Operating Instructions Manual

OPC-Server

Language: English

www.hilscher.com

Index	Date	Version (SyCon)	Chapter	Revision
1	28.05.98	V 2.100	all	New
2	01.07.98	V 2.101	all	Rewritten
3	11.01.99	V 2.120	Errors Extension	Removed Insert
4	29.03.99	V 2.121	For PROFIBUS_DP, Interbus, CANopen	Rewritten
5	07.05.99	V 2.123	New structure of chapters	OPC server now available for PROFIBUS-DP Master, PROFIBUS-FMS, Interbus Master and CANopen Master
6	22.03.00	V 2.620	All	Rewritten and extended
7	03.04.08	V2.703	All Installation	Windows Vista Support PROFIBUS-FMS, CANopen SDO Up-/Download and PROFIBUS-DPV1 removed Chapter Installation updated

Although this program has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this program for any purpose not confirmed by us in writing.

Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this program or its documentation shall be limited to cases of intent.

We reserve the right to modify our products and their specifications at any time in as far as this contributes to technical progress. The version of the manual supplied with the program applies.

1 Introduction

1.1 General

The unification of data exchanges between process hardware and Windows applications has led to the development of OPC (**O**LE for **P**rocess **C**ontrol). The aim of this technology is to achieve a manufacturer-independent interface for access to the process data.This is comparable to the use of a printer under Windows. Here the printer manufacturer must make available for his printer, a driver with which all Windows programs can then print.

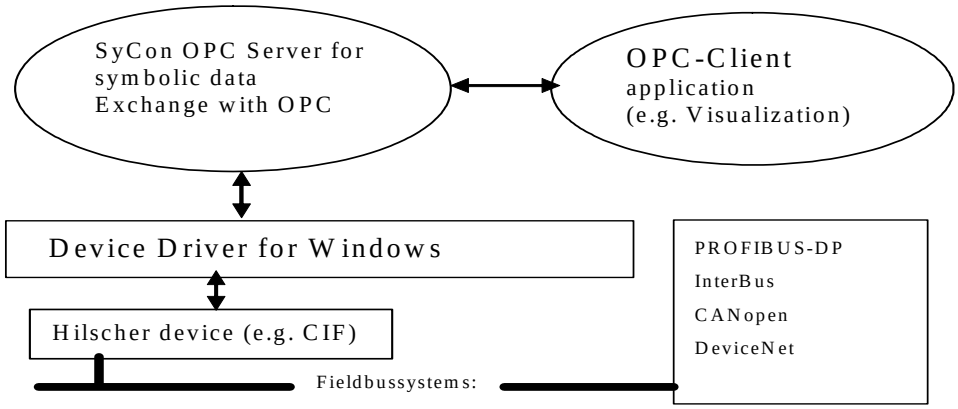
OPC works on the Client/Server principle, whereby both clients and server can be combined from different manufacturers. The data exchange is carried out by means of symbolic definitions, whereby again, the definition of the symbols is carried out in the server. The server is also responsible for the preparation of the data.

An OPC client can access the defined symbols (read and write) without having to worry about the depiction of the physical data. Furthermore, there is the ability to access an OPC server either locally or remote. "Local" access means that OPC server and OPC client are positioned on the same PC. In "Remote" access, the OPC server and OPC client are on different PCs. The transfer of the access to the server data then occurs via a data linkage (e.g. networks such as Ethernet, Internet or similar) between these PCs.

OPC is based on the OLE COM/DCOM model developed by Microsoft for the Windows operating system. A more detailed description can be found in the "OPC Overview" chapter.

Our OPC server is implemented as an additional function in our System Configurator. This results in the following advantages:

- The System Configurator serves at the same time for diagnostic and for start-up for the Field Bus system as also for the OPC functions.
- A configuration tool for the fieldbus system as well as for the OPC server**.**



1.2 Hints

The following two cases has to be distinuished: The User and the Programmer/Developer.

The User uses an OPC Client and an OPC Server in the following way: He configures both and this leads to the data exchange.

The programmer uses an OPC Client and an OPC Server in the following way: He configures the OPC Server and from the OPC Client he writes a program that uses the OPC Automation Interface to do the data exchage with the OPC Server.

Important Note: The OPC server supports only cyclic communication: PROFIBUS-DPV0, InterBus, CANopen PDOs respectively DeviceNet Implicit I/O. The OPC server does **not** support acyclic communication like PROFIBUS-FMS, PROFIBUS DPV1, InterBus PCP, CANopen SDO up-/download respectively DeviceNet explicit messaging.

2 Installation

2.1 System Requirements

- PC with 486-, Pentium processor or higher
- Windows NT4.0/2000/XP/Vista
- Free disk space: 30 - 80 MByte
- CD ROM drive
- RAM: min. 16 MByte
- Graphic resolution: min. 800 x 600 pixel
- Windows NT: Service Pack 6 or higher
- COM/DCOM only for OPC Server
- Keyboard and Mouse

NOTE: Depending on the used operating system, further requirements are necessary for the utilization of the OPC server functionality. Please note the information provided in the "Installation instructions" chapter.

2.2 Installation Instructions

The installation is carried out mostly automatically by means of the setup program provided, whereby the Windows platform being used is also automatically recognized.

Ensure that all the system components for the operating system being used are installed before the installation from the CD is begun.

Please note that the components of the configurator cannot be carried out from the CD but must be installed on the hard disk.

If you have already installed a licensed version of SyCon select the same folder for this installation without uninstall the old version.

If you have already installed a not licensed version of SyCon please uninstall first this version.

Hint: When the OPC client after the installation of the OPC server does not show available OPC server, then install from the CD: OPC Core Components Version 2.00. These are in the Windows installer file 'OPC Core Components 2.00 Redistributable.msi'.

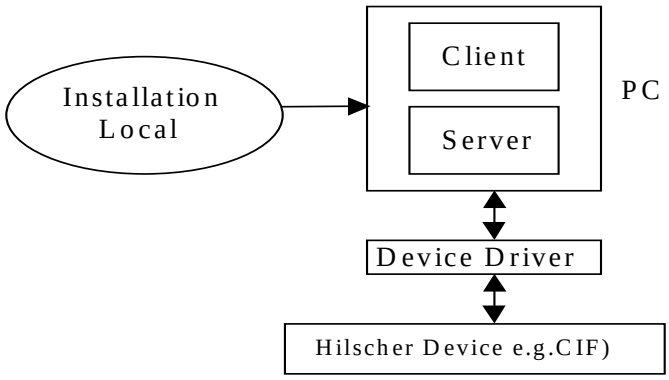
Windows NT 4.0:

For the operation of the OPC SERVER, the following system components must be installed before the installation of the configurator.

- Service Pack 6

Two different installation variants:

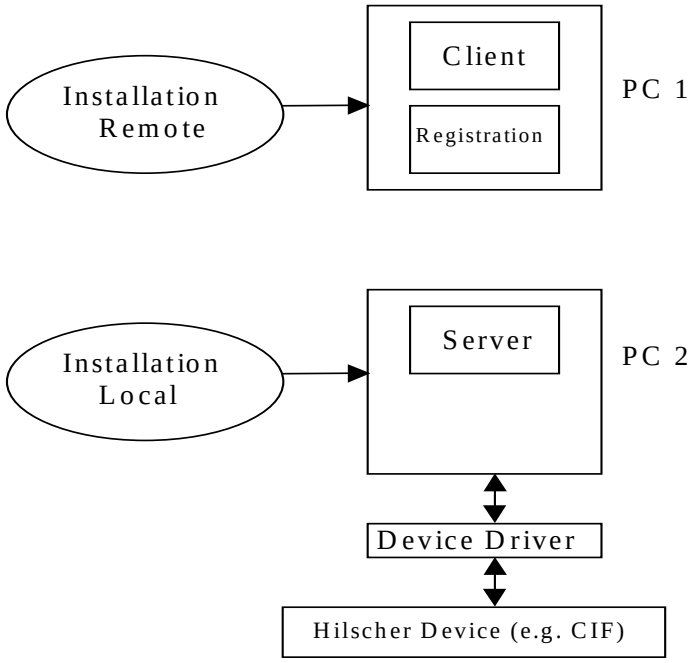
1. OPC server und OPC client are on **one PC**



Only the installation of the system configurator as a local OPC server is necassary.

See chapter ´Installation of the System Configurator as a local OPC SERVER:´

2. OPC server and OPC client are on **different PCs**



The installation of the system configurator as a local OPC server and for the access from a remote PC is necessary.

See chapter ´Installation of the System Configurator as a local OPC SERVER:´ and chapter ´Installation of the System Configurator for access to a remote PC:´

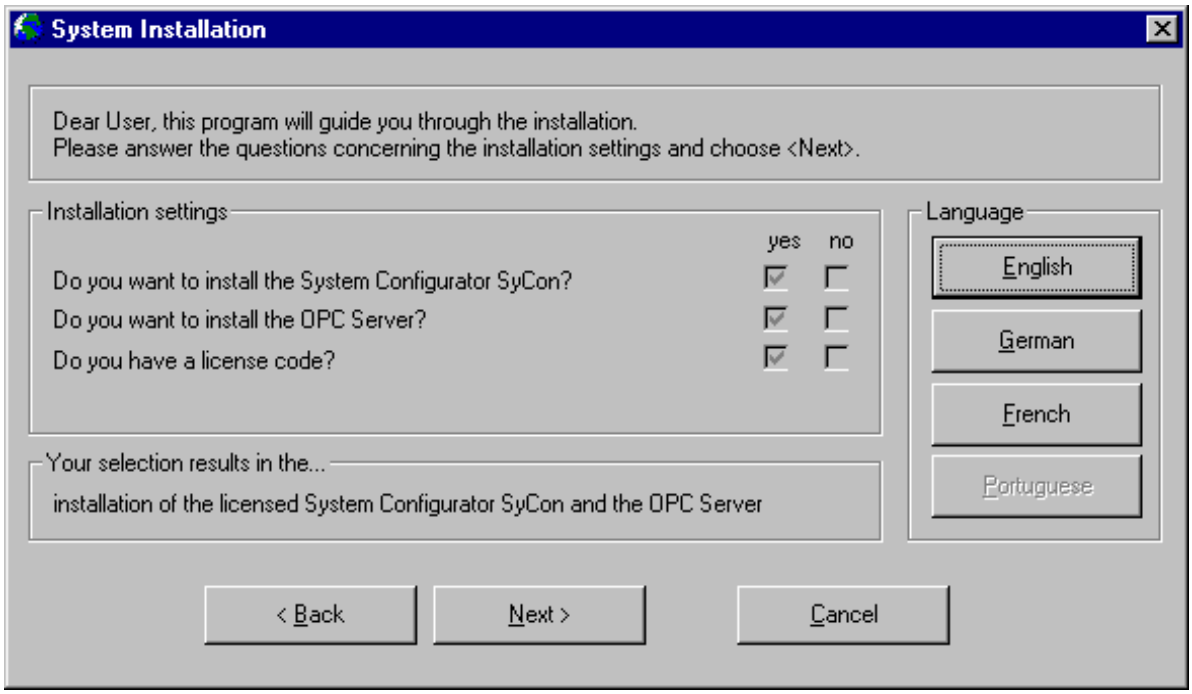
2.2.1 Installation of the System Configurator as a local OPC Server

With this kind of installation you install SyCon with all software components which are neccesary to configure a fieldbus and to serve the I/O data via the OPC technology.

The installation program suggests the following installation path:
Programs\Hilscher\Sycon.

This path can also be altered by the user.

- The user should close all applications in the system.
- Place the CD into the CD ROM drive bay.
- Autostart switched On:
  To install from SyCon select **System Installation** in the Autostart screen.
- Autostart switched Off:
  In the Start menu select the **Execute** command and input **CD:\SyCon\setup.exe** or select this path directly from the Explorer (whereby CD corresponds to the drive letter of the CD ROM drive). Click on the **OK** button.
- Select the desired language. All further descriptions are for the English language.



- The installation settings ask ´Do you want to install the System Configurator SyCon?´. Click to **yes**. In the next line the question ´Do you want to install the OPC Server?´appears. Click to **yes**, too. Then you are asked for a license code. If you have a license code or you already have installed a SyCon with license code, then click to **yes**. If you have not a license code and want to install the SyCon in demo version, then click to **no**.

- The installation will ask for components to be installed additionally. These are the DLL of the PROFIBUS and the device driver.
  No configuration is possible without the DLL of the fieldbus system. The device driver is required in order to be able to directly access a built-in PC card. Both components can also be installed later by means of a Setup program.

The Setup program ends automatically after a successful installation.

2.2.2 The installation of the System Configurator for access to a remote PC

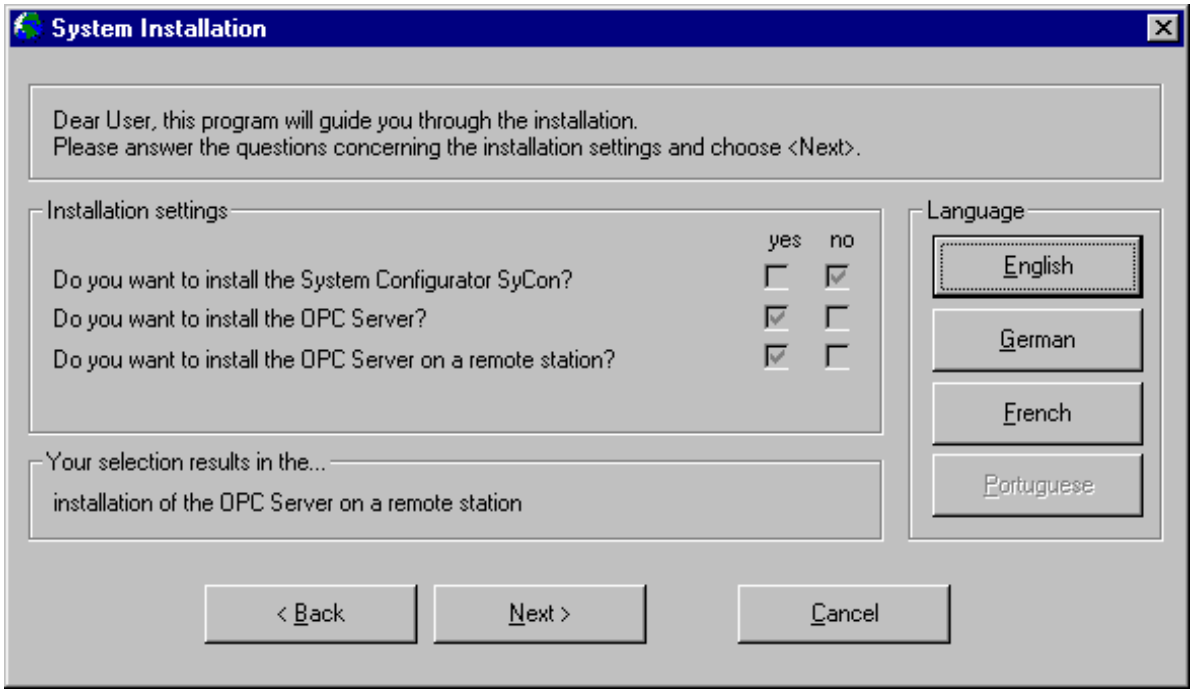
With this kind of installation you just copy a small count of DLLs which are necessary to communicate via OPC. Furthermore some registry entries are done with the same background.

After this installation there is no SyCon installed on this PC.

- The user should close all applications in the system.
- Place the CD into the CD ROM drive bay.
- Autostart switched On:
  To install from SyCon select **Install - System Configurator SyCon** in the Autostart screen.
- Autostart switched Off:
  In the Start menu select the
  Execute command and input **CD:\SyCon\setup.exe** or select this path directly from the Explorer (whereby CD corresponds to the drive letter of the CD ROM drive).

Click on the **OK** button.

- Select the desired language. All further descriptions are for the English language.



- The installation settings ask ´Do you want to install the System Configurator SyCon?´. Click to **no**. In the next line the question ´Do you want to install the OPC Server?´appears. Click to **yes**. Then you are asked ´Do you want to install the OPC Server on a remote station?´ Click to **yes** and then click the button **Next**. The installation for a remote PC is started.

The setup program ends automatically after a successful installation.

2.2.3 Remote Access

On the remote PC, the setup program installs or sets up the following components:

- ATL library (insofar as it is not already available).
- OPC Proxy/Stub-DLL
- System Configurator registration entries
- OPC Automation Interface

However, further settings are still necessary for the remote access to the OPC SERVER. These must partly be carried out manually and the configuration of the DCOM linkage is one of these.

The following chapters describe the procedure for the configuration of the DCOM linkage for the Server and for the Client PC.

A detailed description of all possible setting options can be called up from Microsoft or the OPC foundation.

www.opcfoundation.org or www.microsoft.com.

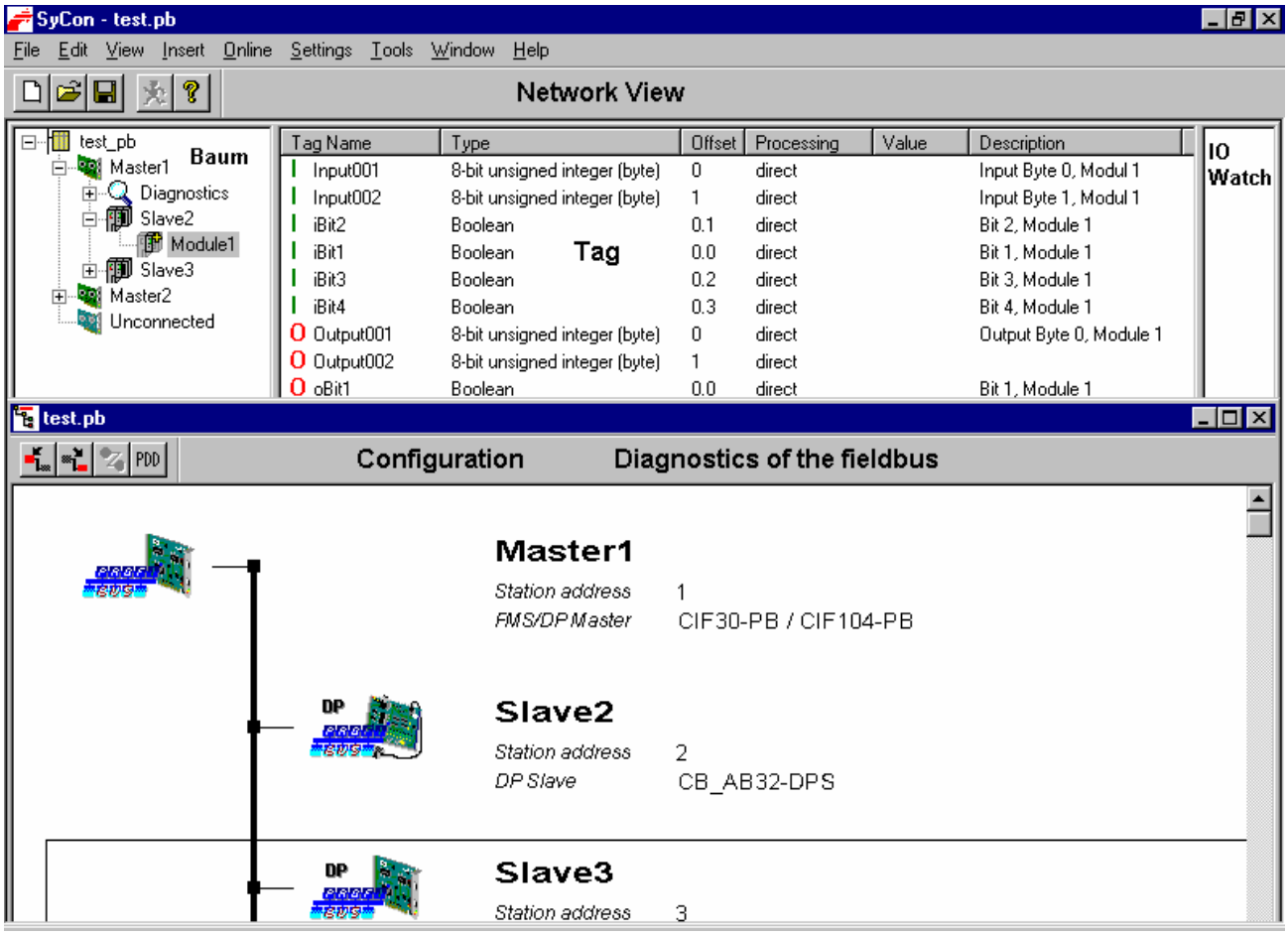
3 Putting into Operation

3.1 Overview

The OPC SERVER is depicted as an additional screen in the already existing SyCon System Configurator. This screen is called Network View.

The fieldbus configuration is created, as before, by means of the System Configurator (see SyCon description). The configuration of the OPC SERVER is created at the same time as the creation of a fieldbus configuration.

The Configurator was provided with a further symbol editor for the configuration of OPC TAGs. With its help, each input and output in the configuration can be allocated symbolic names. The allocation can be carried out directly during the configuration of a module or later in the Network View.



You can always see the same structure in the window Network View:

- OPC Server (HilscherGmbH.CifOpcServer)
- Project
- Master
- Slave/Node
- Modules
- Submodules (if available)
- Tags

Example from the view of the (local) OPC Client:

HilscherGmbH.CifOpcServer.Project.Master1.Slave2.Module1.Input1

Examples:

Level	PROFIBUS-DP	CANopen	DeviceNet	Interbus
Project	Project_pb	Project_CO	Project_DN	Project_IB
Master	Master1	Master	Master1	Master
Slave	Slave1	Node1	Device2	Device1
Module	Module1	PDO_1400	Poll	Module1
Submodule	SubModule001	Object6200Idx1	Input_Data	-
Tag	Input	Output	Input001	Input001

3.2 Configuration

The steps for starting up an available bus configuration and preparing it for OPC are:

- Configuration of the bus topology with the corresponding fieldbus configurator in the SyCon (in the first step only configure a slave next to the master). See also description in the SyCon manual to the used fieldbus (you find this on the delivered CD):

  - PROFIBUS: PB_OIE.PDF
  - Interbus: IB_OIE.PDF
  - CANopen: CO_OIE.PDF
  - DeviceNet: DN_OIE.PDF

- Edit all the required tags in the slave. For this, see also chapter "Configuration Dialogue of the tags".
- Download the configuration to the master or slave (menu **Online - Download**). Make a new download after a change.
- Ensure that the master or slave has been switched on for utilisation with OPC

A condition for the access to the data is a so-called OPC server license. This license must be available on at least one of the cards.

If such a license is not available, no values can be delivered from the hardware. Then the server will provide only "BAD values".

However you are able to simulate all values with the simulation mode. See the chapter ´Simulation Mode`

Check the license: Activate the menu **Online - Device Info**. It is important that you see in Driver1, Driver2 or Driver3 the information **OPCS**.

If not, please contect your distributor or directly Hilscher.

- Change into the debug mode in the menu **Online – Start Debugmode** and ensure that the configured slave is depicted "green".
- In the window Network View, select a module of this slave and look at the values of the available TAGs (Monitoring ON).
- In the OPC client, browse the TAGs of the SyCon and, in a first step, select only one for displaying. If you like to check, if an actual connection via OPC is available, read the chapter ´OPC Statistics`.

4 Settings

4.1 Start Options

After activating the menu **Settings - Start...** in the Network View the following dialogue appears.

A various of start options and modes can be set, which is offered by the OPC SyCon.

- Simulation mode ON/OFF

If Simulation mode ON/OFF is selected, the simulation mode is activated/deactivated by the next start of the OPC server.

- Start SyCon hidden if started via OPC

If this is selected, SyCon starts only hidden, if the start is activated via OPC.

- Start SyCon next Time with last Configuration

If this is selected, the last saved configuration in SyCon is loaded automatically after the next start of SyCon.

- Logical Network View visible

If this is selected, the Network View is also visible, if only SyCon without OPC is installed. Then the watch-list is also available.

- Fast Start ON/OFF

When Fast Start ON/OFF is selected, then no configuration DLL is loaded after the next start of SyCon, but the information of the configuration is read directly from a binary file.

For this reason no alterations to the configuration are possible. The starting of SyCon however is faster by several factors.

- Tag tracing ON/OFF

  If this is selected, SyCon makes a record of all functions of the Symbol Editor and its error messages.

  This option can only be active when the Fast Start Option is switched off, only then the configuration can be altered.

  This option is meant for error searching.

  This option lengthens the program running time.

- OPC tracing ON/OFF

  If this is selected, almost all accesses of an OPC client are recorded in a file. **This option lengthens the program running time.**

4.2 Call up of the Configuration Dialogues

The following dialogues are only opened if no OPC client is connected to SyCon and if SyCon wasn't start with the fast start mode.

If a double click is made on a tree element in the left hand part of the Network View, then, depending on the element, either

- the configuration dialogue of the master,
- the slave or
- a tag is called up.

Alternatively to the double click with the mouse, an element of this tree can also be selected and then with **Return** the corresponding dialogue can be called up.

Symbolic names are existing for:

- Master
- Slave/Nodes
- Modules and
- Submodules.

- **Configuration Dialogue of the Master**

The exact description of this dialogue can be taken from the documentation of the SyCon System Configurator.

Alterations to this configuration are automatically taken over by the configuration of the OPC server.

- **Configuration Dialogue of the Slave/Node**

The description of this dialogue can be also taken from the documentation of the SyCon System Configurator.

The use of the **Symbolic Names** calls up the Symbol Editor, whereby all the modules allocated here, as well as their symbolic names, are automatically taken up in the configuration of the OPC server.

- **Configuration Dialogue of the Tags**

The dialogue for the configuration of tags is only called up when the selection has been set to the lowest hierarchy level (the tags are then visible in the right hand part of the Network View) and when no OPC client is linked to this OPC server.

Click two times on the desired module. Alternatively, one can reach this dialogue when a module is chosen in the slave configuration dialogue and then the **Symbolic Names** button is pressed.

You find the description of that window in the chapter ´Symbolic Names´.

4.3 The Symbol Editor

The Symbol Editor serves for the symbolic definition of the OPC. Here, analogue to the bus configuration, it is always assumed that a tag can only be allocated in one module or sub module.

Default Tags

When allocating slaves or allocating modules within a slave, default tags are laid down with the data width allocated to this slave or module.

These default tags are created with the following syntax:

Inputs: **Input[Number]** or

Outputs: **Output[Number]**

The names of all components, as also of the default tags, can be altered at any time, whereby the following limitations apply to the symbols:

- **Uniqueness of the Tag names**
  The tag name must be unique within a module or sub module.
  This is monitored by the Symbol Editor.
- **Tag Name**
  Maximum of 32 characters whereby the period and empty spaces are not permitted.
  The entered tag must be unique within a module or sub module (this is monitored by the Symbol Editor).
- **The Description**
  Maximum 64 characters, no limitations in the use of the characters.
  No description is possible to a sub module.

Diagnostic Tags

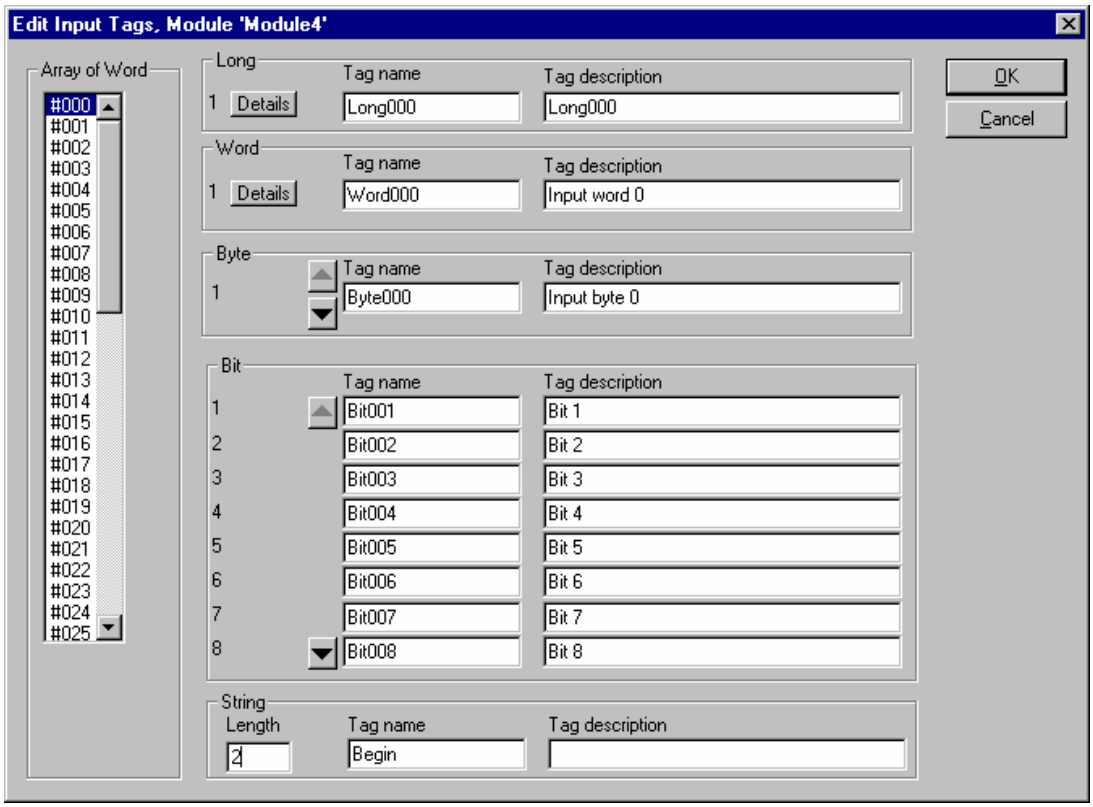
With these tags we are dealing with hardware specific tags which are not required for a normal data exchange.

Under the diagnosis tags there are found data such as firmware name, Firmware version and fieldbus specific values.

Diagnosis tags are automatically made available when a configuration download is made to the corresponding master or a slave card. The diagnosis tags for a master and a slave card are different. See also chapter ´Diagnostic Tags´.

The symol editor can be opened with the button **Symbolic Names** in the slave configuration window. Another possibility is to open in the Network View on the left site the context menu **Edit...** with the right mouse button, if a module or sub module is selected. Or you select in the middle of the Network View the desired tag with the right mouse button and open the menu **Edit...**.

Following window is opened:



In the left hand part of this dialogue there is always given the complete quantity of all data which this module contains as an array with the data width allocated to the module, even then when this module contains only a single tag.

When the module contains the data width "Word", there will appear in the regions for editing the byte and bit tags, arrow keys with which it is possible to scroll upwards and downwards.

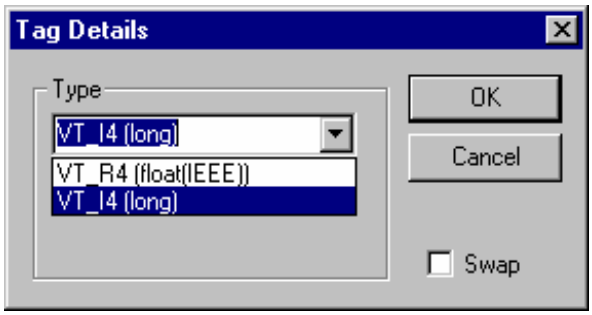
In this manner, in the selected array elements, all bits and all bytes can be provided with a special tag name.

The middle part of the dialogue serves for the editing of tags. In this way it is possible for tags to place different data types on the same offset address.

If the module possesses input as well as output data, then a further button is available to the user (**Output Tags - Input Tags**) in order to change between the configuration of the inputs and outputs.

The **Tag description** field can only be altered when there is at least one character in the associated **Tag name** field.

The button **Details** can be selected, if a tag name is inserted in the field **LONG.**



The data type long or float can be selected. If you select **Swap**, the value for the client is turned in the dual port memory. It does not matter, if byte- or word addressing is selected in the master settings. It is also possible to swap word modules. Then select the button **Details** for Word.

In the group **String** it is possible to define a tag as a data typ string. This tag begins from in the array selected offset and is as long as written in the field **Length**. If the given length exeeds the maximum of the module, you will pointed out and have to reduce the length.

4.4 Special Tags

It`s possible to define socalled special tags. With these tags you are able to support the data of modules which transmit in their data width values with different interpretation.

e.g. modules which have a data width of 16 bit and these 16 bits are devided up like following:

12 bit DA value

1 bit open circuit

1 bit overflow recognize

1 bit negative bit

Please note:

A special tag need more time to calculate the value as a standard tag.

You have following possiblities to define such a tag:

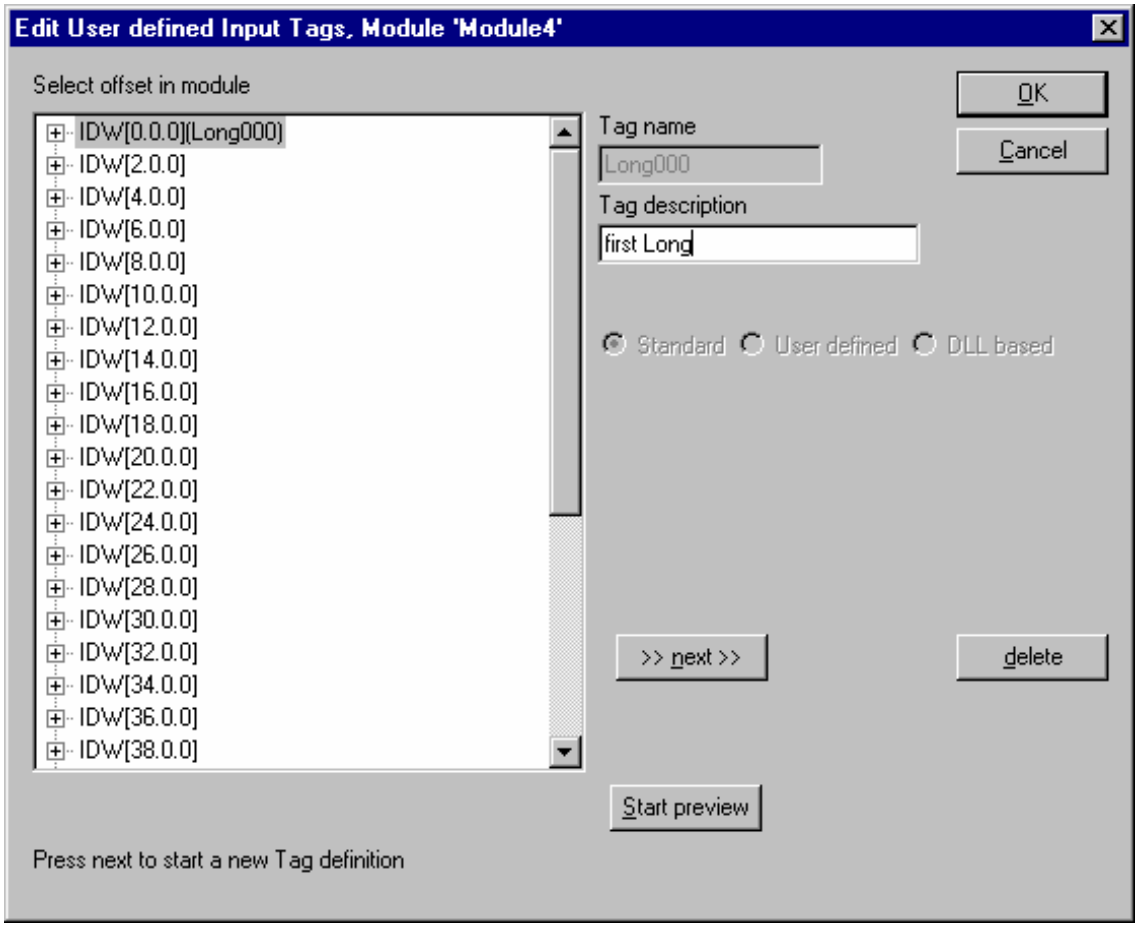
4.4.1 Definition by hand

SyCon offers a dialogue with which the user can specify a tag with a special functionality.

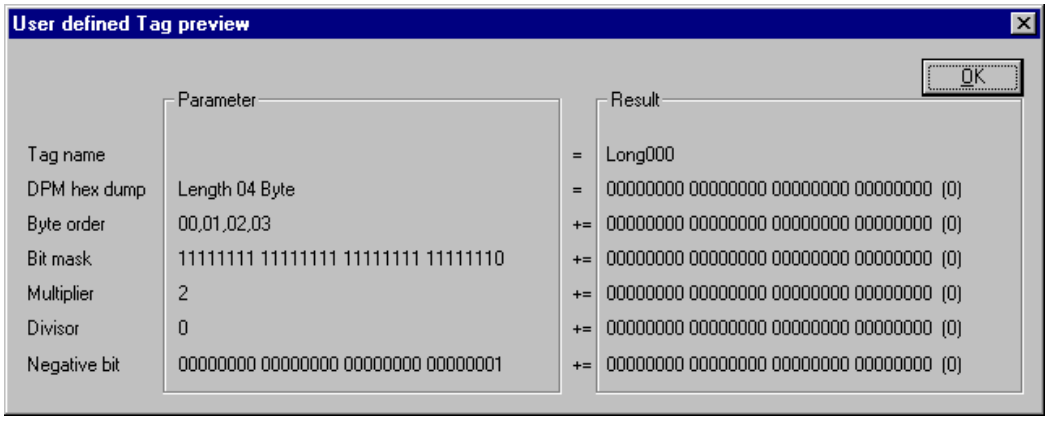
You are able to open this dialogoue if you do the following:

Switch to the Network View and select the module or the submodule in which you want to insert the tag.

Click in the middle of the Network View right mouse button. In the submenu select **Edit User Defined Tags...**. Now a dialogue appears in which you can define standard tags and special tags.



If you select the button **Start preview**, you get an overview from the choosen tag. Parameter, which are adjusted, are shown on the left side. On the right window you see the result.



Examples:

1. DLL based tag

Assume that you will define a Tag with the name (MyOwnDllTag) which should be an input tag based on a function called (MyOwnDllFunction) stored in the DLL (MyOwnDll)

Now you have to do the following actions.

- Select the start address of the tag in the tree in the left hand part.

- Press the button **>>next>>**

- Write down the name of the tag and press again the button **>>next>>** to write down the tag description.

- Select the radio button **Dll based.**

- Press the button **>>next>>**

- Now a combo box appears in which all DLL´s are insert, which exist in the folder **Convert** of the SyCon installation folder.

- Press the button **>>next>>**

- The next box appears. Select the functions which calculate the value of the tag.

- Press the button **>>next>>** and then **Save**.

2. Simple tag

Assume that you will define a Tag with the name (MyOwnSpecialTag) which should be an input tag with the following properties.

- Data width = 4 bytes
- The first 3 Bits of the value are of special meaning and should not be part of the value.
- The value which is send to an OPC client should start with 0
- The last bit mark that the value is negative.
- The OPC client gets the value in Intel format.

Now you have to do the following actions.

- Select the start address of the tag in the tree.
- Press the button **>>next>>**
- Insert the tag name (e. g. MyOwnSpecialTag)
- Press the button **>>next>>** and write down a tag description and select the radio button **Simple**.
- Possible select the typ of the tag (the selection is only possible with data widths of 32 bits) (Example.=VT_I4: 32-bit signed integer long; VT_R4: Float).
- Press the button **>>next>>**
- Selection of the byte disposition. The standard disposition is 0,1,2,3, with the value from the DPM without a change (Example= no change).
- Press the button **>>next>>** till the selection of the bit mask appears.
- Selection of the bit mask, which after a new order of the bytes is laid over them to select just the correct bits. (Example. First bit = Bit[03], then select button **>>next>>** and select second bit = Bit[30]).
- Press the button **>>next>>**

Selection of the bits, which which mark the value as negative.

- Press the button **>>next>>**
- Then you can give a multiplier for the tag.
- Press the button **>>next>>**
- Then you can give a divisor for the tag.
- Press the button **>>next>>** and then **Save**

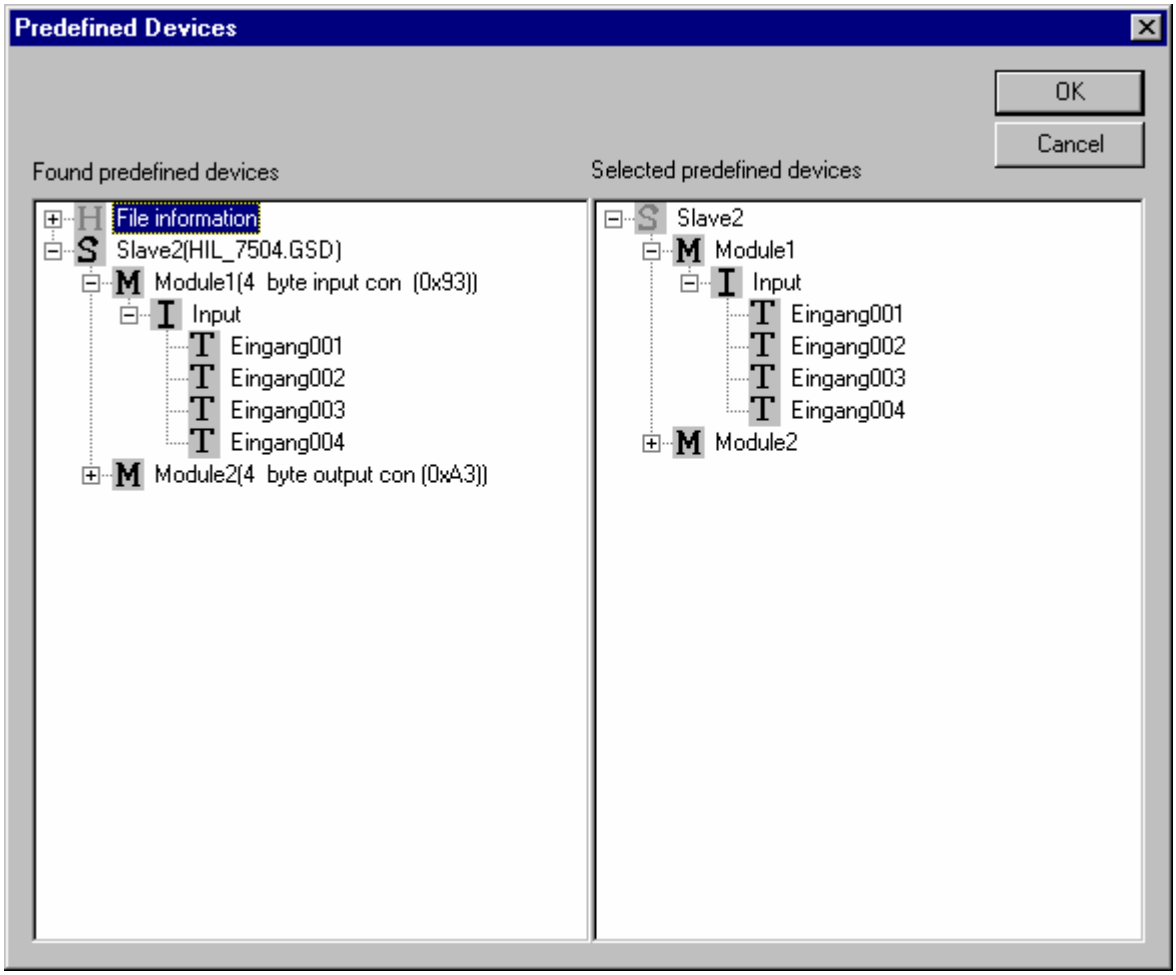
4.4.2 Definition in an ASCII file

With this interface it is possible to export parts of the configuration from the SyCon (only fieldbus PROFIBUS) into an ASCII file and to import these exported parts back to SyCon. **That is a SyCon-SyCon interface.**

These files must have the extension PDD (**P**re**d**efined **D**evices).

The export happens like following description:

- Load or constract the configuration
- The configuration window has to be the active window.
- Select the menu **File - Export - PDD**.
- Select a file or write down a new file name.
- In the following dialogue appears on the left site the actual configuration and on the right site the possible in the file found devices. If an existed file is selected then you cannot delete in this file founded devices from the right window.



- If you want to copy a further device from the left window to the right window you have to select this device and select then the button **>> All**.

You can also make the import over the **Short cut button (PDD).**)It is only for PROFIBUS-DP.) Select this button and click then with the mouse into the configuration window.

4.4.3 Definition in a DLL

This feature is implemented to give companies, who offer special modules, the possibility to offer their customers for these modules one or more DLL´s. These DLL´s habe the conversion of the values for these modules.

Please note the commentaries in the example for a DLL (it is installed in the folder **Convert**) to use the functionalities.

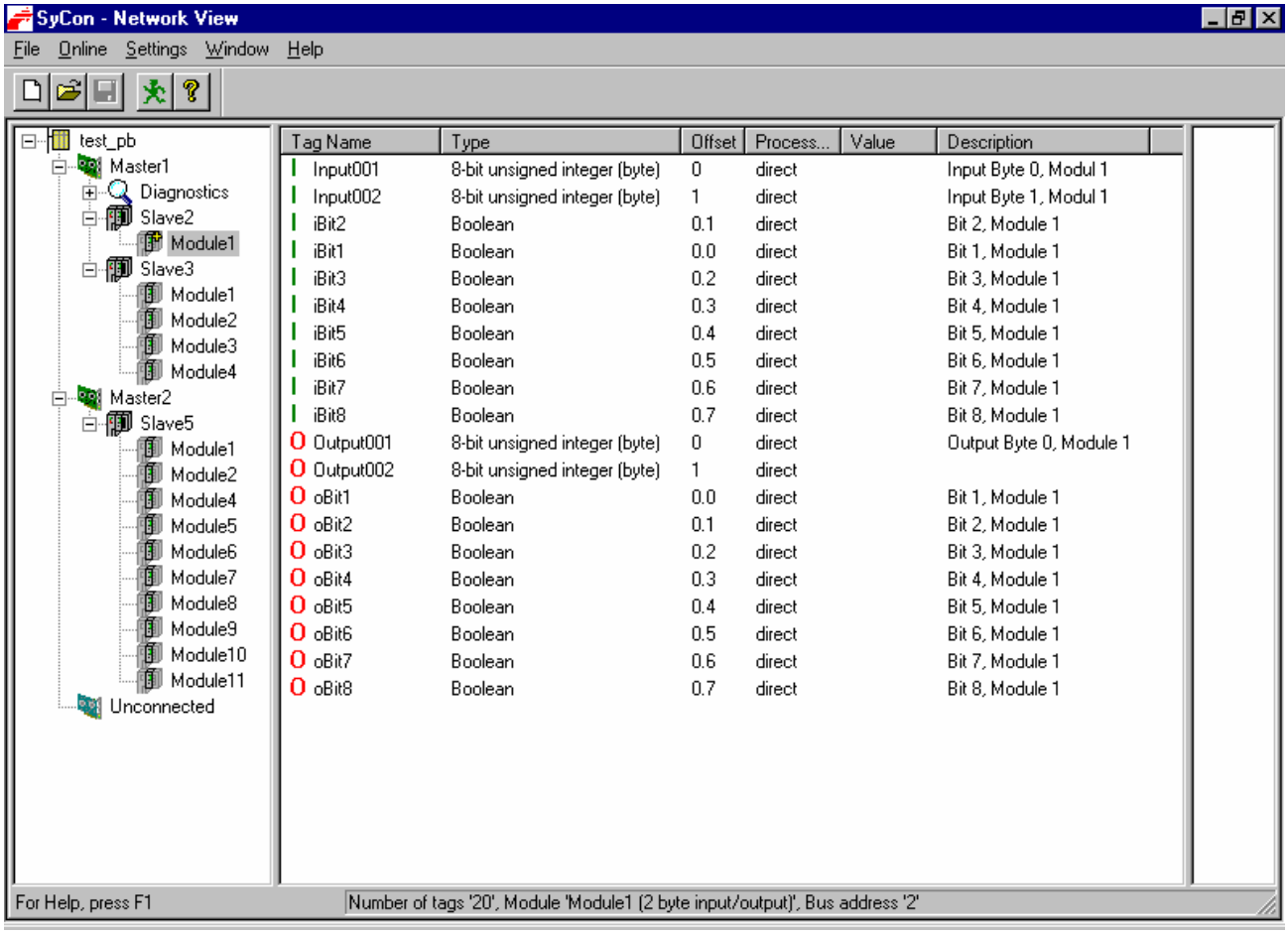
Restricions to the DLL:

The DLL has to exist during the SyCon is running in the folder convert.

Such a DLL has to give near the conversion functions also other functions.

5 Online Functions

5.1 Overview



In the left window the current configuration is shown in the form of a tree structure.

The middle of the Network View contains the tags which are contained in the selected tree.

Description of the display elements:

- **Tag Name**

  Name of the tag as defined in the Symbol Editor.

The order of representation can be changed by a click on the column header.

- **Type**

  Data type

  The order of representation can be changed by a click on the column header.

- **Offset** column

  Offset of the tag in the data picture of the card.

  The order of representation can be changed by a click on the column header.

- **Processing** column

  Output **direct**, if the tag is not simulated but it is read directly from the hardware.

  Output **sines value**, if the tag is simulated with a sines function (values from 0 ..200, wave length ~ 20 seconds).

  Output **random value**, if the tag is simulated with a coincidence function (values from 0..100, every second a new calculation).

  Output **ramp value**, if the tag is simulated with a ramp function (values from 0..100, every second a change).

- **Value** column

  Is updated only in the mode monitor or simulation. The present values are shown in decimal values. If a tag is demanded by a OPC client so it is given in brackets how often this tag is referenced over OPC.

- **Description** column

  Here is shown the descriptive text of the particular tag.

5.2 Watching Tags Online

You have the possibility with the OPC server to see tags with their data (only if they are assigned by a download). That is independent if a OPC client is connected or not.

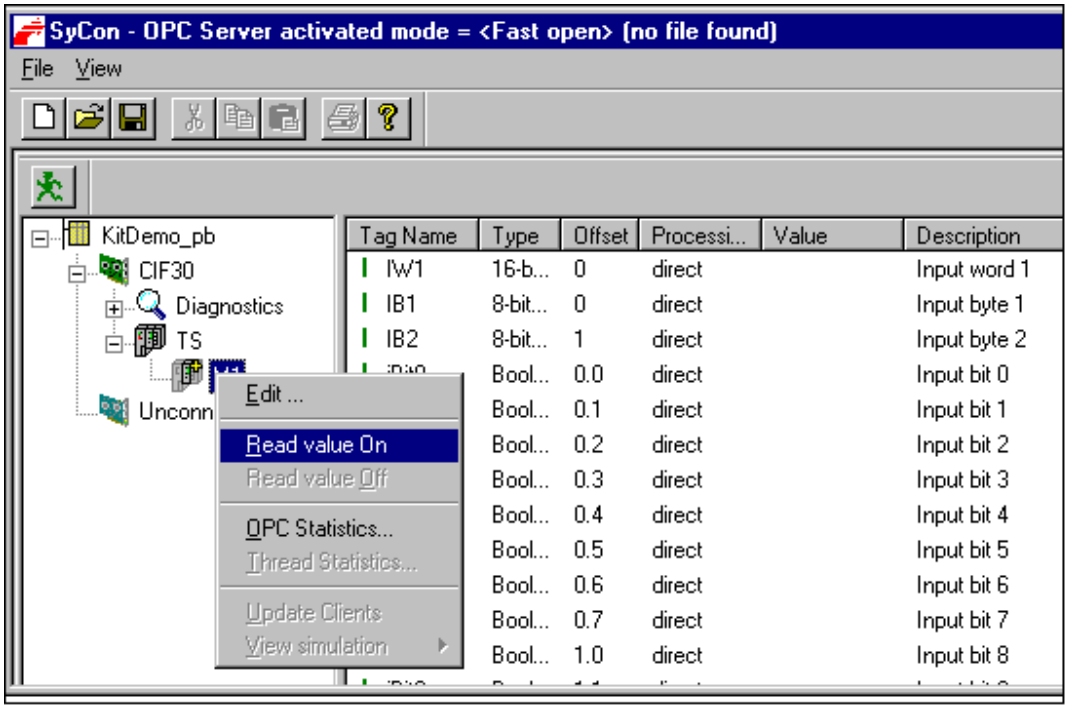
First select the corresponding module in the Network View.

Then select the menu **Monitoring On** via the context menu (right mouse button) or via the switch surface with the current figure or select the menu **Online-Monitor**.

The Monitoring is only available when at least one tag is defined.

When the Monitoring is on, the tag list is actualized with a cyclic time of 0.5 seconds.

If the tags are placed from the selected module at the hardware´s disposal, you can see in the column **Value** the corresponding values of the tags so far.



If the values cannot be read, then a driver error is output in the column **Value** and this is then an aid for further measures.

5.3 Write Values

Select the menu **Write Value** via the context menu from the middle of the Network View.

Following window is opened:

In the line **Current value** the present value is shown. You can write in **Value to write** the desired value, which shall written to the tag. Select the button **Write** to activate the writing. You can repeat the writing as often as you like till you close the window with the button **Exit**.

5.4 Simulation Mode

In that mode you are able to simulate values for a client. That means that you get values in a client although no hardware is available.

To take that mode select the menu **Settings-Start Options**. Select in the top of the window **Simulation mode ON/OFF**. You get a hint that this action will set all tags to ON at the next start of SyCon. Confirm that hint with **Ok** and close the window **Start Options** with the button **Ok**, too.

Follow that advice and start SyCon again. Open your made configuration or make a new configuration. Then change with the menu **Window-Network View** into the other window.

Select on the left side in the tree that module, which has the tags you want to simulate. Click to the tag with the right mouse button in the middle of the Network View. Mark in the context menu **View Simulation**. Select the simulation mode for the tag (sines, random...). You see in **Processing** which simulation is used when the Monitoring is on.

You can also select further tags from the module and take a simulation mode. Select the menu **Monitoring On** via the context menu (right mouse button) or via the switch surface with the current figure or select the menu **Online-Monitor**.

You find in **Processing** the selected simulation modes. Entries with **direct** means that no simulation mode is selected for that tag. If you want to finish the simulation mode select in the menu **Start Options** the simulation mode to **OFF** and start SyCon again.

5.5 Watch-List

The watch-list is able to display an process image for a list of input/output modules from different slaves. Composition of a watch list can be done in the window Network View.

The values are displayed cyclically. The number of tags (IOs) in a list is restricted to 60.

The watch-list requires special functions in the firmware! To use the watch-list you may need to download a new firmware. At the moment are not available every firmware with this special funtions.

Following master firmwares are up-to-date during writing this manual:

- **PROFIBUS-PB: V 1.208**
- **PROFIBUS-DPM: V 1.208**
- **Interbus-Master: V 2.088**
- **CANopen-Mater V 1.092**
- **DeviceNet-Master V 1.104**

5.5.1 Formation a Watch-List

The actual configuration is displayed as a tree in the Network View. The tags for a watch-list can be selected in the tree in the left side or in the center of the window. The selection can be made in every order.

- **Selecting from the tree in the left side of the Network View**

The desired branch (can be hole project) can be taken over with Drag & Drop operation into the right window.

Drag with the left mouse button the wanted item (project-, master-, slave-, module- or submodule-item) to the right window. There the selection is also prensented in a tree.

Multiple selection of items for Drag & Drop are not possible in this view. Simply drag and drop items are carried out one after the other.

- **Selecting from the list in the middle of the Network View**

If you have selected a module or submodule in the left side of the window then appears in the middle of the window the appropriated tags.

You can drag these tags single or multiple (multiple selection with SHIFT or CTRL key in combination with left mouse button) to the left window.

5.5.2 The Context Menu in the Right Side of the Network View

Click with the right mouse button in the right window. You get the context menu.

- **Menu-item Delete Tree**

Select this item to delete the hole selection.

- **Menu-item Delete Branch**

Select a branch first you are going to remove activate popup-menu and select **Delete Branch**.

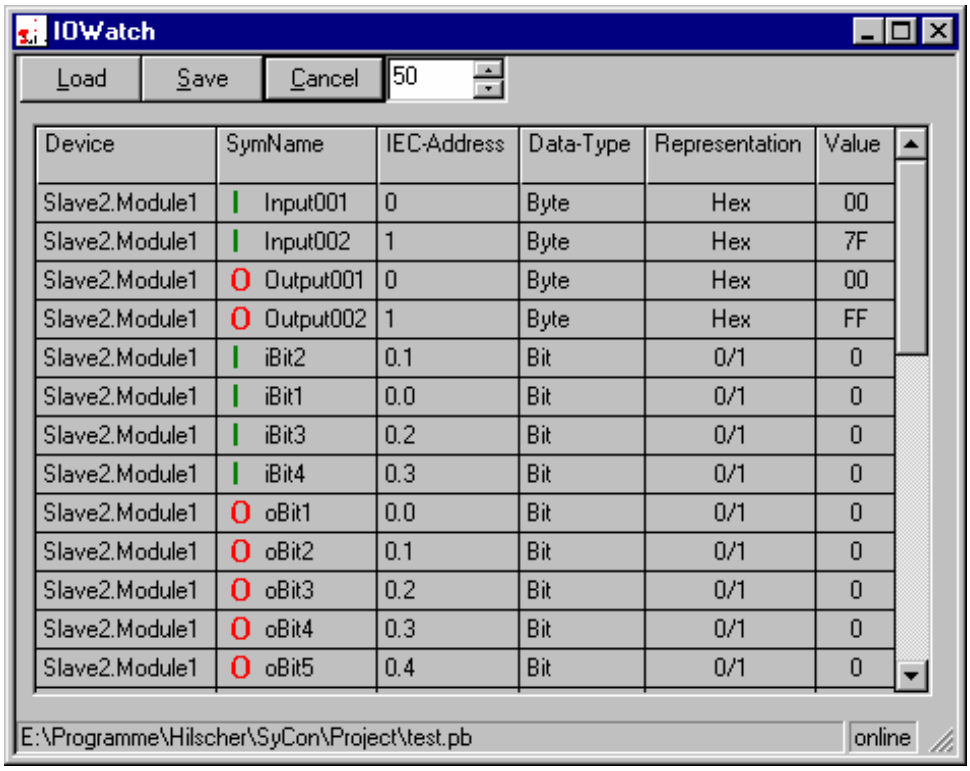
- **Menu-item Accept Watch-List**

Select this item in order to display the selected list in the watch-list.

- **Menu-item Start IOWatch**

This menu starts the watch-list without a specific list.

With the button **Load** in the watch-list dialog-box you can load a file (.iom) with a list.



5.5.3 Watch-List Dialog

The IOs in the watch-list are represented row-oriented. There are displayed:

- Device
- Symname
- IEC-Address
- Data-Type
- Representation: Hex, Float, signed/unsigned Decimal, Char, True/False, 0/1, On/Off, Bit-Pattern
- Value

The order of IOs in the list is editable using Drag & Drop.

The complete list or part of it can be copied with Drag & drop into another application (MS-Excel for example).

Representation for each tag can be set in a popup-listbox (double click on a corresponding cell in column Representation). You can write values for outputs while activating corresponding cell in a value column and changing the value. In case of Bit-Pattern you can change bit values simply by clicking with left mouse button on desired bit position.

The Update Rate

You can set the update rate in SpinButtonCtrl varying from 1 up to 1000. While holding the scroll button pressed the value is changing in greater steps.

Buttons in the Dialog Box

- **Load**

Prompts FileDialog, where you can select a file (.iom) with desired list and load into the watch-list.

- **Save**

The button **Load** activates a FileDialog, where you can select a file (.iom), in which you are going to save your actual watch-list.

- **Cancel**

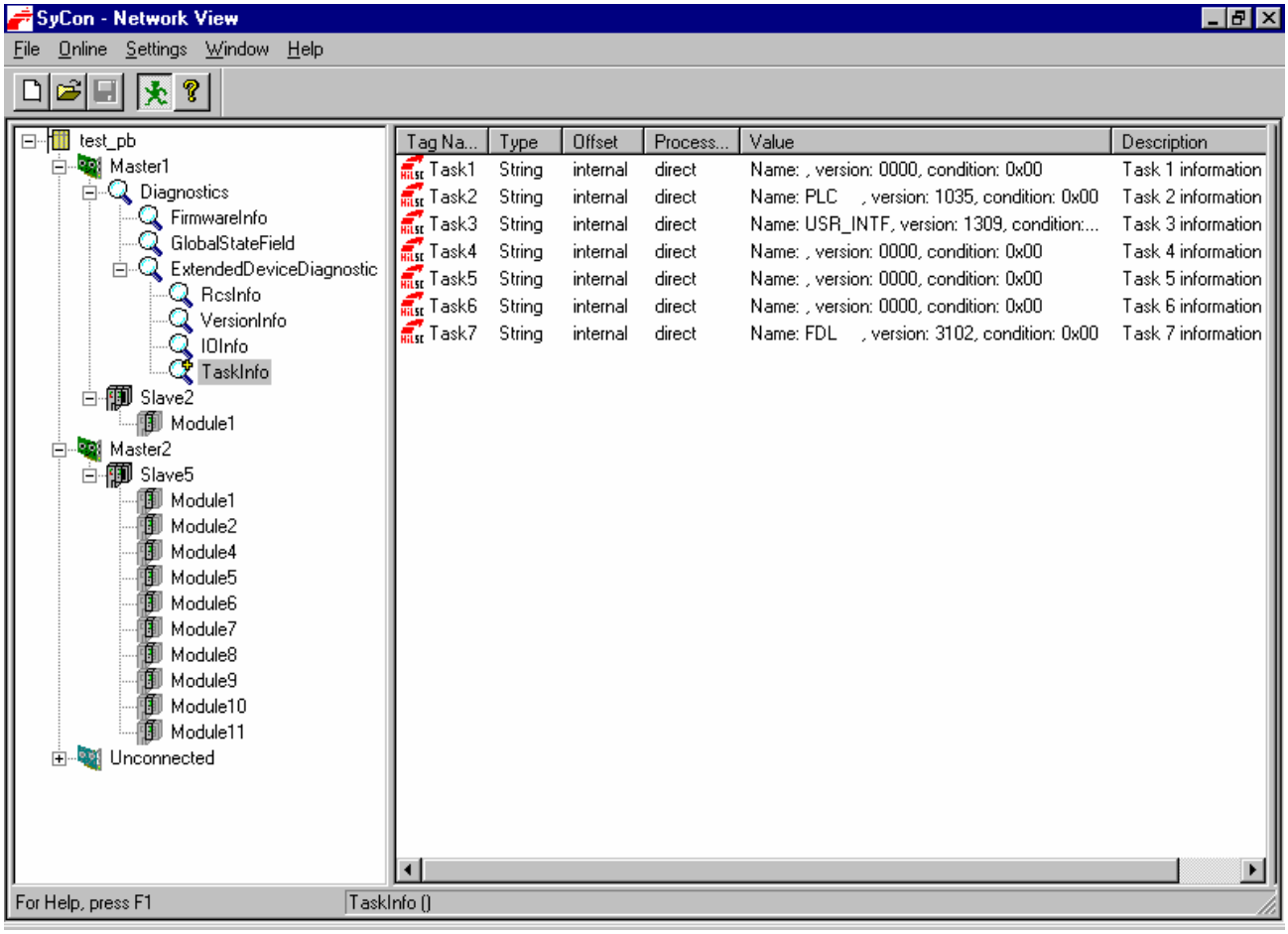
The button **Close** ends the watch-list dialog.

6 Diagnostics

6.1 Diagnostics data

You get a new branch called **Diagnostics** in the tree left in the Network View after a configuration download. Specific hardware and fieldbus specific diagnostic information are offered.

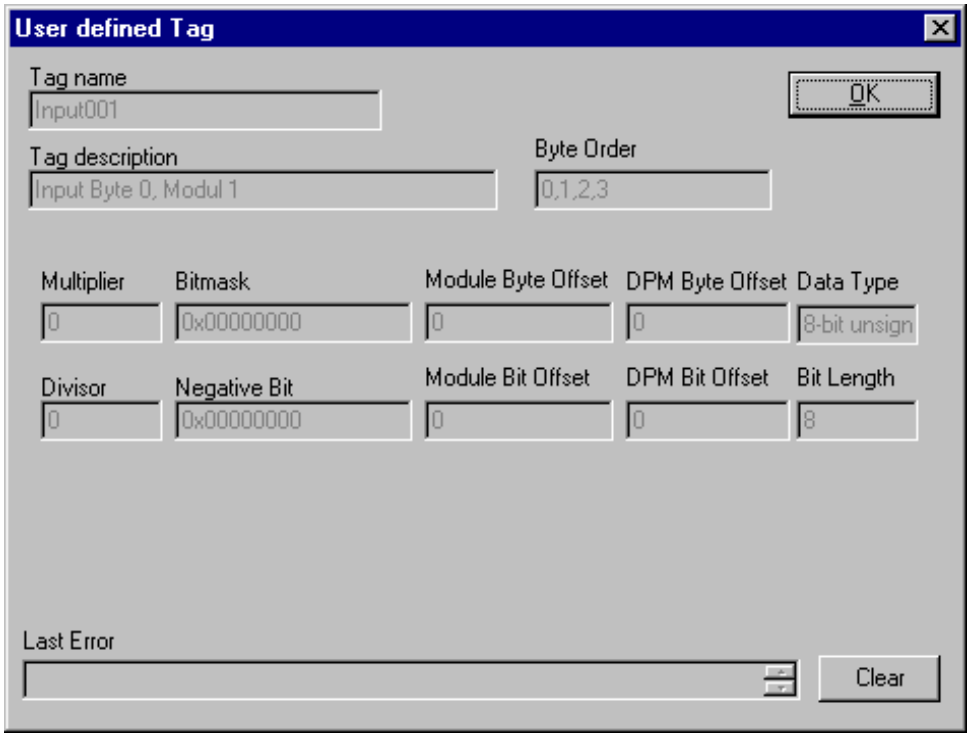
Because of the circumstances that a master and a slave card give different diagnostics they are not described closer here.



6.2 Tag Info

Click on the desired tag with the right mouse button in the center of the Network View. Select in the context menu the menu point **Tag Info**.

You get a window with following information:



You see the tag name and the tag description, if used, in the upper area.

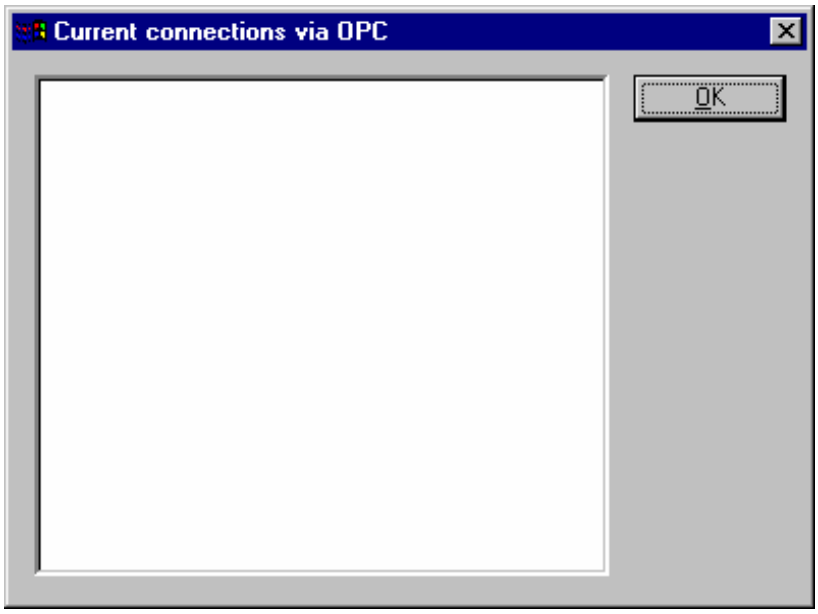
The information under Message Based Tags shows on which data the tag is based. That is a description for the process how the tags are established. This is not relevant for the standard user.

6.3 OPC Statistics

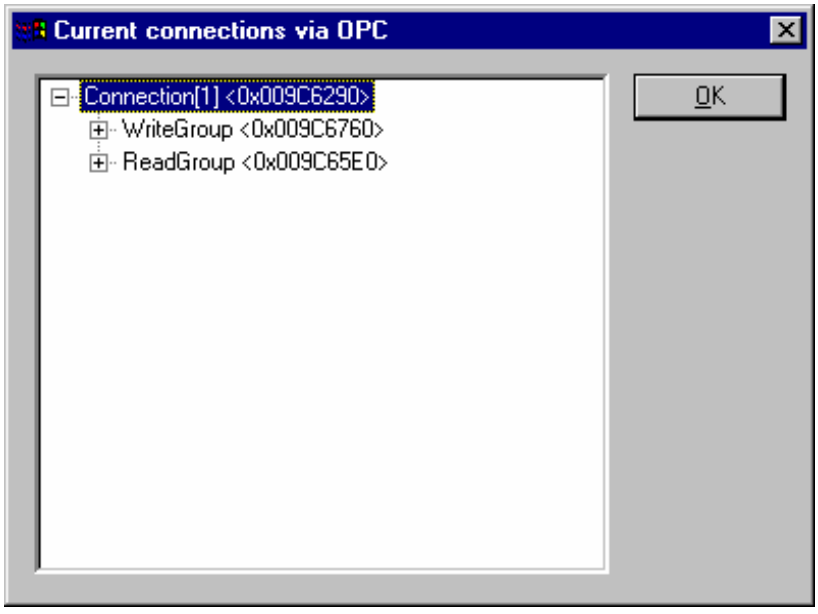
In that menu you get information about actual connections via OPC.

Place the mouse cursor in the left part of the Network View and select in the context menu **OPC Statistics...**.

If no connections exist via OPC to a client, the window is empty.

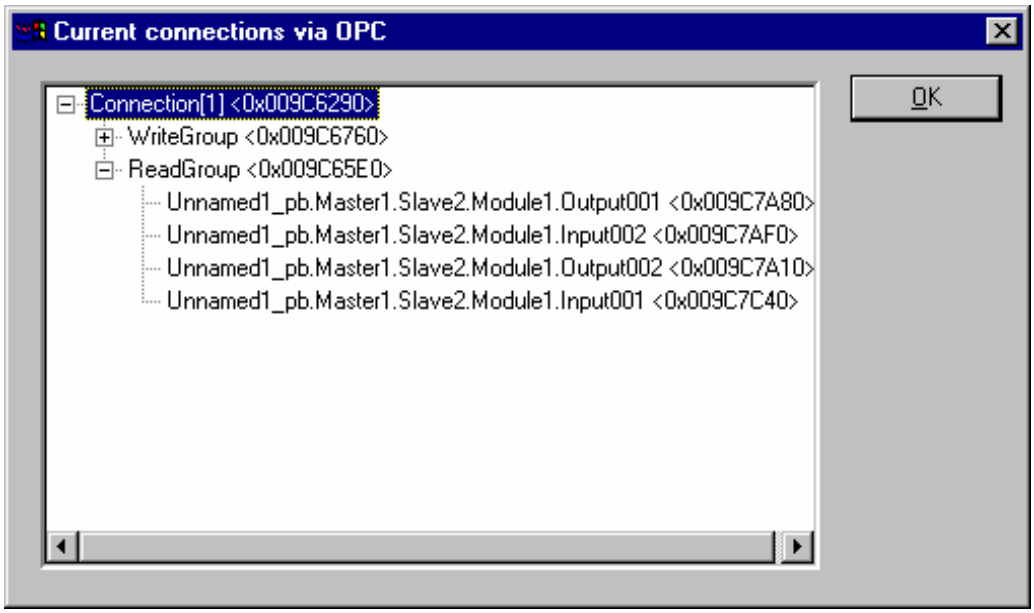


If at least one connection via OPC exists you see an entry in that window.

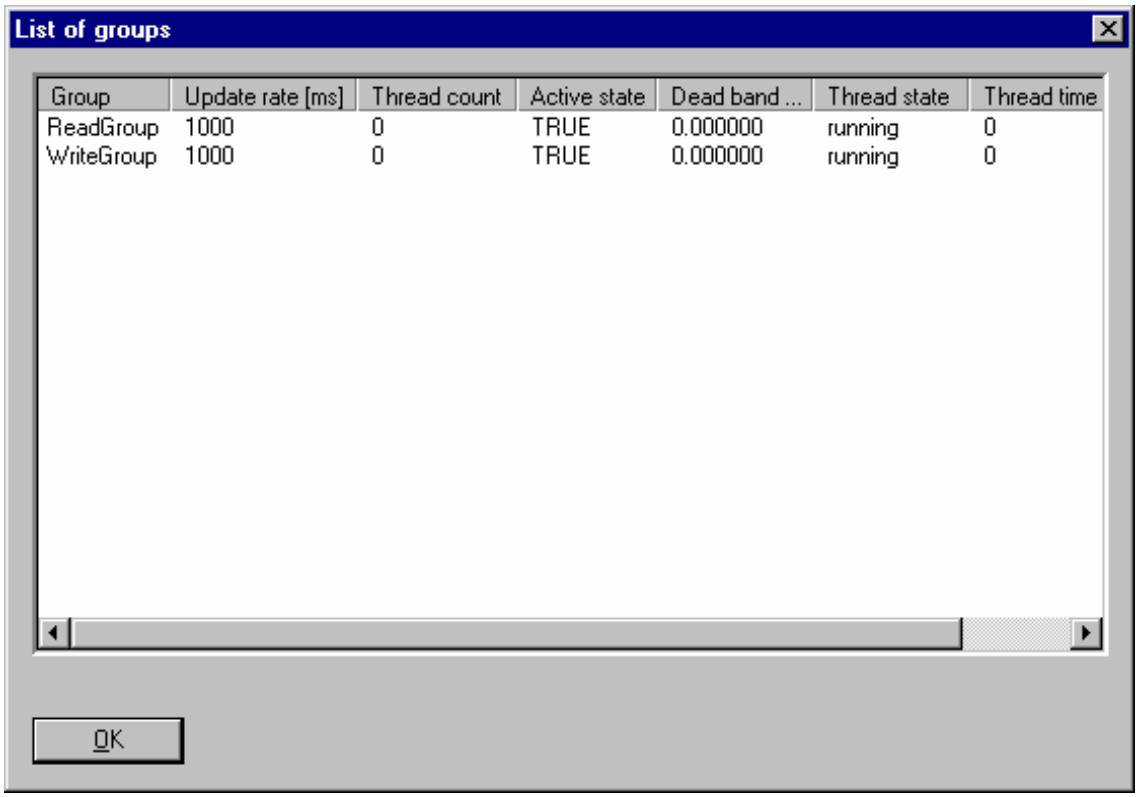


Here is a connection called Connection[1]. You see an abstract representation of the client. This is because of the client does not support the text representation. You see two groups WriteGroup and ReadGroup in the client.

You find in the group the defined tags in each case.

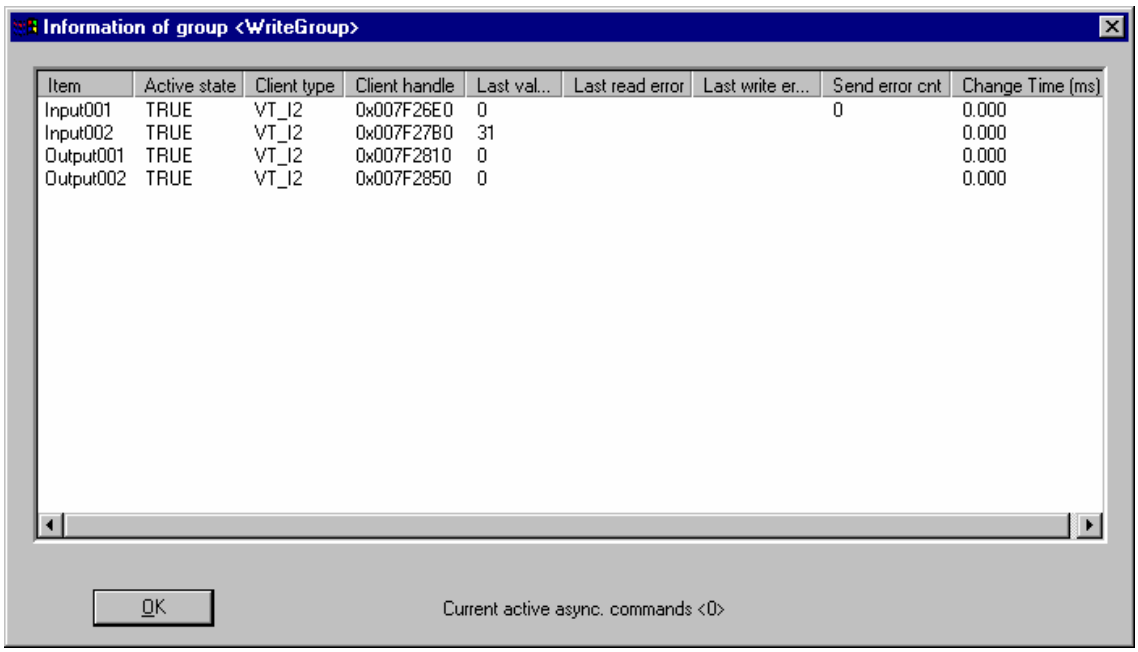


Make a double click on the first line to the client. It appears a list from all existing groups of that client with further information.



Make a double click on a group. The group information of that group appears. This group information lists all tags of that group and shows the actual state.

The line **Client Type** shows the data type of the tag, which was defined in the client. That type can be different from the type in the client. The client defines the data type which is used by the client.



Further information in the statistics are the last value of the tags, last read error and write error, if appeared and so on.

If you make a double click to a tag in the window **Current Connetions via OPC** the window **Statistics of Tag** is opened.

For more information see chapter ´Statistics of Tags´.

6.4 Statistics of Tags

Click in the center of the Network View with the right mouse button to open the window **Statistics of Tag**. Select in the context menu the menu point **Statistics**.

You get a window with following appearance:

All information on the left side is from the client. Against them you see on the right side the information of the server.

The interfaces with the version 1.0A and V 2.0 (OPC specification) are supported from version 2.6xx (sycon.exe) of the OPC server. From there the information can be OPC 1.0A or OPC 2.0 in the first line. That is dependent on what the client supports.

Left side:

A synchronous and/or an asynchronous read and/or write can be generally supported by the client. It is also distinguished, if the values are from the cache or from the device. The shown values are values, which are read or written at the moment. The reference counter counts how often the tag is referenced by the client.

Right side:

The information from the OPC server are also present values. You get information about the quality of the tag, if it is good or bad and the real value can be read. Behind **Count** (for the interface 1.0A) and **Count OPC 2.0** is given how often the value from the server is updated by the client. This number is unequal zero if the client does not fetch the values fast enough and then the server itself gives the values to the client. The last read time is also shown actually and is always updated. The intern data type of the tag in the OPC server and the access order are given.

7 FAQ

7.1 Download is working but Tags cannot be displayed Online

This can be caused by the following:

- No enabling for OPCS in entered on the card (see chapter ´Configuration´)
  Ask your distributor for an enable code for the card.
  For this the serial number of the card is required (menu **Online - Device Info**).
- An old device driver is installed on the PC (version earlier than 2.010). Please install the device driver found on the CD.

7.2 OPC Start up without a physical Bus

Select a tag in the center of the Network View and, in the context menu, select a permissible function for this tag under **View simulation**. This tag then delivers values, simulated by OPC to the client. The time span in which a new value is calculated is fixed at one second.

7.3 Does my OPC Client receive Values and, if so, which Values

Select a tag and select the menu **Statistics...** in the context menu. If the value for "Count" under the title "Last send data" counts up, then this value is transmitted to a client. The value itself is displayed under "Value" in the same title.

Caution: It can only be said with certainty that **your** client receives this value when **your** client is the only one currently connected to the SyCon.

7.4 How Many Clients are Linked to SyCon

Place the mouse cursor in the left part of the Network View and select **OPC Statistics...** in the context menu.

A list of all OPC clients currently connected to SyCon will appear.

If the window is empty, no client is connected via OPC.

Every active group and every active Item is shown with the green A.

Every inactive group or inactive Item is characterised with a red S.

7.5 Unconnected Region

This region is used for internal administration of slaves, which are no longer allocated to a master. These are not visible via the OPC browser Interface as no type of readable data can be kept here.

7.6 Why Do I Have No Diagnostic Data Available

The reason for this is that the initialisation of the driver with the given Board number is not functioning properly.

- Has the allocation between board number and card been altered in the driver set up program?

This can only be rectified when the old allocation is restored again or when a renewed download on to the card has been carried out.

- Do you posses the enable code for the OPC on the card.

See "**Download is Working but Tags Cannot be Displayed Online".**

8 Error Numbers

8.1 CIF Device Driver (Dual-port memory) Error Numbers (-1 .. -49)

These is the list of error numbers of dual-port memory access using the CIF Device Driver.

Error Number	Description
-1	Driver: Board not initialised The communication board is not initialized by the driver. No or wrong configuration found for the given board, check the driver configuration. Driver function used without calling DevOpenDriver() first.
-2	Driver: Error in internal ‘Init state‘
-3	Driver: Error in internal ‘Read state‘
-4	Driver: Command on this channel is active
-5	Driver: Unknown parameter in function occured
-6	Driver: Version is incompatible The device driver version does not correspond to the driver DLL version. From version V1.200 the internal command structure between DLL and driver has changed. Make sure to use the same version of the device driver and the driver DLL.
-10	Device: Dual port memory RAM not accessable (board not found) Dual-ported RAM (DPM) not accessible / no hardware found. This error occurs, when the driver is not able to read or write to the Dual-port memory. Check the BIOS setting of the PC Memory address conflict with other PC components. Try another memory address, check the driver configuration for this board, check the jumper setting of the board.
-11	Device: Not ready (RDY flag=Ready flag failed) Board is not ready. This could be a hardware malfunction or an other program writes inadmissible to the dual-port memory.
-12	Device: Not running (RUN flag=Rrunning flag failed) The board is ready but not all tasks are running, because of an initialisation error. No data base is loaded into the device or a wrong parameter can causes that a task can't initialise.

Table of CIF Device Driver error numbers

Error Number	Description
-13	Device: Watch dog test failed
-14	Device: Signals wrong Operating System version No license code found on the communication board. Device has no license for the used operating system or customer software. No firmware or no data base on the device is loaded.
-15	Device: Error in dual port memory flags
-16	Device: Send mailbox is full
-17	Device: Function PutMessage timeout No message could be send during the timeout period given in the DevPutMessage() function. If you use an interrupt, then check the interrupt on the device and in driver setup. These settings have to be the same! Is an interrupt on the board set? Is the right interrupt set? The interrupt could already be used by an other PC component, also if the operating system reports it as unused. If you use polling mode, then make sure that no interrupt is set on the board and that polling is set in the driver setup. These settings have to be the same! Device internal segment buffer full and therefore PutMessage() function is not possible, because all segments on the device are in use. This error occurs, when only PutMessage() is used but not GetMessage(). HOST flag is not set for the device. No messages are taken by the device. Use DevSetHostState() to signal a board an application is available.
-18	Device: Function GetMessage timeout No message received during the timeout period given in the DevGetMessage() function. If you use an interrupt, then check the interrupt on the device and in driver setup. These settings have to be the same! Is an interrupt on the board set? Is the right interrupt set? The interrupt could already be used by an other PC component, also if the operating system reports it as unused. If you use polling mode, then make sure that no interrupt is set on the board and that polling is set in the driver setup. These settings have to be the same! The used protocol on the device needs longer than the timeout period given in the DevGetMessage() function.
-19	Device: No message available

Table of CIF Device Driver error numbers

Error Number	Description
-20	Device: Reset command timeout The board is ready but not all tasks are running, because of an initialisation error. No data base is loaded into the device or a wrong parameter can causes that a task can't initialise. The device needs longer than the timeout period given in the DevReset() function. Using device interrupts. The timeout period can differ between fieldbus protocols. If you use an interrupt, then check the interrupt on the device and in driver setup. These settings have to be the same! Is an interrupt on the board set? Is the right interrupt set? The interrupt could already be used by an other PC component, also if the operating system reports it as unused. If you use polling mode, then make sure that no interrupt is set on the board and that polling is set in the driver setup. These settings have to be the same!
-21	Device: COM flag not set The device can not reach communication state. Device not connected to the fieldbus. No station found on the fieldbus. Wrong configuration on the device.
-22	Device: IO data exchange failed
-23	Device: IO data exchange timeout The device needs longer than the timeout period given in the DevExchangeIO() function. If you use an interrupt, then check the interrupt on the device and in driver setup. These settings have to be the same! Is an interrupt on the board set? Is the right interrupt set? The interrupt could already be used by an other PC component, also if the operating system reports it as unused. If you use polling mode, then make sure that no interrupt is set on the board and that polling is set in the driver setup. These settings have to be the same!
-24	Device: IO data mode unknown
-25	Device: Function call failed
-26	Device: Dual-port memory size differs from configuration
-27	Device: State mode unknown

Table of CIF Device Driver error numbers

Error Number	Description
-30	User: Driver not opened (device driver not loaded) The device driver could not be opened. Device driver not installed. Wrong parameters in the driver configuration. If the driver finds invalid parameters for a communication board and no other boards with valid parameters are available, the driver will not be loaded.
-31	User: Can't connect with device board
-32	User: Board not initialised (DevInitBoard not called)
-33	User: IOCTRL function failed A driver function could not be called. This is an internal error between the device driver and the DLL. Make sure to use a device driver and a DLL with the same version. An incompatible old driver DLL is used.
-34	User: Parameter DeviceNumber invalid
-35	User: Parameter InfoArea unknown
-36	User: Parameter Number invalid
-37	User: Parameter Mode invalid
-38	User: NULL pointer assignment
-39	User: Messagebuffer too short
-40	User: Size parameter invalid
-42	User: Size parameter with zero length
-43	User: Size parameter too long
-44	User: Device address null pointer
-45	User: Pointer to buffer is a null pointer
-46	User: SendSize parameter too long
-47	User: ReceiveSize parameter too long
-48	User: Pointer to send buffer is a null pointer
-49	User: Pointer to receive buffer is a null pointer

Table of CIF Device Driver error numbers

1.000	If the operating system of the device reports an initialisation error, then a value of 1000 will be add to the error number and shown to the user

8.2 CIF Serial Driver Error Numbers (-20 .. -71)

These is the list of error numbers using the serial driver.

Error Number	Description
-20	Driver: No COM port found or COM port already in use.
-21	Driver: COM port already opened
-22	Driver: Function call into driver has failed
-23	Driver: Internal driver error
-24	Driver: Could not create read thread
-25	Driver: Could not create read event
-26	Driver: Could not create write event
-27	Driver: Could not create timer event
-28	Driver: Error by writing data
-29	Driver: Wrong COM state
-30	Driver: COM state error is set
-31	Driver: COM buffer setup failed
-32	Driver: COM set timeout failed
-33	Driver: Receive buffer overrun
-34	Driver: Receive buffer full
-35	Driver: Send busy
-36	Driver: Error during close driver
-40	User: COM port not opened
-41	User: Invalid handle value
-42	User: Invalid COM number
-43	User: Size parameter invalid
-44	User: Size parameter zero
-45	User: Buffer pointer is NULL
-46	User: Buffer too short
-47	User: Setup error

Table of CIF Serial Driver error numbers

Error Number	Description
-50	User: Send message, timeout error
-51	User: Could not send a message Cable not connected. Wrong cable. Device does not respond.
-52	User: Send message, no device connected
-53	User: Error by send message, message receiving
-54	User: Telegram collision
-55	User: Telegram, no acknowledgement received
-56	User: Telegram, noise
-57	User: Telegram, data overrun
-58	User: Telegram, parity error
-59	User: Telegram, framing error
-60	User: Telegram, unknown error
-70	User: Timeout by receive a message
-71	User: No message received

Table of CIF Serial Driver error numbers

8.3 RCS Error Numbers (4 .. 93)

This is the list of error numbers returned by the RCS (Realtime Communication System), that is the operating system of Hilscher devices. The error number is returned in an answer message. Command messages and answer messages are used to communicate between the application (e.g. the system configurator) and the Hilscher device. An example of this communication is the download of a configuration.

Error Number	Description
4	Task does not exist
5	Task is not initialised
6	The MCL is locked
7	The MCL rejects a send command because of an error
20	The user will doanload a database into the device that is not valid for this device type.
21	Daten base segment not configured or not existend
22	Number for message wrong during download
23	Received number of data during download does not match to that in the command message
24	Sequence identifier wrong during download
25	Checksum after download and checksum in command message do not match
26	Write/Read access of data base segment
27	Download/Upload or erase of configured data base type is not allowed
28	The state of the data base segment indicated an error. Upload not possible
29	The access to the data base segment needs the bootstraploader. The bootstraploader is not present
30	Trace buffer overflow
31	Entry into trace buffer too long
37	No or wrong licence. The OEM licence of the system configurator allows only communication to devices that have the same licence inside
38	The data base created by the system configurator and the data base expected by the firmware is not compatible
39	DBM module missing

Table of RCS error numbers (answer message)

Error Number	Description
40	No command free
41	Command unknown
42	Command mode unknown
43	Wrong parameter in the command
44	Message length does not match to the parameters of the command
45	Only a MCL does use this command to the RCS
50	FLASH occupied at the moment
51	Error deleting the FLASH
52	Error writing the FLASH
53	FLASH not configured
54	FLASH timeout error
55	Access protection error whil deleting the FLASH
56	FLASH size does not match or not enough FLASH memory
60	Wrong structure type
61	Wrong length of structure
62	Structure does not exist
70	No clock on the device
80	Wrong handle for the table (table does not exist)
81	Data length does not match the structure of this table
82	The data set of this number does not exist
83	This table name does not exist
84	Table full. No nore entries allowed
85	Other error from DBM
90	The device info (serial number, device number and date) does already exist
91	Licence code invalid
92	Licence code does already exist
93	All memory locations for licence codes already in use

Table of RCS error numbers (answer message)

8.4 Database Access Error Numbers (100 .. 130)

The following table lists the error numbers of the database access errors

Error Number	Description
100	Database already opened
101	Dataset could not be opened
103	Error while opening database occured
104	No valid path name
105	No connection to data base. Call function DbOpen().
106	Error in parameter
107	Error during opening a table
108	Nullpointer occured
109	Table not opened. Call function OpenTable() first.
110	The first record is reached
111	The last record is reached
112	Unknown type in the record found
113	Data has to be truncated
114	No access driver installed on the system
115	Exception received
116	This table is set to read only
117	There is no data set in the table
118	The requested table could not be edit
119	An operation could not be completed
120	User gives an unexpected length in WritsDs().
121	An assertion failed
122	DLL not found
123	DLL couldn´t be freed
124	Specified function not found in the DLL
125	ODBC Funktion returns an error
126	Count of data bytes in the record exeeds 1938
127	DBM32 DLL is not loaded
128	Field with the given index was not found
129	This table contains no records
130	Invalid character (' ') found in a Table or Column

Database Access Error Numbers

8.5 Online Data Manager Error Numbers

8.5.1 Online Data Manager Error Numbers (1000 .. 1018)

The following table lists the error numbers of the Online Data Manager.

Error Number	Description
1000	Driver OnlineDataManager not opened
1001	Initialization of the OnlineDataManager has failed
1002	No DriverObject found. OnlineDataManager Sub DLL not found.
1003	No DeviveObject found. Device not found.
1004	Application not found
1010	Application has requested an unknown event
1011	Application has requested an unknown function mode, operating mode. Known function modes, operating modes are Reset, Download, Register Server, Unregister Server.
1012	Application has requested an unknown command
1013	Message Server already exists
1014	Message Server not registered
1015	Device already in use
1016	Device not assigned
1017	Device has changed
1018	Command active

Table of the Online Data Manager Error numbers

8.5.2 Message Handler Error Numbers (2010 .. 2027)

The following table lists the error numbers of the Message handler of the Online Data Manager.

2010	Message handler: Messagebuffer empty
2011	Message handler: Messagebuffer full
2021	Message handler: Invalid Message ID (msg.nr)
2022	Message handler: No entry
2023	Message handler: Message already active
2024	Message handler: Wrong Application
2025	Message handler: Message Timeout
2026	Message handler: Wait for Delete
2027	Message handler: No cyclic Message

Table of the Error Nubrs of the Message Handler of the Online Data Manager

8.5.3 Driver Functions Error Numbers (2501 .. 2512)

The following table lists the error numbers of the Driver Functions of the Online Data Manager.

2501	OnlineDataManager Sub DLL not found
2502	Function missing
2503	'Read Thread' not created
2504	'Write Thread' not created
2505	'IO Thread' not created
2510	Function failed
2512	Assign reports error. Return neither OK or cancel

Table of the Error Nubrs of the Driver Functions of the Online Data Manager

8.5.4 Online Data Manager Subfunctions Error Numbers (8001 .. 8035)

The following table lists the error numbers of the Subfunctions of the Online Data Manager.

Error Number	Description
8001	Driver not opened. E.g. CIF Devive Driver
8002	Application has requested an unknown event
8003	Application has requested an unknown command
8004	Command has failed
8005	Command active
8006	Device invalid
8010	No device was assigned
8011	Device was already assigned
8020	Driver not connected
8021	Driver already connected
8030	Faulty 'GetState'
8031	Send error (PutMessage returns error)
8032	Send active (PutMessage active)
8033	Receive error (GetMessage returns error)
8034	Receive active (GetMessage active)
8035	IO Error (ExchangeIO returns error)

Table of the Subfunctions of the Driver Functions of the Online Data Manager

8.6 Data Base Functions Error Numbers (4000 .. 4098)

The following table lists the error numbers of the converting functions.

Error number	Description
4000	File does not exist
4001	Succes in comprimizing
4002	Dataset does not exist
4003	Last respectively first entry reached
4004	Not enough memory
4005	File directory full
4006	Max number of entries reached
4007	No writing to this table possible, because the table is located in the FLASH
4008	Table name does already exist
4009	File name does not exist
4010	Free RAM length from RCS_CNF.P86 is smaller than E_F_INDEX * 2
4011	Parameter ‘next’ wrong
4012	Not enough free space to copy data set
4013	Set is deleted
4014	Value for Index is wrong
4015	Access not allowed
4016	open_file used before init_file
4017	Drive is not ready
4018	Not enough drive memory
4019	File name or path does not exist
4020	Cannot create path
4021	Wrong path
4022	Wrong flag
4023	The delete path is the root path
4024	Path file exists
4025	Write error during write a file
4026	Error during create a file
4027	Error during close a file
4028	No DBM file
4029	Length of the read data is unequal of the file length

Table of error numbers of converting functions

Error number	Description
4030	Path too long
4031	Directory changed
4032	Directory created
4034	Length of converting stream is 0
4035	Non equal data set found
4036	Non equal data set found
4037	Non equal data set found
4038	Data set has length 0
4039	The function DbmInit has assigned a Zero pointer during RCS initialisation
4040	Printer not ready
4041	The data base is used from an other function
4042	New length of data base is smaller than used
4043	Unknown access mode
4044	Old data base has to be converted
4045	Error while converting. Function not known
4046	Unknown type in set 0 found
4047	No float function available
4048	Function not in RCS module
4049	Check failed
4050	Checksum ckeck failed
4051	More segments are existing in file, than in the structure FILE_INFO_T in wMaxEintraege
4052	SegLen in structure FILE_INFO_T is smaller then the length in the file. Return of function dbm_restore_data
4053	The header file holds an other information for a length than in the segment itself
4054	Not enough memory for allocation on the PC
4055	No index for file handle in structure FLASH_DIR of RCS found
4056	
4057	File type 2 can not be printed because of too many definitions
4058	The definitions need too many lines to display them, than in the program available
4059	An unknown format for the parameter. Valid is U, H, or S
4060	Unknown parameter type

Table of error numbers of converting functions

Error number	Description
4061	The data base was transmitted into the FLASH
4062	Set 0 contains no structure definition
4063	Set 0 can not be deleted
4064	Error during execution of a ODBC data base access
4065	Initialising of DBM through RCS had no success
4066	Passed data length incorrect
4067	Sorting function not linked
4068	Error in function parameter
4069	Error from ODBC table
4070	No free handle available. Too many data base links are already opened
4071	Unknown data type found in the table
4072	Structure of table GLOBAL not correct or no such table existing
4073	No name of an ACCESS data base
4074	Download window can't be created
4075	Download not fully performable

Table of error numbers of converting functions

Error number	Description
4082	More than 32 tables should be created
4083	No entry in element szSourceFile
4084	ODBC connection initialisation not possible. This could happen when in file ODBCINST.INI in section [Microsoft Access Driver (*.mdb)] is no valid path to ODBCJT16/32.DLL.
4085	Error in structure in the ACCESS data base that is in DBM format
4086	Error in structure in the ACCESS data base that is in DBM format
4087	No data in a ODBC table
4088	No entry
4089	ODBC set length not valid
4090	Not enough data sets in ODBC table
4091	Table CreateTab not found
4092	Error in structure of table CreateTab
4093	No entry in element szSourceTable
4094	No entry in element szDestTable
4095	Entry in iSourceType of table CreateTab is wrong
4096	Entry in iTranslate of table CreateTab is wrong
4097	Function SQLAllocStmt reports an error
4098	ODBC source table not found
4099	ODBC data truncated
4100	Download timeout
4101	Library load error
4102	Library function error
4103	Error in description 'toggle'
4104	Error in description 'KB'
4105	Column does not exsist
4106	ODBC structure different
4107	ODBC address error
4108	No CRC sum exists (table GLOBAL exists or old)
4109	Table GLOBAL is old
4110	Calculated CRC different to CRC in table GLOBAL
4199	Programming error

Table of error numbers of converting functions

8.7 Converting Functions Error Numbers (5001 .. 5008)

The following table lists the error numbers of converting functions.

Error number	Description
5000	Function PackLongToByteShort: Not enough space in pvD (Number of elements greather than reserved memory)
5001	Function PackLongToByteShort: Not enough space in pvD. Detected during converting of pvS
5002	Function PackLongToByteShort: Not enough space in pvD
5003	Function StringToByte: Not enough space in pvD
5004	Function IntToByte: Not enough space in pvD
5005	Function LongToShort: Not enough space in pvD
5006	Function PackStringDumpToByteArray: Not enough space in pvD
5007	Function PackStringBumpToByteArray: A character was found, which is not convertable into a HEX value
5008	Function PackStringDumpToByteArray: Number of character odd
5009	Funktion PackStringDumpToByteArray: Not enough space in pvD
5010	Funktion PackStringDumpToByteArray: The current data set needs to be appended the previous one
5011	Funktion PackStringDumpToByteArray: No corresponding function to the given number exist
5012	Converting error

Table of error numbers of data base functions

9 Appendix

9.1 OPC Fundamentals

OPC (**O**LE for **P**rocess **C**ontrol) describes the utilization of the OLE technology for automation applications. OLE, based on the COM/DCOM functionality of the Microsoft Windows operating system, is the Windows-specific method of communication between different program parts. OLE/(D)COM

9.1.1 OLE/(D)COM

OLE/(D)COM describes the communication always between two partners:

- One client that requires a functionality,
- and one server that makes this functionality available.

A server makes this functionality available in the profile of objects.

In OLE/(D)COM, objects of a server are described by means of their object interfaces (collection of functions, abstract base classes of the objets). An object can possess several interfaces. The client obtains access to the functions of the object interfaces locally. In the server, the Interface functions must be implemented in reality. client and server partners of an OLE/(D)COM linkage can be parts of a process or different processes on a computer (OLE/COM), or different processes on different computers (OLE/DCOM).

The OLE/DCOM functionality for communication between processes on different computers is a part of the operating system of Windows NT 4.0 and is obtainable as an addition to Windows 95.

In the communication between different processes it is necessary to code the call up parameters and results of the functions called up, in a unified format, to transmit them to the partner process and to then to decode them suitably there. The coding/decoding processes are undertaken here by the Proxy or Stub functions which can be automatically generated from the interface definition (in Microsoft-IDL) with the aid of the MIDL tool. Thus, every interface function is a Proxy function (as local representation of the function for the calling-up client process) and is allocated to a Stub function (call up of the function implementation in the server process).

In order to make the information on accessible objects and interfaces as well as implemented Proxy or Stub functions to a possible client and also an operating system possible, it is necessary to make the corresponding entries in the Windows registration data base (registry).

9.1.2 OPC Overview

OPC describes the application of the OLE/(D)COM functionality for automation tasks. For this purpose, OPC-specific Interfaces are defined with their associated functions, as well as objects which contain these Interfaces.

The OPC specification was created by the OPC Task Force and is available as Version 1.0A (30th July 1997). It contains the specification of the Interfaces in IDL (directly translatable with MIDL) as well as the required OPC definitions required for the automation interface (see below).

9.1.3 OPC Server Model

As described, the OPC specifications contain the definitions of the OLE/(D)COM Interfaces and the functions summarized therein. Thus, this is a pure interface specification of the interface between an OPC SERVER without defining how the interface functions in a concrete Server realization are to be implemented.

In order to understand the semantics of the OPC Interface, it is necessary to understand the model of an OPC SERVER to be used. The OPC server model describes the unique depiction of an automation component from the point of view of an OPC client, independently of the actual realization (hardware, programming language, program architecture, etc.).

The OPC specification is based on the following model of a server.

A server is comprised of a number of Items (= variable objects) which, on their side, depict the linkage to real data objects. Each Item has a unique characteristic inside the OPC server. The client has the option of searching for a particular Item within the access path (similar to the directory tree). An Item is allocated a handle which, in a concrete context, can be accessed faster.

The Items can be laid down in various ways (pre-defined, by means of a configurator, on the basis of the characteristic syntax, etc.). The laying down, deletion and management of Items is no part of the OPC specification.

A client usually requires only a small quantity of the Items which a server makes available. In order to be able to access these Items, it arranges them into a group that it has created. A client form one or more groups allocated to it in a server.

Besides the groups fixedly allocated to a client, it is also possible to create generally accessible (public) groups. These groups are created either by a client and then declared as public or they are built up by the server itself without the cooperation of a client.

When inserting Items into a group, the type of data, which is to be allocated to this Item for communication with the client, can be given. If possible, during the access to the Item, the original values are converted to this type of data, if not, then a default type is used in place of the per-defined type. Permissible types of data are the values of the "VARIANT" types defined for OLE automation. It comprises the common basic types and arrays. The value is always transmitted with a time stamp and a quality characteristic.

In Items contained in a group it is possible to read and write synchronously as well as asynchronously, whereby with the call up of a read/write function, the values of several Items can also be read and written.

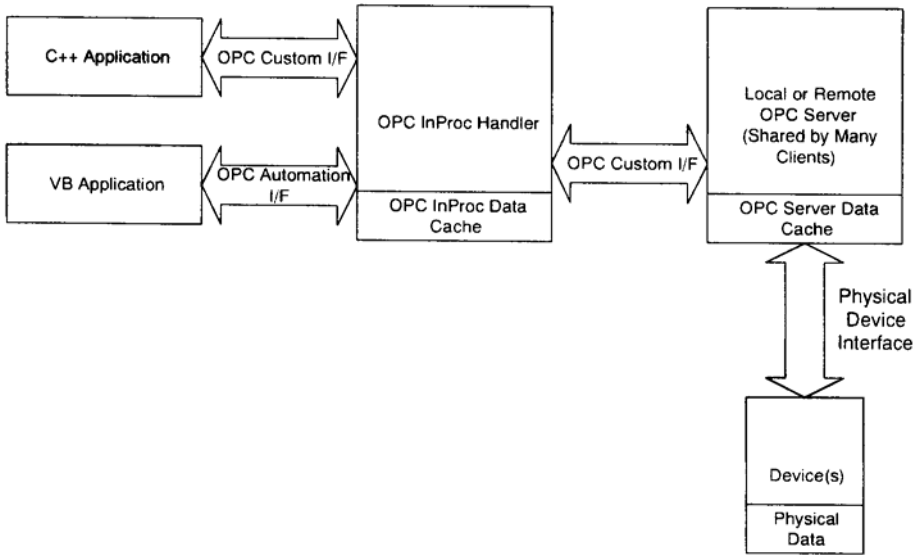
Synchronous reading means that the values of all Items that can be read with the call up of the read function have possibly been created at the same time (the time differences are dependent on the concrete server implementation) and the read function is only ended after the complete processing. Contrary to this, asynchronous reading means that with the function call-up of the reading procedure, the reading function is only started and after that, the read values are transmitted cyclically to the client until the transmission is explicitly closed by the client. For analogue values, there is the possibility of providing a value transmission only after exceeding a given percentage limit. The transmission of values of individual Items in a cyclical procedure can furthermore be switched On or Off by activating or deactivating a group.

Synchronous and asynchronous writing differs only in the type of confirmation (synchronous results are returned by the writing function, asynchronous results are only returned later).

9.1.4 OPC Interface Types

OPC defines 2 types of Interfaces (actually Interface collections) by means of which an OPC client can access an OPC SERVER:

- The "Custom Interface" contains the interfaces for the direct call up of the server functions by, e.g. a C++ Client (early linkage) and
- The interfaces of the "Automation-Interface" make possible the indirect call up of the server functions by means of OLE automation (late linkage), e.g. for Visual Basic- or Excel Clients.



Picture 1: OPC architecture (source: OPC specification)

As can be seen from the architectural proposal of the OPC Task Force in Picture 1, the actual server implementation is accessed via the Custom Interface. The automation interface provides a client in a different way the same Server functionality as the custom interface. If required, the automation interface, in an OPC handler in the processing environment of the client, can later be depicted with the aid of functions of the custom interface. For the reasons stated, the object and interface definitions of the custom interface will be dealt with exclusively in the following.

9.1.5 OPC Objects

Individually, OPC defines the following objects and interfaces which must be implemented for every server that conforms to OPC.

OPC Server

Contains the administration of the OPC SERVER and the interfaces for managing of (public and client specific) groups:

- **IOPCServer**

  Group management

- **IOPCServerPublicGroups (optional):**

  Group management

- **IOPCBrowseServerAddressSpace (optional):**

  Browser for Server Items

- **IPersistFile (optional):**

  Loading and saving of the current Server contents

OPCGroup

A group of Items. Items can be added and deleted. An automatic update can be selected for individual or all Items. The asynchronous transmission of values is carried out via the data interface, a callback function in the IDataSink interface of the client is called up.

- **IOPCGroupStateMgt**

  Group Status management.

- **IOPCPublicGroupStateMgt (optional):**

  Group Status management.

- **IOPCSyncIO, IOPCAsyncIO:**

  Reading and writing Item values

- **IOPCItemMgt:**

  Inserting and deleting Items

- **IDataObject:**

  Asynchronous value transmission

- **IEnumOPCItemAttributes:**

  Interface for iterating the items of the group

9.2 Test Systems

All client application listed here are available in our company and are used to test our OPC server.

Here is the list of the test systems with details (name and version).

- **CIMPLICITY GE Fanuc Version 3.22 (Build 514)**
- **FactorySoft OPC Inspector V1.1**
- **GEFASOFT GraphPic NT 5.0**
- **ICONICS Genesis-32 GraphWorX32 V 5.00.23 (Build 5.2)**
- **Ing. Büro Dr.Ing. Schoop WinErs V 3.1**
- **Intellution FIX32 OPC Power tool v7.01 also used by FIX32 dynamics**
- **PCSoft WICZON V 7.5**
- **Siemens WinCC OPC Tag Manager V4.0, WinCC V4.01**
- **Wonderware FactorySuite 2000 OPC Link V7.0.1.1, WWClient V7.0**

9.3 OPC-Implementation

The implementation of the OPC functionality is carried out in accordance with the OPC specification:

1.: OLE for Process Control

Data Access Standard, Version1.0A, July 30, 1997

The following interfaces are implemented:

OPCServer

- **IOPCServer** **(mandatory)**
- **IOPCBrowseServerAddressSpace** **(optional)**

OPCGroup

- **IOPCGroupStateMgt** **(mandatory)**
- **IOPCSyncIO,** **IOPCAsyncIO** **(mandatory)**
- **IOPCItemMgt** **(mandatory)**
- **IDataObject** **(mandatory)**
- **IEnumOPCItemAttributes** **(mandatory)**

and 2.: OLE for Process Control

Data Access Standard, Version2.0

The following interfaces are implemented:

- **IOPCAsyncIO2**
- **IconnectionPointContainerImpl**
- **IconnectionPointImpl**

9.4 Technical Data and Hints

- The OPC server bases on the specification version 1.0A and also on the specification version 2.0 from version 2.6XX (sycon.exe)
- The OPC server is compiled with C++ Compiler version 6.0
- The OPC Server name is: HilscherGmbH.CifOpcServer
- The OPC server is available for following fieldbussystems:
  - PROFIBUS-DP Master/Slave
  - CANopen Master/Slave
  - Interbus Master/Slave
  - DeviceNet Master/Slave
- At the moment it is only possible to construct or load *.pdd (predefined devices) files at PROFIBUS-DP
- Only CANopen PDO´s but no CANopen SDO´s are supported by the OPC Server
- No InterBus PCP is supported by the OPC Server
- No DPV1 service is supported by the OPC Server
- No PROFIBUS-FMS is supported by the OPC Server
- Deadband is supported not yet
- Maximum length of symbols: 32 symbols
- Maximum length for the description: 64 symbols
- Permissible symbols: a-z, A-Z, numbers and underline (_)
- Values for a client can be simulated in a simulation mode, although that no hardware is existing

Important Note: The OPC server supports only cyclic communication: PROFIBUS-DPV0, InterBus, CANopen PDOs respectively DeviceNet Implicit I/O. The OPC server does **not** support acyclic communication like PROFIBUS-FMS, PROFIBUS DPV1, InterBus PCP, CANopen SDO up-/download respectively DeviceNet explicit messaging.

10 Contact

Headquarter

Germany
Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China
Hilscher Ges.f.Systemaut. mbH
Shanghai Representative Office
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support
Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France
Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support
Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

Italy
Hilscher Italia srl
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support
Phone: +39 / 02 25007068
E-Mail: it.support@hilscher.com

Japan
Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support
Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support
Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA
Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support
Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com