	1				14..17 V
	Comm failure	6	0 0 0 1 1 0 L L	10	1				18..1B V
	Out of Service	7	0 0 0 1 1 1 L L	10	1	1			1C..1F V
Limit Bits	N/A	8..15	0 0 8..15 L L	10	1				20..3F V
	Not Limited		0 Q Q S S S S 0 0						V
	Low Limit		0 Q Q S S S S 0 1						V
	High Limit		0 Q Q S S S S 1 0						V
Constant									
N/A									

Figure 44: Quality mapping, other OPC Servers but IEC 61850

This means that

- SB is set when OPC quality of the item is GOOD and its sub-status is 6, Local Override
- BL is set when OPC quality of the item is UNCERTAIN and its sub-status is 1, Last Usable or OPC quality of the item is BAD and its sub-status is 7, Out of Service
- OR is set when OPC quality of the item is UNCERTAIN and its sub-status is 5, Engineering Units Exceeded

The mentioned attributes SB, OR and BL are set to '0' when mentioned sub-statuses are not reported. The meaning of OPC sub-statuses are described in detail in chapter 'Quality handling with IEC 61850 OPC Server'.

RM (Running Mode) attribute, Bit 1, controls if QL (Qualifier) attribute of the process object is updated. If RM bit 1 is set, QL (Word) is updated followingly:

- Lower byte (bits 0..7) of QL is updated using standard OPC Quality, i.e. Quality, Sub-status and Limit fields
- Upper byte (bit 8..15) of QL is updated using the vendor specific byte in the OPC quality

RM attribute setting is done from command line or from the configuration file.

QL can be used e.g. make project specific quality mapping to NCC protocols. If RM bit 1 is 0, QL attribute of the process object is not updated.

## 4.7

## Buffered events

With some OPC servers, duplicate events may be received from the underlying process. E.g. with SYS600 IEC 61850 OPC Server duplicate events can be received from connected IED buffers when the OPC server is restarted or resynchronization of data with connected IEDs fails after a reconnection. Since these retransmitted events may be very old, filtering is needed to not display the duplicates in MicroSCADA Event List.

For event data sent via SYS600 internal protocol ACP communication to the process database there is a flag indicating that the event is buffered (BE). MicroSCADA base system will discard the events marked buffered (BE=1) which are older than the current valid value (with OS=0) in database. In order to identify and filter out duplicate events, External OPC Data Access Client sets the BE flag for OPC data change events from OPC server based on event timestamp. BE=1 is set if the event timestamp is older than a compare timestamp of External OPC Data Access Client.



BE is not set for data received via OPC refresh. E.g. when setting UP=1 for a STA unit in MicroSCADA the current data from the OPC server is refreshed and sent without BE=1 regardless of the timestamp of the data item.

For comparing event timestamps a STA specific compare timestamp is used. The compare timestamp is initialized to External OPC Data Access Client startup time and updated to the connected OPC server startup time if it is newer. If a 'Device connection status'-item (DCS) is configured for the STA the compare timestamp will be updated to the timestamp of DCS=TRUE change. All events older than the connection time will be marked with BE=1. If DCS is not configured or System Messages Enabled equals to 1, events older than External Data Access Client / OPC server startup time are marked with BE=1.

If External OPC DA Client is connected to IEC61850 OPC Server and SYS600 version is 9.4FP2HF3 or newer, bit 15 of the quality field of the OPC update indicates that the update is originated from the general interrogation of the IEC61850 report control block initialization. These updates are never marked as 'buffered' and they will update the process object as such.

If the loaded configuration file from .ini file contains setting "HistoricalEventsAsBuffered=No" in [OPC] section, the timestamp comparison mechanism above is not used and the duplicates are not filtered. With this configuration, the functionality is similar to version 9.3FP3 and older. This setting is worth to consider if the timestamps of the subscribed OPC items are permanently invalid and the timestamp based duplicate filtering cannot be used. If setting HistoricalEventsAsBuffered does not exist or it is set to "Yes", duplicate filtering as described above is used. Please note that updating the .ini file from configuration tools may overwrite this setting. In case of the IEC61850 OPC Server, setting 'Discard Old Buffered Events' in IEC61850 OPC Server configuration can also be used to eliminate duplicate events. See manual 'SYS600 IEC61850 Master Protocol (OPC)' for more details.

## 4.8 OPC error codes

*Table 11: Standard COM errors that are used in OPC servers*

Error	Description
E_FAIL (0x80004005)	Unspecified error.
E_INVALIDARG (0x80000003)	The value of one or more parameters is not valid. This is generally used in place of a more specific error where problems are unlikely or easy to identify (for example when there is only one parameter).
E_NOINTERFACE (0x80000004)	No such interface supported.
E_NOTIMPL (0x80000001)	Not implemented.
E_OUTOFMEMORY (0x80000002)	Not enough memory to complete the requested operation. This can happen any time the server needs to allocate memory to complete the requested operation.
CONNECT_E ADVISEDLIMIT (0x80040201)	Advise limit exceeded for this object.
OLE_E_NOCONNECTION (0x80040004)	Cannot Unadvise - there is no existing connection.
DV_E_FORMATETC (0x80040064)	Invalid or unregistered format specified in FORMATETC.

*Table 12: OPC-specific errors*

Error	Description
OPC_E_BADRIGHTS (0xC0040006)	The Items AccessRights do not allow the operation.
OPC_E_BADTYPE (0xC0040004)	The server cannot convert the data between the specified format/ requested data type and the canonical data type.
OPC_E_DUPLICATENAME (0xC004000C)	Duplicate name not allowed.
OPC_E_INVALIDCONFIGFILE (0xC0040010)	The server's configuration file is in an invalid format.
OPC_E_INVALIDFILTER (0xC0040009)	The filter string is not valid.
OPC_E_INVALIDHANDLE (0xC0040001)	The value of the handle is invalid. Note: a client should never pass an invalid handle to a server. If this error occurs, it is due to a programming error in the client or possibly in the server.
OPC_E_INVALIDITEMID (0xC0040008)	The item ID does not conform to the server's syntax.
OPC_E_INVALID_PID (0xC0040203)	The passed property ID is not valid for the item.
OPC_E_NOTFOUND (0xC0040011)	Requested Object (for example a public group) was not found.
OPC_E_PUBLIC (0xC0040005)	The requested operation cannot be done on a public group.
OPC_E_RANGE (0xC004000B)	The value is out of range.
Table continues on next page	

Error	Description
OPC_E_UNKNOWNITEMID (0xC0040007)	The item ID is not defined in the server's address space (on add or validate operations) or no longer exists in the server's address space (for read or write operations).
OPC_E_UNKNOWNPATH (0xC004000A)	The item's access path is not known to the server.
OPC_S_CLAMP (0x0004000E)	A value passed to WRITE was accepted but the output was clamped.
OPC_S_INUSE (0x0004000F)	The operation cannot be performed because the object is being referenced.
OPC_S_UNSUPPORTEDRATE (0x0004000D)	The server does not support the requested data rate but will use the closest available rate.

# Section 5      Technical description

## 5.1      OPC DA Client software components

OPC DA Client package includes three program modules:

1. DAOPCCL.EXE  
This is a SYS600 OPC DA Client, which provides access to various OPC- enabled devices from SYS600 applications. Another purpose is to give access to communication links for SYS600 based tools, when the connectivity to, for example LON and SPA devices, is granted by an OPC server, instead of the SYS600 communication services (like PC-NET). This is a console application with full integrity of the command line interface.
2. DAOPCCP.EXE  
This application provides a GUI interface for the SYS600 OPC DA Client. It allows starting, stopping and restarting of OPC DA Client Instances. The user can also start and stop the trace monitoring on the fly.
3. DAOCT.EXE  
This is the SYS600 OPC DA Client Configuration Tool. This tool allows creating and modifying parameter files that are used by the SYS600 OPC DA Client program. The tool allows loading OPC servers and browsing the server address space. Alternatively, it is possible to enter item names manually. This tool helps create and modify OPC groups, add OPC items to the groups and connect those items to SYS600 objects (process objects or custom STA:S attributes).

## 5.2      Visual SCIL tools

The visual SCIL part of the External OPC Data Access Client implementation includes one tool:

- OPC Process Object List Tool

The VSO file is OPCPOLIST.VSO, the initialization file is OPCPOLIST.INI. These two files can be placed in any path accessible for SYS600, for example in the SC\OPC\ directory.

OPC PO List tool is a SYS type tool. It can be started from the Tool Manager. To do this, its shortcut should be added to one of the pages in the Tool Manager main dialog. To make the last action easier, the tool should be registered in the \SC\STool\Misc\TOOLS.INI file:

```
[OPC_POLIST]
Tool_Dir_ = \SC\OPC
```

The \SC\OPC path is shown here only as an example. It should be the actual path to the VSO and INI files of the tool. If the tool is registered in the TOOLS.INI file, it is added to the available tools list in the **Add tool** dialog of the Tool Manager. See SYS600 System Configuration for details about working with the Tool Manager.



# Index

<b>A</b>	
Add STA.....	19
Alarm flags.....	25, 38
Alarm Items.....	28, 33, 38
Alarm Items Editor.....	33
Alarm limit types.....	25
All Events updated.....	62
Analog buffering.....	23
Analog Input.....	25, 32, 38
Auto Configure.....	41
Auto Configuration process.....	42
Available Items.....	31
Available Servers.....	15
<b>B</b>	
Back Application Number.....	14
Back IP Address.....	14
Back Node Number.....	14
Back Station Number.....	14
Base Application Number.....	14
Base IP Address.....	14
Base Node Number.....	14
Base Station Number.....	14
Base system configuration.....	11
Bit.....	27
Bit address.....	36, 51
BL.....	60
Block.....	27
Block address.....	36, 51
Blocked.....	60
Browsing process objects.....	45
Buffered.....	60
<b>C</b>	
Cause of transmission.....	60
Change application.....	48
Comment text.....	36, 51
Configuration tool.....	39
confirmation dialog.....	42
Connect to MicroSCADA Process Object.....	32
Connect to SYS600 Custom STA Attribute.....	27
Connect to SYS600 Process Object.....	27
CPI interface.....	13
CPI Node Properties.....	17
CpiTimeout.....	15
CT.....	60
Custom STA object attributes.....	24
CX.....	36, 51
<b>D</b>	
DAOCT.EXE.....	15, 29, 39
Data quality.....	57
Default OPC Group Name.....	42
Device connection status item.....	35
Device connection status item configuration.....	35
Digital Input.....	25, 38
<b>E</b>	
Enable Input Process Object.....	27, 32
Enable limit value alarms mapping.....	28
Enable Output Process Object.....	27
Exclude process objects.....	48
<b>F</b>	
Filter text.....	37, 51
<b>G</b>	
general interrogation.....	60
Group.....	25
<b>H</b>	
HistoricalEventsAsBuffered.....	15, 62
<b>I</b>	
Import.....	27
Importing process.....	36
Importing process objects.....	25
Import Process Objects.....	40
IN.....	37, 51
Include process objects.....	48
Index.....	27
Input process objects.....	23
Invalid address.....	50
Item.....	26
Item Editor.....	11, 25
IX.....	36, 51
<b>L</b>	
LN.....	36, 51
<b>M</b>	
Main window.....	30, 35
Mapping.....	25
Mapping of process objects.....	25
Matrikon OPC Simulation Server.....	15, 39
MS Max Reconnections.....	14
<b>N</b>	
Name.....	27
Node Name.....	42
<b>O</b>	
OA.....	36, 51
OB.....	36, 51
Obj. type.....	32
Object index.....	36, 51
Object name.....	36, 51
Object type.....	36, 51
ObjPOLIST.pl.....	39
OI.....	36
OPC DA Client Configuration Tool.....	12
OPC DA Client Control Panel.....	52
OPC DA Client package.....	65
OPC DA Control Panel.....	55
OPC data items with read-write access rights.....	24
OPC- enabled devices.....	65
OPC group.....	25
OPC Groups.....	25
OPC Item access rights.....	23
OPC Item icons.....	19
OPC item name.....	37, 51
OPC Item Name Delimiter (IN).....	42
OPC Item Path Delimiter.....	42
OPC items.....	23
OPCPOLIST.PL.....	44, 50
OPC Process Object list tool.....	44
OPC quality flags.....	57
OPC Server Prog ID.....	42
OPC Servers.....	13
OPS_CALL function.....	53

OR.....	60
OS.....	58
Out of Range.....	60
Output process objects.....	24
Own Node Number.....	14
Own Station Number.....	14
OX.....	36, 46, 51
<b>P</b>	
Percent deadband.....	25
PO.....	36, 39
Primary SYS600 base system.....	14
Process Object List File.....	42
Process objects list file.....	36
Process object status.....	58
Property.....	26
PT.....	36, 51
Pulsecounter buffering.....	23
<b>Q</b>	
QL.....	13, 61
Qualifier.....	13, 61
Quality.....	57
QueueUpdatesOfAllDataTypes.....	23
<b>R</b>	
Req. Type.....	26
RM.....	61
Running Mode.....	14, 61
RX.....	37, 46, 51
<b>S</b>	
save process objects.....	49
Save Process Objects list.....	49
Saving.....	49
SB.....	60
Secondary SYS600 Pro base system.....	14
Server properties.....	16
Server Properties dialog.....	15
Set filters.....	46
Set station number.....	48
Signal text.....	36, 51
Simulation server.....	15
SPA stations.....	11, 14
SPA type.....	45
STA_TN.....	36, 51
STA_UN.....	36, 51
Starting OPC DA Client.....	52
Sub-status.....	60
Substituted.....	60
SYS_BASCON.COM.....	11
System Messages Enabled.....	62
System Messages Enabled (SE).....	14
<b>T</b>	
Time bias.....	25
Timeout CPI.....	15
<b>U</b>	
UN attribute.....	36, 51
Update Rate.....	25



---

**Hitachi ABB Power Grids**  
**Grid Automation Products**  
PL 688  
65101 Vaasa, Finland



Scan this QR code to visit our website

<https://hitachiabb-powergrids.com/microscadax>