

Scalable PLC  
for Individual Automation

OPC Server  
OLE for Process Control

# OPC Server

OPC

OLE

COM

DCOM

TCP/IP

Ethernet



# Contents OPC Server

<b>1 Overview of OPC</b> .....	3
<b>1.1 General information of the OPC server</b> .....	3
<b>1.2 Fields of application of the OPC server</b> .....	5
<b>1.3 What has to be taken into account?</b> .....	6
1.3.1 General.....	6
1.3.2 System requirements.....	6
1.3.3 Preparations of the CoDeSys project.....	6
1.3.4 Configuration of the OPC server.....	7
1.3.5 Project planning of the client.....	8
<b>2 Installation of the CoDeSys OPC server</b> .....	9
<b>2.1 Required files</b> .....	9
2.1.1 Gateway files.....	9
2.1.2 OPC server files.....	10
2.1.3 Other files.....	10
<b>2.2 Installation and registration</b> .....	10
2.2.1 Installation of Gateway and OPC server.....	10
2.2.2 OPC server, Installation and registration.....	10
2.2.3 Register the OPC server (update).....	11
2.2.4 Deinstall the OPC server.....	11
<b>3 Use of the CoDeSys OPC server</b> .....	12
<b>3.1 General notes</b> .....	12
<b>3.2 Settings in the programming software CoDeSys</b> .....	12
3.2.1 Step 1: Creating the data objects.....	12
3.2.2 Step 2: Configuration of the symbol files.....	12
3.2.3 Step 3: Setting of the communication parameters in the project.....	14
3.2.4 Step 4, optional: Store project.....	14
3.2.5 Step 5: Creating of the symbol file and transmission to the gateway.....	14
<b>3.3 Configuration of the OPC server with OPCconfig.exe</b> .....	15
3.3.1 Menu <File>.....	16

3.3.2 Menu <Edit> .....	17
3.3.3 Single-PLC configuration .....	18
3.3.4 Multi-PLC configuration .....	23
3.3.5 Registry entries (example).....	24
<b>3.4 Start of CoDeSysOPC.exe .....</b>	<b>25</b>
<b>3.5 End of CoDeSysOPC.exe .....</b>	<b>25</b>
<b>3.6 Starting and ending of Gateway.exe.....</b>	<b>25</b>
<b>4 Behaviour of the OPC server.....</b>	<b>26</b>
4.1 General recommendations .....	26
4.2 Time response of the OPC server .....	26
<b>5 Connection to an OPC server on another PC.....</b>	<b>27</b>
5.1 DCOMCNFG.EXE .....	27
<b>6 Example of an ini file for the OPC server configuration.....</b>	<b>27</b>
6.1 Example of a multi-PLC configuration .....	27
<b>7 Short checklist .....</b>	<b>29</b>
7.1 Short checklist.....	29
<b>8 Index .....</b>	<b>30</b>
<b>9 Glossary .....</b>	<b>31</b>

# 1 Overview of OPC

## 1.1 General information of the OPC server

OPC is a standardized interface for accessing process data. It is based on the Microsoft COM/DCOM standard and has been expanded according to the requirements when accessing data in the field of automation. Here, it is primarily used to read/write values from/to the controller. Typically, OPC clients are visualizations or programs for the acquisition of operating data, etc. OPC servers are usually provided for PLC systems and field bus cards.

The OPC server is not a passive subprogram library, but an executable program which is started when the connection between client and server is established. This is why it is able to notify the client when the value or status of a variable has changed.

Due the characteristics of DCOM, it is even possible to access an OPC server which is running on another computer. Furthermore, a data source can be simultaneously accessed by more than one client via OPC. Another advantage OPC gains by the usage of COM is that different programming languages (C++, Visual Basic, Delphi, Java) can be used. However, a resulting disadvantage is the considerably higher usage of resources (memory and CPU time).



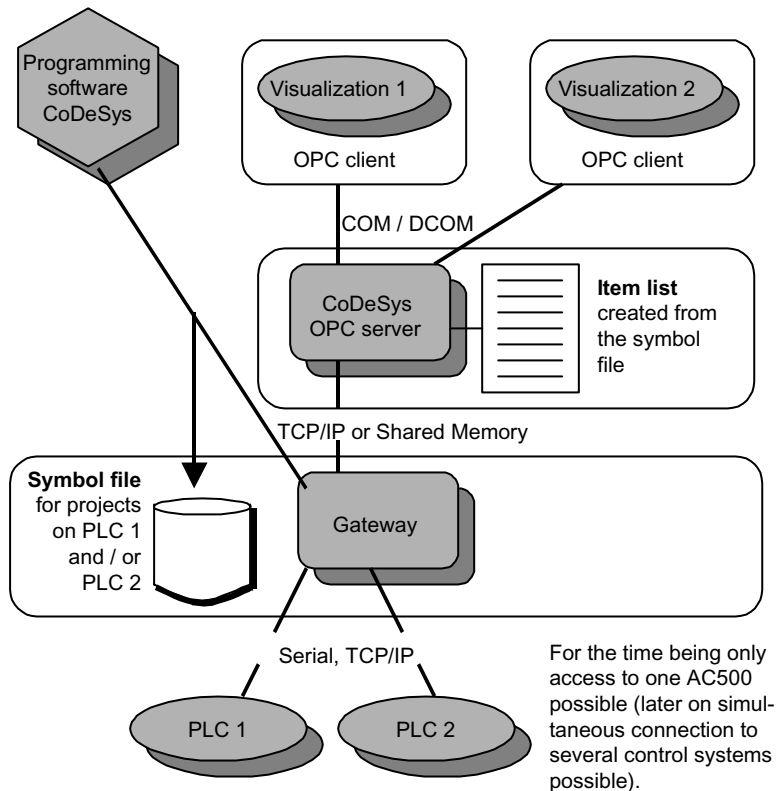
**Note:** CoDeSys OPC server V2.3.9.0 is able to communicate with the controllers listed in section 1.2 Fields of application of the OPC server. It fulfills the requirements of OPC standard V2.3.9.0.

- 
- 1) OPC = OLE for Process Control  
OLE = Object Linking and Embedding

For further information see also <http://www.opcfoundation.org> and <http://www.opceurope.org>

- 2) COM = Component Object Model (Basis für OLE)  
DCOM = Distributed Component Object Model

## Architecture of the CoDeSys OPC server V2.x



The CoDeSys OPC server uses the CoDeSys gateway server as a gateway.



**Caution:** The symbol file is the basis for the data exchange via the CoDeSys OPC server.

When a project is loaded from the CoDeSys programming system to the controller, it is possible to simultaneously create a **symbol file** (\*.sym or \*.sdb) and store it in the gateway. The symbol file contains so-called items. An item (data object) exactly corresponds to one variable in the controller program. Using these items the variable values on the controller can be called.

The OPC server requests the content of the symbol file from the gateway and creates an **item list** from it. Since the content of the item list is determined by the variables available in the controller, it cannot be influenced by the OPC client. The OPC server reads the symbol file last loaded for a project via the gateway channel.

The item list in the OPC server is updated in definable intervals with the values from the controller(s). Compared to the direct access to the controller, reading and writing the variable values via this cache list has the advantage of fast access times (max. approx. 1 ms per item).

A figure drawn from past experience for the amount which can be handled by the OPC without problems when taking over variables into the item list, is approx. 15 000 items, symbol files size approx. 1.5 MB. The utilization for reading and writing the variable values naturally depends on the number of items which are set to 'active' and therefore are to be considered when updating the values.

The OPC server supports grouping of data. Here, it is distinguished between "public groups" which are allocated by the OPC server and "private groups" which can be composed by a client.

If the corresponding option is activated in the configuration, the OPC server groups the items (i.e. the variables of a project) block-wise. Then, one 'public group' is created per block.

'Private groups' can be combined of individual items as desired in the client. First, they do not affect the grouping in the OPC server but they can be made to 'public groups', if required. For example, private groups are suitable to activate or deactivate specific variable groups with only one command depending on whether they are to be accessed or not.

Grouped data should be read by the OPC server in a consistent way, i.e. all variables at the same time. Please note that this is not always possible for target systems with limited communication buffers!



**Note:** It is possible to access an OPC server which is running on another computer in the network even if an OPC server is also running on the local computer.

### What's new compared to CoDeSys OPC server V2.0:



**Caution:** For the time being, multi PLC operation with several controllers (i.e. connection of the client to several controllers) is not allowed. At the beginning, only the use with one controller will be possible.

- The item list is also generated during offline operation, i.e. when no communication can be established with the corresponding subscriber. Then, the item list is loaded from the symbol file. Thus, the visualization can be designed in the office without having the hardware available and then be taken into operation on-site in the system. However, if a wrong (old) project is accidentally stored in the selected PLC, no variables can be provided.
- Furthermore, individual variables (items) can be added to the symbol file. Therefore, it is not mandatory to insert the variables of an entire block.

## 1.2 Fields of application of the OPC server

The OPC server represents the connection between client (e.g. visualization software) and controller. The data (items) are read from the controller via the gateway server. The OPC server makes all read items available for the clients. The client software (visualization) displays the required items.

The following AC500 controllers can be operated via the OPC server using the corresponding drivers:

- AC500-Mini serial
- AC500-Mini Ethernet



**Note:** Please refer also to the documentation CoDeSys / Volume 7 / Chapter 10 " Programming and Test".



**Note:** Only the listed controllers and drivers can be used.

## 1.3 What has to be taken into account?

### 1.3.1 General

#### Available drivers:

- **Serial (RS232)** CoDeSys standard driver for serial communication (3S serial RS232 driver)
- **TCP/IP** CoDeSys standard driver for Ethernet communication (3S Tcp/Ip level 2 driver)

#### New function:

The item list is also generated during offline operation, i.e. when no communication can be established with the corresponding subscriber. Then, the item list is loaded from the symbol file. Thus, the visualization can be designed in the office without having the hardware available and then be taken into operation on-site in the system.

However, if a wrong (old) project is accidentally stored in the selected PLC, no variables can be provided.

Furthermore, individual variables (items) can be added to the symbol file as of CoDeSys Version 2.3.9.0. Therefore, it is not mandatory to insert the variables of an entire block.

#### Peculiarities:

The first version of the CoDeSys OPC server for AC500 only allows communication with one subscriber. The operation mode Single-PLC can be selected (see chapter 3.3.3 Single-PLC configuration). For the present, the operating mode Multi-PLC can be used with only one PLC (see chapter 3.3.4 Multi-PLC configuration).

### 1.3.2 System requirements

When using the OPC server, the used PC plays a major role. Particularly for extensive configurations (multiple users, many items (variables)) a fast PC should be used to ensure fast communication and stable function.

The PC should fulfill the following minimum requirements:

- Pentium IV
- Clock frequency 500 MHz
- 128 MB RAM memory
- Operating system WIN 2000 with at least service pack 4 (or higher) , WIN XP with at least service pack 1 (or higher)



**Note:** The better the system, the faster and more stable the communication with the OPC server. Particularly for extensive configurations with multiple users and many items, a fast PC should be used. Fast systems also guarantee that the transmission times do not fall below the values listed in section 4.2 "Time response of the OPC server".

### 1.3.3 Preparations of the CoDeSys project

First, the OPC variables (items) have to be defined in the CoDeSys project. For this, either all variables of the entire project can be defined or the variables of individual program parts (e.g. programs, defined functions and function blocks) can be enabled. Furthermore, it is also possible to select individual variables for the OPC server.



**Note:** Particularly for extensive systems the selection of the variables to be enabled is important. The more variables are enabled the higher is the system load and the slower becomes the communication. In order to keep the system load as low as possible, only those variables (items) should be enabled that are actually used in the visualization software (client). The corresponding variables must be grouped during the project planning (e.g. Var\_Global: visualization or in individual subgroups, functions or function blocks).

Further information about defining OPC variables can be found in chapter 3.2 "Settings in the programming software".

During this process, a file named <projectname>.sym is created which contains all enabled OPC variables (items). Now, this project must be sent via the current gateway to the PLC and stored there in the **Flash** memory.



**Note:** This procedure is important as the OPC server compares the project settings in the gateway with the settings in the PLC. In case of any differences, the items are not displayed or updated in the OPC server.



**Note:** When preparing a project in the office without having the corresponding hardware available, the symbol file must be copied to the system path \Gateway Files after finishing the CoDeSys project. This means that the file <project name>.sdb must, for example, be copied to the directory C:\WINNT\Gateway Files.

### 1.3.4 Configuration of the OPC server

In the OPC configuration (OPCConfig.exe) all subscribers are defined from which items shall be read. Here, it is important that the program name matches the set gateway driver (e.g. node address). This means that the RAM of the PLC to be accessed via the set gateway channel must contain the same program as set with Program name. Otherwise, no items can be read.

In the OPC configuration also the transmission rate for the items is set. For each subscriber an individual timeout can be set.



**Caution:** The following settings must be observed for the AC500 controllers. Otherwise, no communication is possible.

- Buffer size = 4800
- No Login Service = set
- Motorola byteorder = set



**Note:** Further information can be found in the OPC documentation, chapter 3.3 "Configuring the OPC server using OPCconfig.exe".



**Note:** If the project name and the gateway driver do not match, no items are available for the client. If the transmission rate and the timeout settings are not correct, the items cannot be updated. The status BAD is displayed.

### 1.3.5 Project planning of the client

For the client, either a client test software (for testing the availability of items) or a visualization software supporting OPC can be used.

The OPC server is automatically started when the client software is started and the communication is established. The corresponding variables are selected from the item list.

When planning the client (visualization software) the communication must be optimized. For this, the items are divided in individual groups. These groups are defined in a way that they only contain items that have to be updated at the same time. The groups are only activated when they are needed.

Example:

- Group 1:	all error and malfunction messages	* always active
- Group 2:	Measurement data (e.g. diagram)	* always active
- Group 3:	Variables screen 1	* only active if screen 1 is displayed
- Group 4:	Variables screen 2	* only active if screen 2 is displayed
- ...		
- Group n:	Variables screen n	* only active if screen n is displayed

## 2 Installation of the CoDeSys OPC server

### 2.1 Required files

#### 2.1.1 Gateway files

The gateway files are installed automatically with the programming software CoDeSys. All corresponding files are located in the system directory WINNT\system32:

- Commsym.dll                      Communication DLL
- Commusr.dll                     Communication DLL
- Gateway.exe                     Gateway for communication protocol
- GatewayDDE.dll                 DLL
- Gclient.dll                      Communication DLL
- GDrvBase.dll                    3S standard driver
- GDrvStd.dll                     Communication DLL
- Ghandle.dll                     Communication DLL
- Gsymbol.dll                     Communication DLL
- Gutil.dll                        Communication DLL

The following drivers (for AC500 and AC31 Series 90) are in preparation and will be released with CoDeSys V2.x:

- ArcnetX.dll                      Communication DLL
- GDrvABBRS232RouteX.dll        ABB RS232(routed) driver DLL
- GDrvABBTcpIpL2X.dll            ABB Ethernet driver
- GDrvArcAC.dll                    ABB ARCNET driver

The following drivers cannot be used with AC500. These drivers are only intended for AC31 Series 90 controllers.

- Arcnet32.dll                     Communication DLL
- Arcnetn.dll                     Communication DLL
- GDrvArcnet3f4f.dll              ABB ARCNET driver
- GDrvArcnetRouteFast.dll        ABB ARCNET driver
- GDrvABBArcnet.dll                ABB ARCNET driver
- GDrvABBModemRoute.dll         ABB modem driver
- GDrvABBRS232.dll                ABB serial driver
- GDrvABBRS232Route.dll         ABB serial driver
- GDrvABBTcpIpL2.dll              ABB Ethernet driver

OPC server V2.3.9.0 requires the gateway from the CoDeSys CD-ROM V1.0 or higher!

The following drivers are recommended for creating new projects:

Serial: - Serial (RS-232)    3S standard driver

Ethernet: - TCP/IP            3S standard driver

After the first start of the gateway the path for the 'Gateway Files' directory is set in the registry (default: C:\WINNT\Gateway Files). When the connection to the controller is established, the symbol files created by CoDeSys and stored in the project directory are copied to this directory. These files are either the symbol files \*.sym or their binary version \*.sdb. The latter can be read faster by the OPC server.



**Note:** When preparing a project in the office without having the corresponding hardware available, the symbol file must be copied to the system path \Gateway Files after finishing the CoDeSys project. This means that the file <project name>.sdb must, for example, be copied to the directory C:\WINNT\Gateway Files.

### 2.1.2 OPC server files

The OPC server has to be installed using the installation menu of the CoDeSys CD-ROM. The files can be saved to any directory. The DLLs and the file OPCenum.exe, however, must be kept in a subdirectory named REDIST:

- CoDeSysOPC.EXE - OPC server
- DiagnosticOPCClient.exe - Client test software
- OPCCCommonSetup.EXE - Setup for the files listed below
- OPCCConfig.exe - Configuration of the OPC server
- OPCCConfig\_e.exe - Configuration of the OPC server

Subdirectory REDIST (standard OPC files):

- CALLRPROXY.DLL
- OCSDAAuto.DLL
- OCSSpy\_PS.DLL
- OPCCOMN\_PS.DLL
- OPCenum.EXE
- OPCPROXY.DLL

### 2.1.3 Other files

DCOMCNFG.EXE (C:\WinNT\System32):

This file is used to establish a connection to an OPC server which is installed on another computer (refer to OPC documentation, chapter 4 "Behaviour of the OPC server").

## 2.2 Installation and registration

### 2.2.1 Installation of Gateway and OPC server

The gateway is automatically installed and registered together with the programming software CoDeSys. The OPC server is installed and registered by clicking the button "Installation OPC Server" in the installation menu of the CoDeSys CD-ROM. Follow the instructions given in the setup.

### 2.2.2 OPC server, Installation and registration

If you are installing the OPC server afterwards, make sure that the correct gateway server is used (from CoDeSys CD-ROM, version V1.0 or higher). On the computer, a separate directory path is created for the program files of the OPC server (e.g. C:\Program Files\3S-Software\CoDeSys\OPC). All OPC server files (including the subdirectory REDIST) must be stored to this path.

Using the command

"CoDeSysOPC /Install&rdquo; (if necessary, the path to the exe file must be entered)

in the 'Run' dialog box, a setup program is started which executes the installation of the OPC files. Follow the instructions given on the screen. Then, the OPC server performs an automatic registration. A message is displayed informing about the success of the registration process.

### 2.2.3 Register the OPC server (update)

The possibility of a separate registration is important for program updates or the creation of a setup. All files of the OPC server update are copied to the present program path.

Using the command

"CoDeSysOPC /RegServer" (if necessary, the path to the exe file must be entered)

only the registration of the OPC server is initiated. The installation of the required files must have been finished before. If no message is displayed, registration was carried out successfully.

(Further notes can be found in chapter 3.3.5 "Registry entries (example)".)

### 2.2.4 Deinstall the OPC server

The registration of the OPC server is deleted using the following command:

"CoDeSysOPC /UnRegServer" or "CoDeSysOPC /DeInstall".

This deletes the entries in the registry. However, the installed files are not deleted by this command!

## 3 Use of the CoDeSys OPC server

### 3.1 General notes

After the OPC server is installed, it should be provided by the OPC client (e.g. visualization) for selection.

The gateway and the OPC server are automatically started by the operating system as soon as one of the clients (visualization software) establishes a connection. The OPC server is automatically exited as soon as all clients have cleared the connection. The gateway remains open, but it is inactive.

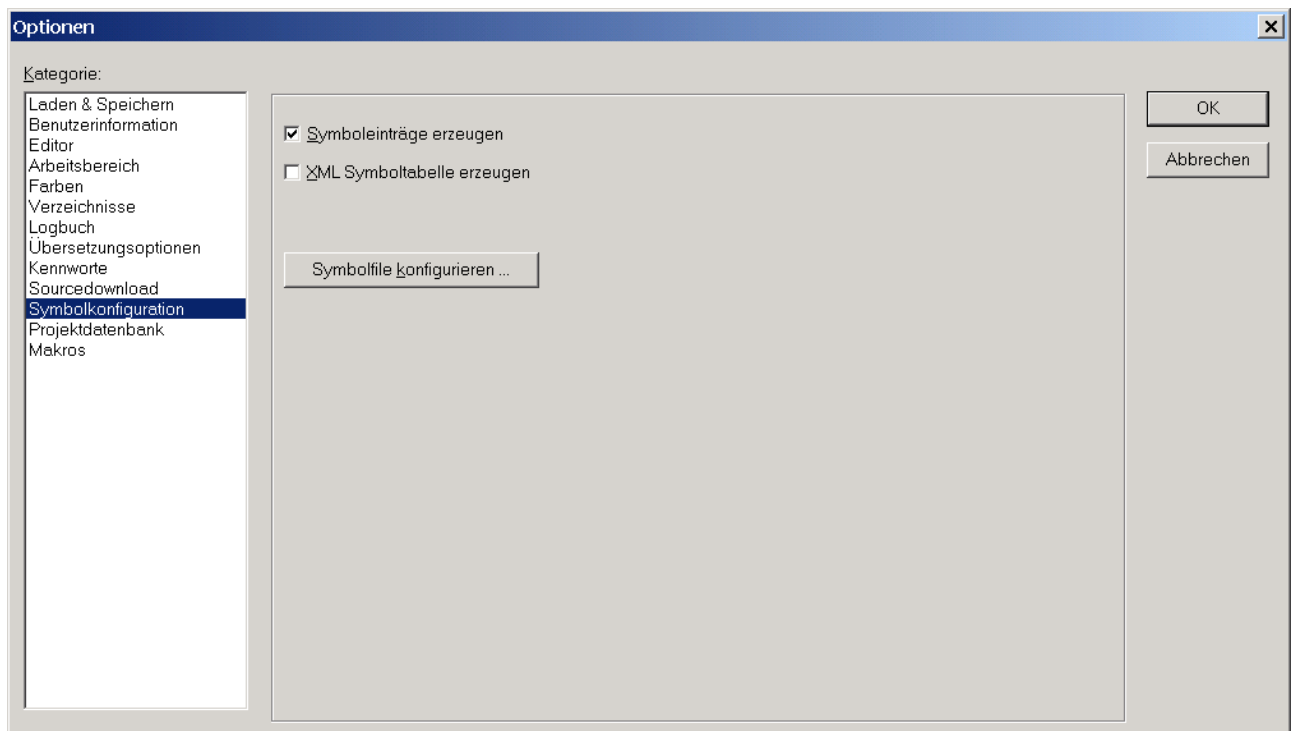
The steps 3.2.1 to 3.2.3 described below must be carried out to make data objects of a project in the CoDeSys programming system available for the OPC server and to establish a connection via OPC.

### 3.2 Settings in the programming software CoDeSys

In order to enable the OPC server to access data objects of a project, first a symbol file has to be created. For this, start the programming software CoDeSys and open the project.

#### 3.2.1 Step 1: Creating the data objects

The gateway does not use a project file, but a symbol file of the same name with the file name extension "sym" or "sdb" (binary). This file contains symbol entries (item list) for the project variables. To automatically create this symbol file with each compilation process, the option 'Dump symbol entries' must be selected. The corresponding settings can be made by calling the menu item <Project><Options>. The automatic creation of the symbol file is set by selecting the category <Symbol configuration> and then checking the checkbox 'Dump symbol entries'. The dialog stays open for entering further settings.

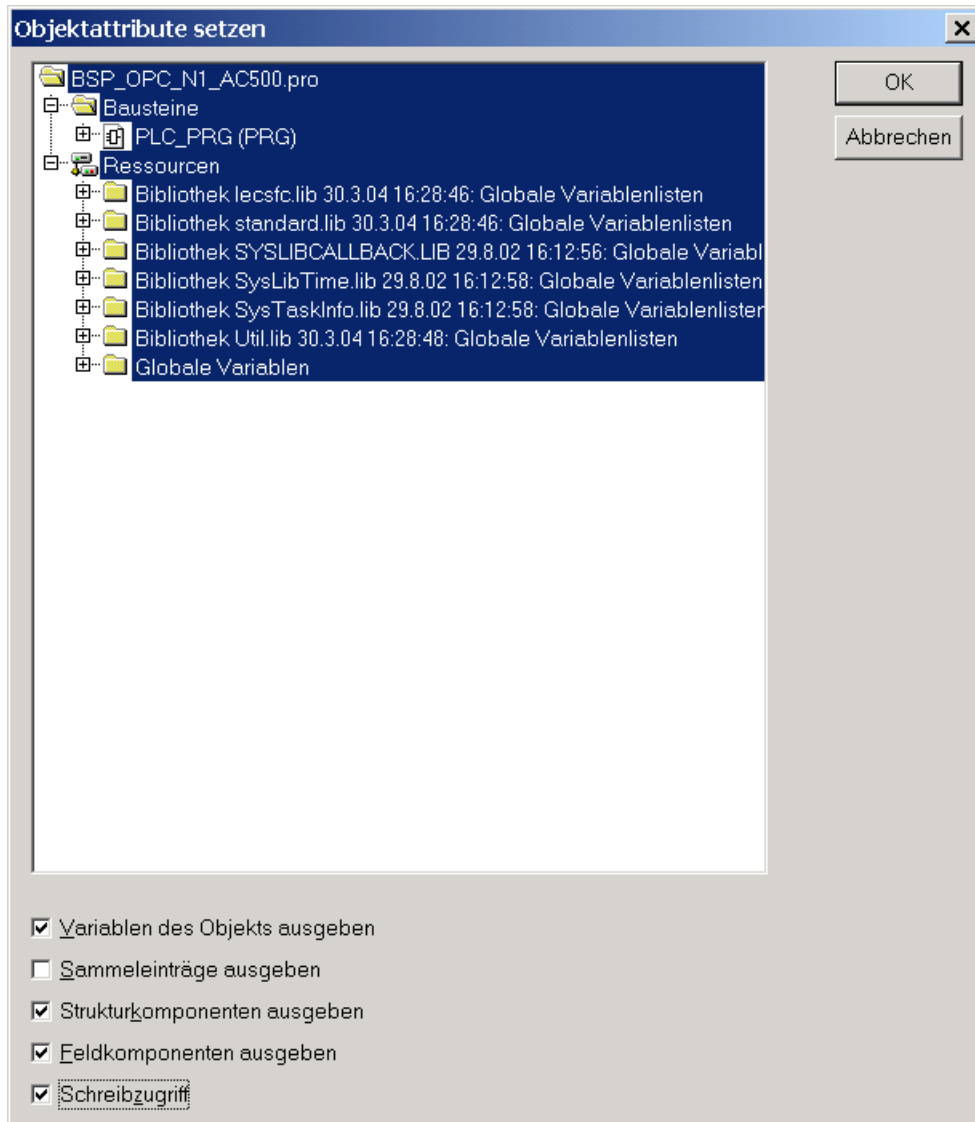


#### 3.2.2 Step 2: Configuration of the symbol files

In the project structure tree shown in the dialog 'Set object attributes', the POUs or individual variables (items) can be selected for which symbol options shall be set. For this, click the button <Configure symbol file> in the presently open dialog or select <Project> <Options> from the menu and then <Symbol configuration> <Configure symbol file>.

## Configuring all POUs

If all project variables shall be defined with the same properties, all POUs must be selected. The desired attributes are activated by clicking the checkboxes. When an attribute is active, the checkbox is filled with a black checkmark. Inactive attributes must be unchecked (white). Gray checkmarks are not allowed.



## Configuring individual program blocks or variables

If only variables of individual POUs or individual variables shall be activated, first all attributes of the entire project must be deactivated. To do so, select all POUs and deactivate all attribute checkboxes. Now, all checkboxes must be white.



**Caution:** The deletion of object attributes is not applied until it is confirmed with the <OK> button. Each deletion must be confirmed with the <OK> button prior to defining further settings. Otherwise, the object attributes are not deleted.

The dialog 'Set object attributes' must be opened again. Now, the individual project POUs or variables can be selected and set with the desired attributes.

The following attributes can be set:

<b>Export variables of object</b>	The variables of the selected object are written to the symbol file.
<b>Export data entries</b>	Entries for accessing the total variables are created for structures and arrays of the object. Assumption: 'Export variables of object' is activated.
<b>Export structure components</b>	For structures of the object, one entry is created for each component of the variable. Assumption: 'Export variables of object' is activated.
<b>Export array entries:</b>	For the arrays of the object one entry is created for each component of the variable. Assumption: 'Export variables of object' is activated.
<b>Write access:</b>	The OPC server can modify the variables of the object.

### 3.2.3 Step 3: Setting of the communication parameters in the project

The channel of the used gateway is set in the dialog <Online> <Communication Parameters>. (Further information on the gateway configuration can be found in the documentation [CoDeSys / Volume 7 / Chapter 10 "Programming and Test"](#)). This setting must match the OPC server configuration (see section 3.3 "Configuration of the OPC server using OPCconfig.exe").



**Note:** For Single-PLC configuration:

The OPC server automatically starts with the connection settings last used (these settings are kept in the registry). These settings can be verified or changed in the file OPCconfig.exe prior to starting the OPC server. This means that the setting of the gateway is automatically updated in the configuration. Each time a project is sent to the PLC, the corresponding gateway settings are applied.

The gateway connection can be set to local or TCP/IP. A local connection is used, if the OPC server and the used gateway reside on the same computer. A TCP/IP connection is used, if the gateway of another computer is accessed by the OPC server via a network.



**Note:** A TCP/IP connection to the gateway can also be used, if the OPC server and the gateway reside on the same computer. However, with this setting, higher transmission times must be expected compared to the setting 'local'.



**Caution:** For parallel operation of the OPC server and CoDeSys or SyCon.net, the settings of the communication channels must match completely. Multiple access to a PLC is only possible via the same gateway channel. Here, not only the corresponding driver and address are important, but also all further settings such as the timeout value.

### 3.2.4 Step 4, optional: Store project

The communication parameters are saved with the project and, after a login, also in the gateway.

### 3.2.5 Step 5: Creating of the symbol file and transmission to the gateway

When compiling the project, the symbol file is created and stored to the project directory. When downloading the project (<Online> <Login> -> 'Download'), the symbol file (\*.sym or \*.sdb) with the present settings is additionally loaded to the gateway (directory 'Gateway Files'). The communication with the OPC server uses the settings of the symbol file located in the path \\WinNT\\Gateway Files\\, for example.



**Note:** The number of symbols should not exceed approx. 15 000 (corresponds to a symbol file size (\*.sym) of approx. 1.5 MB). Please refer to chapter 4 "Behaviour of the OPC server".



**Note:** When preparing a project in the office without having the corresponding hardware available, the symbol file must be copied to the system path \Gateway Files after finishing the CoDeSys project. This means that the file <project name>.sdb must, for example, be copied to the directory C:\WINNT\Gateway Files.

The representation of variables in the symbol file is the same as in the watch window.

Example:

```
PLC_PRG.A
PLC_PRG.structure.X[4]
.GlobVar1
```



**Caution:** Direct addresses cannot be accessed.

### 3.3 Configuration of the OPC server with OPCconfig.exe



With the configuration it is determined which controller shall read the symbol entries. For this, the gateway channel must be determined, the project name must be entered and the individual parameters (e.g. transmission rate, timeout values, etc.) must be defined. The OPC standard does not include an interface for the transmission of this information. For that reason, the project identifier is written to the registry on each download of a project (refer to section 3.3.5 "Registry entries (example)"). The OPC server reads this identifier and searches the symbol file with the same name.



**Caution:** Primarily it is important that in the OPC server configuration the project name corresponds to the used communication driver (i.e. if communication channel = TCP/IP\_Address 192.68.0.251 and project name = OPC\_test\_program, the Ethernet subscriber with the IP address 192.68.0.251 must contain the project OPC\_test\_program).

For setting these parameters, please refer to section 3.2.3, step 3: "Setting the communication parameters in the project" and the CoDeSys documentation respectively.

First, it has to be determined which mode is used: Single-PLC or Multi-PLC. The mode of operation depends on the following factors:

Mode of operation	Factors
Single-PLC	<p>- is used, if the symbol entries are read by only one controller.</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">  <p><b>Note:</b> In Single-PLC mode, the configuration of the gateway channel is adapted automatically. This means that each time a project is sent to the PLC, the corresponding gateway settings are applied to the OPC configuration.</p>  <p><b>Caution:</b> Single-PLC mode should only be used if the configuration shall be adapted automatically.</p> </div> <p>- Generally, we recommend to use the setting Multi-PLC.</p>
Multi-PLC	<p>- is used if the symbol entries are read by only <b>one controller</b> and the <b>configuration shall not be adapted automatically</b>. This means that the settings in</p>

the OPC configuration are adapted manually. For instance, this is important if you have set up an Ethernet network with several controllers and the symbol file shall only be read by one controller (head controller). If the programming of the other controllers would be performed via the same gateway, the OPC configuration would be changed with each program download.



**Note:** For the time being, OPC can only access variables of one controller. Communication with several subscribers will be released in a future version.

The following notes should be observed for the use:



**Note:** A download of the desired project to the corresponding target controller should be performed immediately before using the OPC server. If the target controller is not available, the current symbol file should be copied into the gateway files directory.



**Note:** When adding or deleting variables in the project, the item list can be updated without the need to exit the OPC client and the OPC server by downloading once again (with the option 'Dump symbol entries' activated). For instance, the client then receives a corresponding message (status "Bad") when it tries to access a deleted variable.

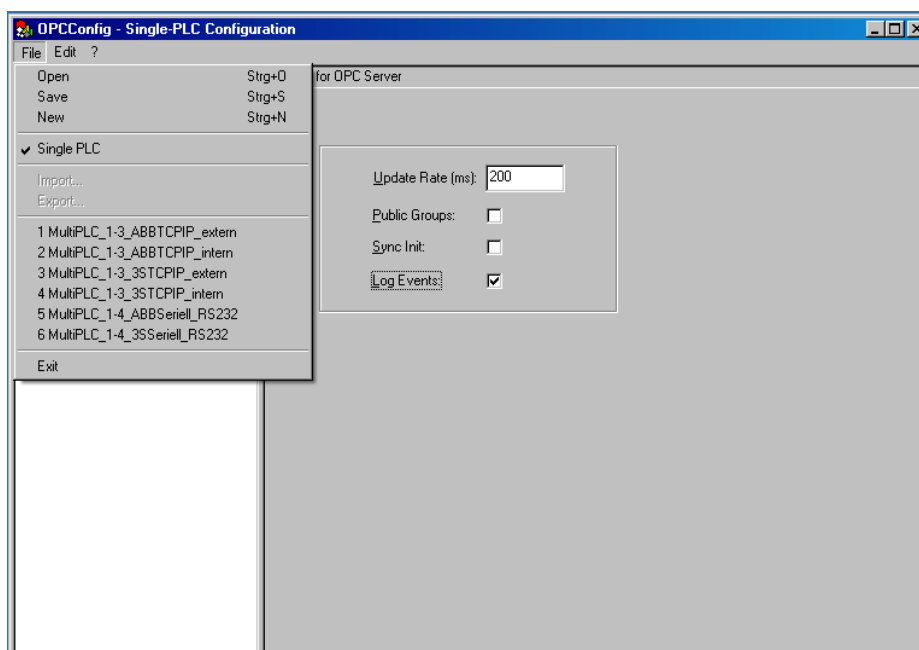



**Note:** Please note that a possibly existing boot project does no longer match a project which was changed and downloaded again. Hence, it is recommended to store the program in the PLC Flash memory each time after downloading it.

For configuring the OPC server and modifying the registry entries of the server, the configurator OPCconfig.exe is started (refer to OPC documentation, section 3.3.5 "Registry entries (example)").

The following sections describe the commands of the <File> and <Edit> menu as well as the specific dialogs for the Single-PLC or Multi-PLC configuration.

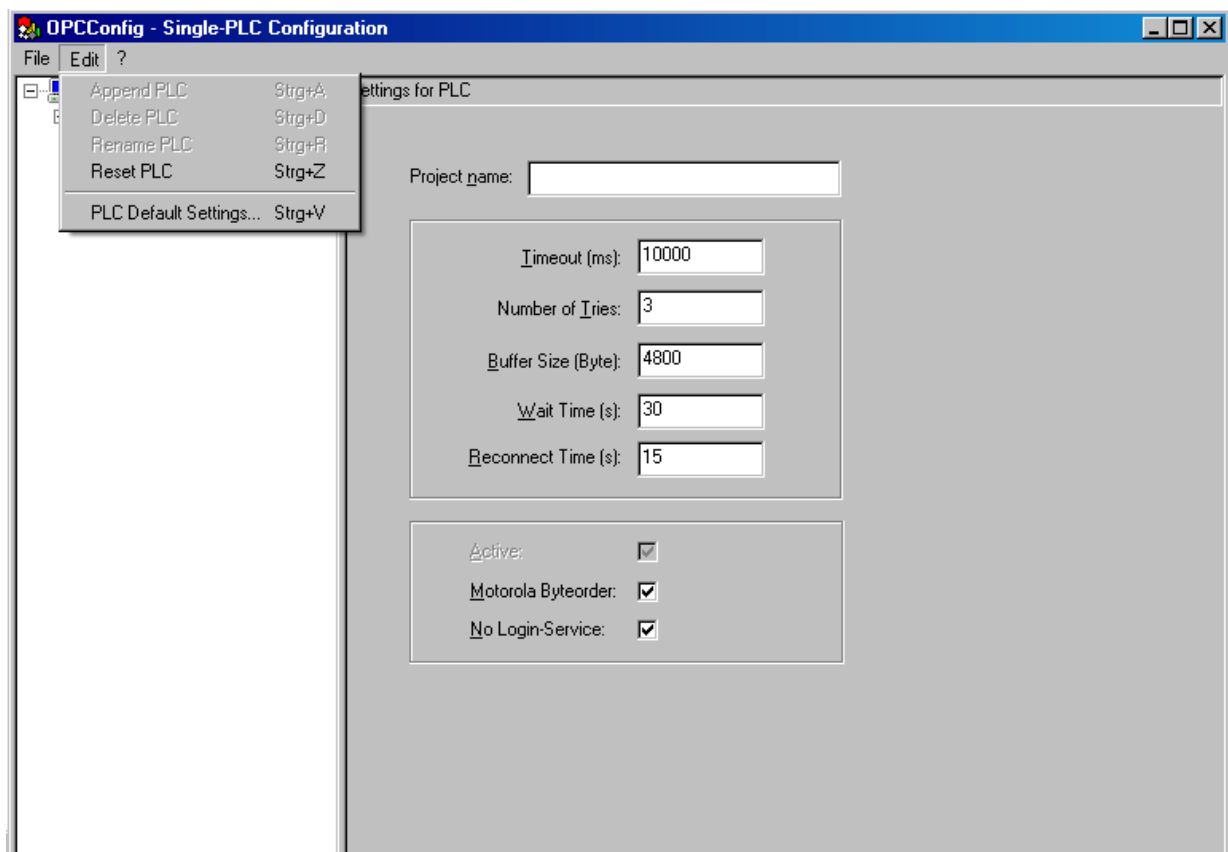
### 3.3.1 Menu <File>






<b>Open</b>	This command opens the configuration saved last using <File> <Save>.
<b>Save</b>	After the configuration of the server, the PLC(s) and the connection(s) is finished, it has to be saved using this command in order to become valid.
<b>New</b>	This command opens (creates) a new configuration. The dialogs contain default settings or empty input fields.
<b>Single-PLC</b>	Using this option, the mode Single-PLC or Multi-PLC is selected. If the menu entry 'Single-PLC' is displayed, Multi-PLC mode is currently active. If 'A; Single-PLC' is displayed, Single-PLC mode is active.
<b>Export Import</b>	<div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; margin-bottom: 10px;">  <b>Caution:</b> This menu item is only available in Multi-PLC operating mode. </div> <p>Using the &lt;Export&gt; command, the current configuration can be exported to a text file. For this, the standard dialog 'Save file as' is opened where you can enter a file name. The default file name extension ".ini" is appended automatically. The ini file contains the configuration parameters and can be edited and re-imported using the command &lt;File&gt; &lt;Import&gt;. When importing a file, the stored ini files can be selected in the standard dialog 'File open'.</p>

### 3.3.2 Menu <Edit>

The following commands are available depending on the selection in the configuration tree ('Server', 'PLC' or 'Connection') or depending on the type of configuration (single PLC or multi PLC).



<b>Append PLC</b>	<div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; margin-bottom: 5px;">  <b>Caution:</b> This menu item is only available in Multi-PLC operating mode. </div> <p>Multi-PLC mode, settings for the OPC server: Using this option, another controller is inserted to the configuration; refer to section 3.3.4 "Multi-PLC configuration" in the OPC documentation.</p>
<b>Rename PLC</b>	<div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; margin-bottom: 5px;">  <b>Caution:</b> This menu item is only available in Multi-PLC operating mode. </div> <p>Multi-PLC mode, settings for the OPC server: Using this option, a controller previously inserted to the configuration can be renamed; refer to section 3.3.4 "Multi-PLC configuration" in the OPC documentation.</p>
<b>Delete PLC</b>	<div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; margin-bottom: 5px;">  <b>Caution:</b> This menu item is only available in Multi-PLC operating mode. </div> <p>Multi-PLC mode, settings for the OPC server: Using this option, a controller can be deleted from the configuration; refer to section 3.3.4 "Multi-PLC configuration" in the OPC documentation.</p>
<b>Reset PLC</b>	<p>Single-PLC and Multi-PLC, settings for the PLC: The configuration parameters set with &lt;Edit&gt; &lt;PLC Default Settings&gt; are loaded from the registry.</p>
<b>PLC Default Settings</b>	<p>Always available: The dialog 'PLC Default Settings' appears. Here, you can enter self-defined values for the PLC configuration settings. This dialog corresponds to the 'Settings for PLC' dialog (refer to OPC documentation, section 3.3.3. "Single-PLC configuration"). The values set in this dialog are saved to the registry after confirming the dialog with &lt;OK&gt;. Using the command &lt;Edit&gt; &lt;Reset PLC&gt;, these settings can be re-loaded for a PLC selected in the configuration tree.</p>

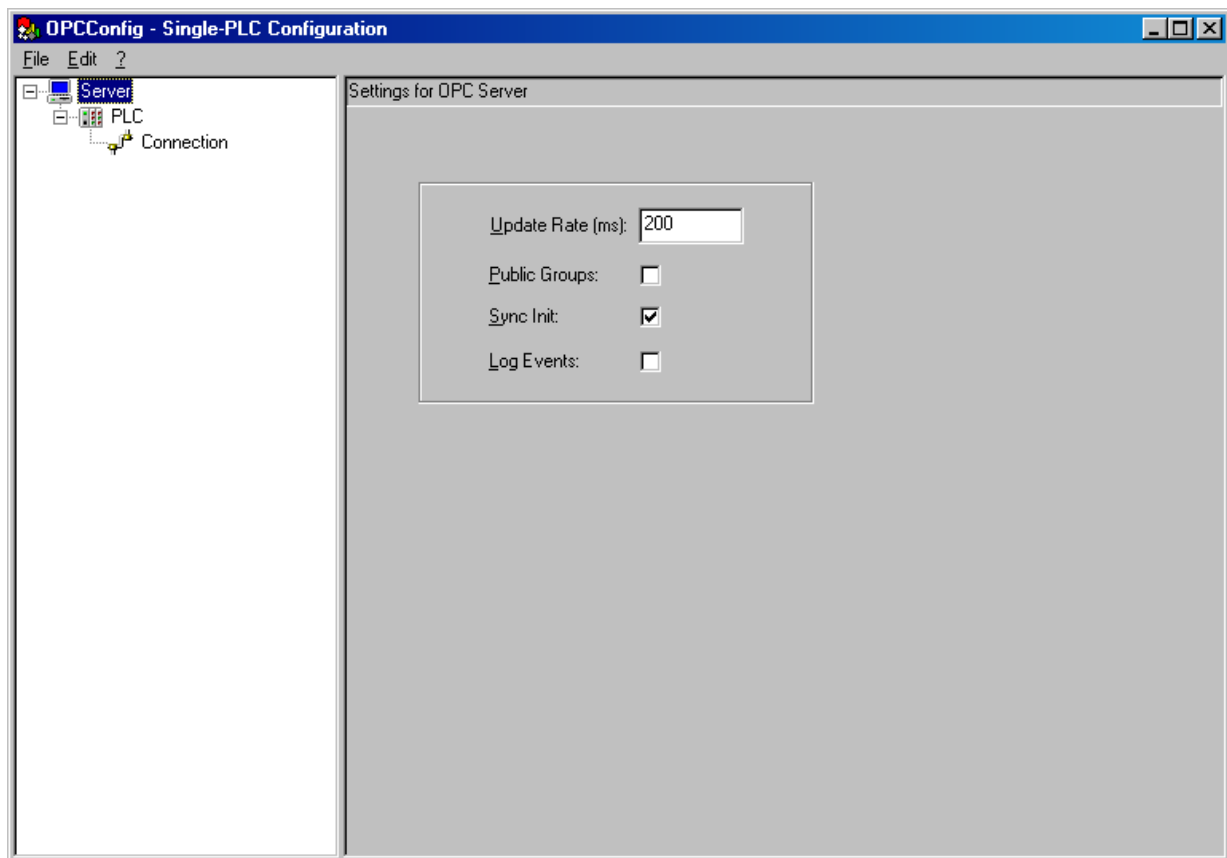
### 3.3.3 Single-PLC configuration

Generally, we recommend to use the setting Multi-PLC. Proceed as follows to configure an OPC server for Single-PLC mode:



**Note:** Possibly appearing dialogs for saving are not mentioned in the following procedure.

Start the program OPCconfig.exe. First, activate the Single-PLC mode by selecting the option <Single-PLC> from the <File> menu. The menu entry is then marked with a checkmark. -> **A; Single PLC**. Select the menu option <File> <New> to open the following default dialog.



In the left area of the window, the **server**, the controller (**PLC**) and the **connection** are represented in a tree structure and can be selected by clicking with the mouse. The right part of the window displays the corresponding configuration dialog for the entry selected in the tree structure. The <Edit> menu is not used for the Single-PLC configuration.

The following server settings can be made in the right part of the window, if the topmost entry 'Server' is selected in the configuration tree.

**Update Rate** (ms): Default: 200

Basic update rate of the OPC server in milliseconds = Cycle time used for reading all item values from the controller. These values are written to the cache the client communicates with using a separately defined update rate.



**Caution:** If the update rate is set too low, no communication can be established to the controller. The status of the items is displayed as BAD in the client. Please also refer to chapter 4 "Behaviour of the OPC server".

**Public Groups:** Default: deactivated (no checkmark)

The OPC server creates one public group for each IEC block as well as for the global variables.

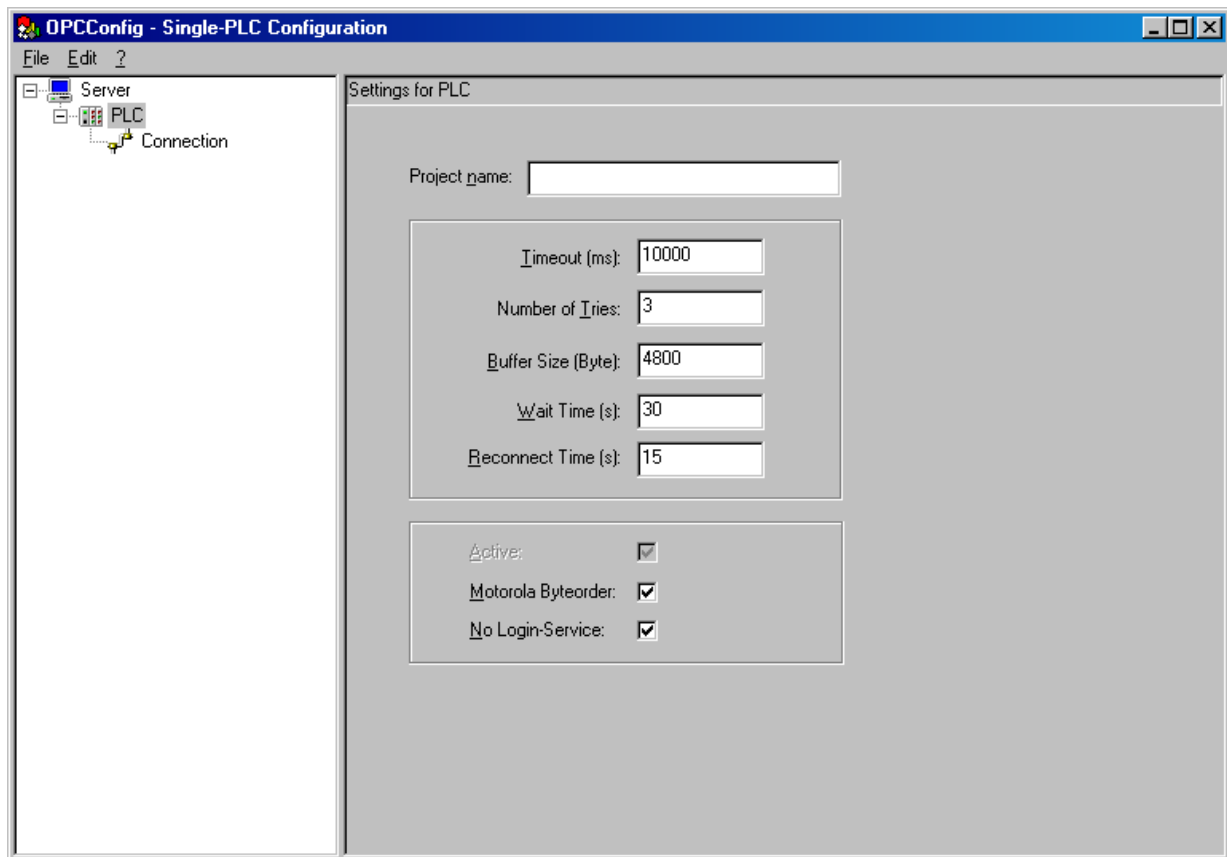
**Sync Init:** Default: activated (checkmark)

Synchronous initialization: When starting, the OPC server does not react until the symbol configuration is loaded.

**Log Events:** Default: deactivated (no checkmark)

If this option is activated, the actions performed and the errors occurred at the OPC server are logged and saved to a log file. The file is saved to the project directory and named OPCServer.log. The messages of several OPC sessions are subsequently listed in a log file.

The following controller settings can be made in the right part of the window, if the entry 'PLC' under 'Server' is selected in the configuration tree.



#### Project name:

Name of the project currently loaded. For Single-PLC mode, this entry is not mandatory.



**Note:** If the communication with the OPC server is performed via the gateway channel which was used for downloading the program to the PLC, no project name has to be entered. If another gateway channel is used for the OPC server, the project name has to be entered here.

#### Communication Timeout (ms): Default: 10000

The OPC server exits automatically after this time period if it does not receive an answer to a transmitted service from the controller.

#### Number of Tries: Default: 3

Number of attempts the OPC server tries to re-establish the connection to the controller after a timeout. The time interval between the re-establishment attempts is defined at 'Reconnect Time'.

#### Buffer Size (Byte): Default: 4800

Communication buffer size on the target system. For all AC31 controllers, a buffer size of < 5000 must be entered.

#### Wait time (s): Default: 10

Time in seconds for which the OPC server waits until the controller is ready (important for controller autostart).

After the OPC server is started, it tries to communicate with the configured controllers. Then it notifies the client whether communication to the controllers is possible or not. The controllers must log in within the set wait time, otherwise the status is passed to the client.



**Note:** The wait time must be defined long enough to enable all subscribers to log in. If this time is set too short it could happen that some subscribers cannot log in during the start. In this case, the wait time setting should be increased.

This time depends on the number of subscribers and the number of items defined in the symbol file.



**Note:** After modifying the CoDeSys project and adding new items to the symbol configuration, the previously set parameters are possibly no longer correct. In this case the parameters must be adapted (i.e. the wait time must be increased).

**Reconnect Time (s):** Default: 10

Time interval during which the OPC server attempts (as often as set at 'Number of Tries') to re-establish the connection to the controller via the gateway after a communication break-off.

**Active:** Not available (only available in Multi-PLC mode)

**Motorola Byteorder:** Default: deactivated (no checkmark)

The target system does not use the Motorola byte order. This option must be activated for all AC500 controllers.



**Note:** This function is active in all AC500 controllers. It is not active for all AC31 controllers.

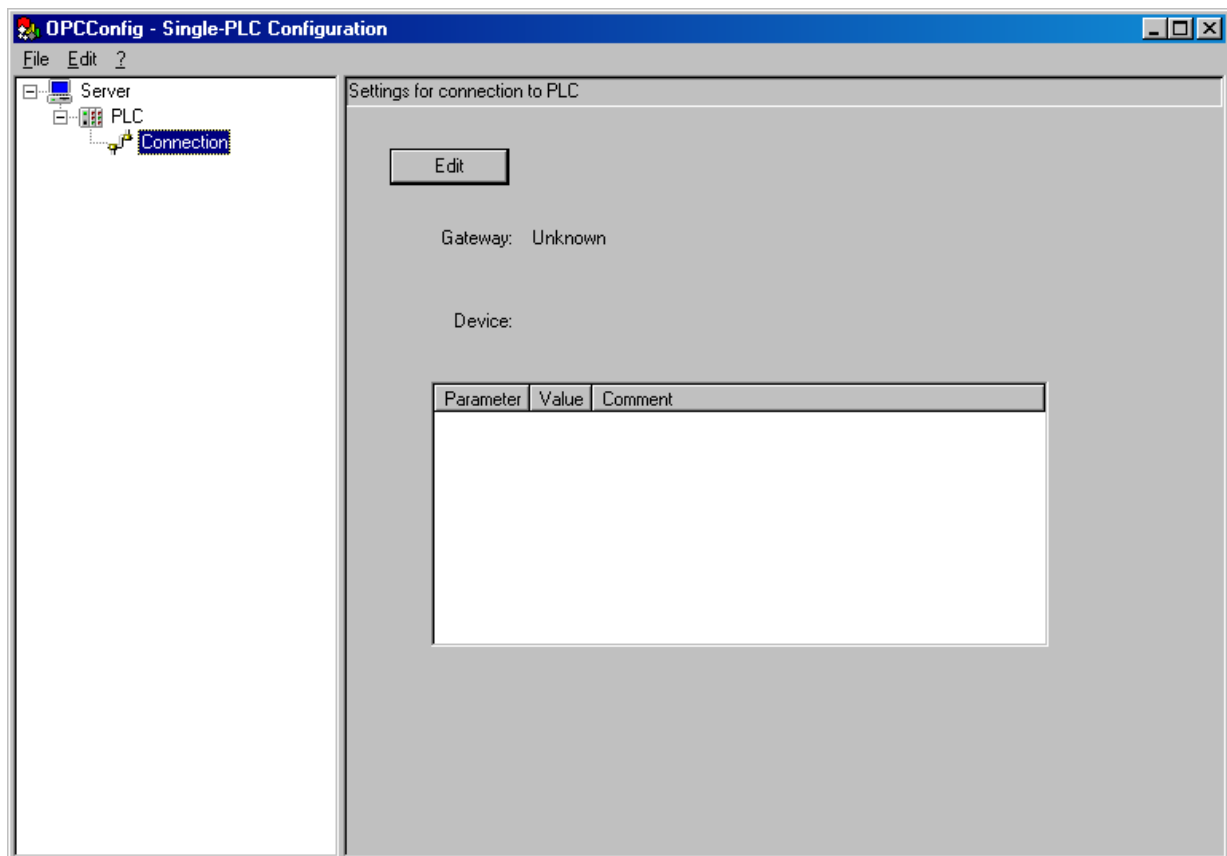
**No Login Service:** Default: activated (checkmark)

This option must be deactivated for specific target systems which require the transmission of a login service.

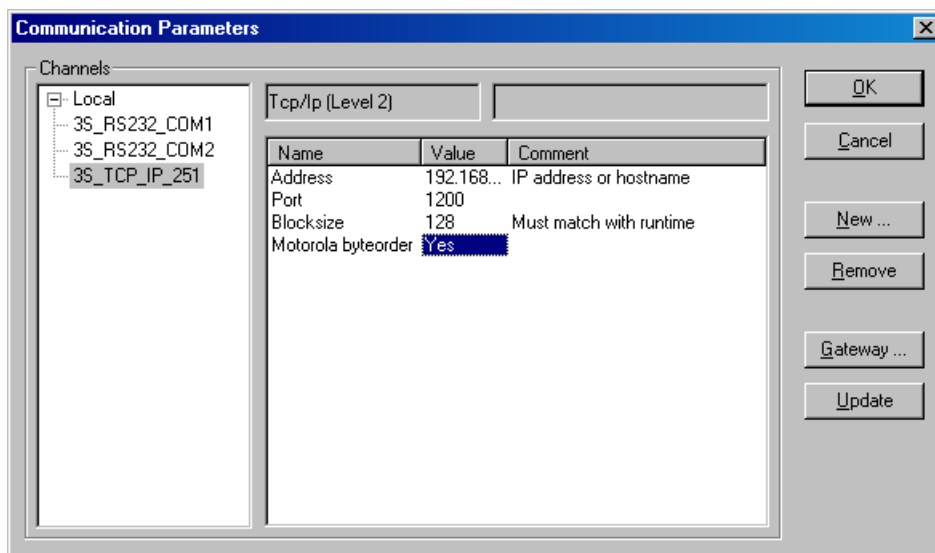


**Note:** The login service must not be transmitted for all ABB controllers. This function must be activated.

The following settings for the connection to the PLC can be made in the right part of the window, if the entry 'Connection' is selected in the configuration tree.



If no settings are made yet or if the existing settings shall be modified, the **Communication Parameters** dialog can be opened by clicking on the button **Edit**. This dialog is identical to the gateway settings available in the programming software CoDeSys when selecting <Online> <Communication Parameters>.

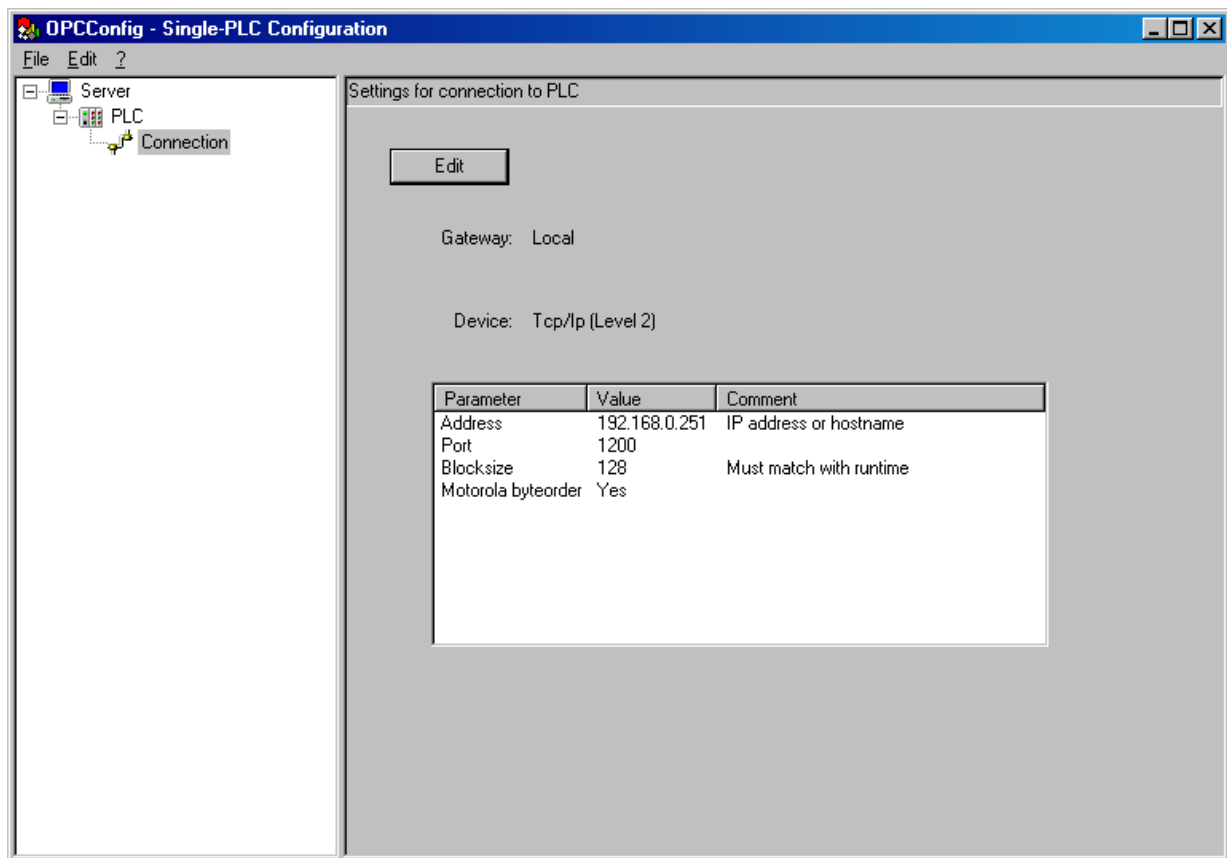


[Click here to magnify / reduce image](#)

First you have to define whether the communication shall use a local or a TCP/IP gateway channel. In addition, a list of all driver settings loaded via the gateway up to now is displayed. Here, the driver must be selected with which the OPC server shall communicate. Confirm the settings with <OK>.

Please refer to section 3.2.3, step 3: "Setting the communication parameters in the project" and the CoDeSys documentation respectively.

Now your selection should be applied. The dialog box looks as follows:



### 3.3.4 Multi-PLC configuration

Generally, we recommend to use the setting Multi-PLC. To activate this operating mode, the option 'Single-PLC' has to be deactivated in the <File> menu.



**Note:** For the time being, OPC can only access variables of one controller. Communication with several subscribers will be released in a future version.

First, only the entry 'Server' is displayed in the left part of the window. The desired controllers must be inserted using the command <Edit> <Append PLC>. Then they can be configured individually as already described for the single PLC configuration.

Differences compared to the Single-PLC configuration:

In the **Edit** menu, the following commands are additionally available for the PLC currently selected in the configuration tree:

<b>Append PLC</b>	A new 'PLC' entry is inserted. Additional PLCs are always inserted below the currently highlighted entry. The name "PLC<n>" is automatically assigned, where n is counted up beginning with 1.
<b>Delete PLC</b>	The highlighted entry is removed from the configuration.
<b>Rename PLC</b>	Each PLC can be assigned a user defined name. To do so, highlight the desired PLC entry and then select <Edit> <Rename PLC>. An edit field is opened where you can enter the name.

The name of the project which was downloaded to this controller via the gateway can be entered to the **Project name** input field of the 'Settings for PLC' dialog.



**Note:** If communication with the OPC server is performed via the same gateway channel as for downloading, the program to the PLC, it is not required to assign a project name. The project name has to be entered, if another gateway channel is used for the OPC server.

By selecting or deselecting the option Active in the 'Settings for PLC' dialog it can be defined whether the controller shall be considered by the OPC server or not.

### 3.3.5 Registry entries (example)

The registry entries can be displayed using the command "regedit" in the command prompt.

Entries for the 'Settings for OPC server' dialog of the OPC configurator:

#### [HKEY\_CURRENT\_USER\Software\3S-Smart Software Solutions GmbH\CoDeSys OPC\Connections]

\* corresponds to the values set in the dialog 'Settings for PLC' of the OPC configurator (refer to OPC documentation, section 3.3.3 "Single-PLC configuration"): \*

- "Buffersize"="0x000012c0 (4800)"
- "Instances"="00 00 00 00 00 00 00 00"
- "Motorola0"="Yes"
- "Name0"=""
- "NoLogin0"="No"
- "Project0"=hex (current communication parameters for the OPC server with project identifier)
- "Timeout0"="0x00002710 (1000)"
- "Tries0"="0x00000003 (3)"

#### [HKEY\_LOCAL\_MACHINE\SOFTWARE\3S-Smart Software Solutions GmbH\CoDeSys OPC\Config]

\* corresponds to the values set in the dialog 'Settings for OPC server' of the OPC configurator (refer to OPC documentation, section 3.3.3 "Single-PLC configuration"): \*

- "LogEvents"="No"
- "Multi"="no"
- "Public groups"="No"
- "Reconnect Interval"="0x00000005 (5)"
- "Sync startup"="yes"
- "Updaterate"="0x00000064 (100)"
- "Wait for target"="0x0000000a (10)"

#### [HKEY\_LOCAL\_MACHINE\SOFTWARE\3S-Smart Software Solutions GmbH\CoDeSys OPC\PLCDefaults]


\* corresponds to the values entered at 'Edit' 'PLC Default Settings' of the OPC configurator (see OPC documentation, section 3.3.2 "Menu <Edit>") \*

- "Activ"="true"
- "Buffersize"="4800"
- "Motorola"="false"
- "Nologin"="true"
- "project"=""
- "Reconnecttime"=10"
- "Timeout"="10000"
- "Tries"="3"
- "Waittime"="10"

### 3.4 Start of CoDeSysOPC.exe

The OPC server (and the gateway server) is automatically started by the operating system as soon as one of the clients establishes a connection. It is not necessary to start it explicitly. If an OPC server has already been started manually, the client automatically connects to the running server.

The server is automatically exited again as soon as all clients have cleared the connection.

While the OPC server is running, the icon  is displayed on the right side of the status bar on the bottom of the screen.

Right-clicking on this icon displays information about the version of the OPC server.

#### Starting in test mode:

The connection between the OPC server and the client can also be tested if no controller is connected. For this purpose, the command

```
CoDeSysOPC.exe /TestMode
```

has to be entered to start the OPC server manually. The server then runs in test mode and automatically generates a series of test items which can be used to test the client communication.



**Caution:** The items generated in test mode are only test variables. These are not the variables from the symbol file of your project.


The OPC icon in the status bar is also displayed while simulation mode is running.

### 3.5 End of CoDeSysOPC.exe

The OPC server is automatically exited as soon as all clients have cleared the connection.

### 3.6 Starting and ending of Gateway.exe

The gateway is automatically started by the operating system as soon as one of the subscribers (e.g. CoDeSys, OPC server, OPCConfig or similar) tries to access the gateway server.

The icon  on the right side of the status bar on the bottom of the screen indicates that the gateway was started and whether it is in use or not.

Once the gateway is started, it stays opened. However, it is indicated whether it is in use or not. When the gateway is inactive, the icon is shaded in gray . It is not necessary to close the gateway.

If required, the gateway can be exited manually. To do so, right-click on the gateway icon to open the context menu. Select <Exit> to close the gateway server.

## 4 Behaviour of the OPC server

### 4.1 General recommendations

The configuration should be optimized as good as possible to enable shortest possible update times. This is influenced by the following:

#### - Number of subscribers (controllers):

This setting is performed in the OPC configuration. The lower the number of subscribers, the faster the communication.



**Note:** For the time being, OPC can only access variables of one controller. Communication with several subscribers will be released in a future version.

#### - Number of items:

This setting is defined in the CoDeSys project (creation of symbol file). Only variables which shall be visualized later should be added to the symbol file. The lower the number of variables, the faster the configuration.

#### - Number of active items:

This setting is defined in the client (visualization software). The items should be assigned to individual groups which are updated at the same time. The groups are only enabled when they are needed, i.e. not all items of the project must be updated simultaneously. As a result, only the currently needed items are read.

#### - Correct parameter assignment when setting the OPC configuration:

\*The setting of the following parameters is mandatory for AC500 controllers:

- Buffer size: 4800
- Motorola Byteorder: active
- No Login Service: active

\* Parameter is used for optimizing the communication:

In order to reach a stable communication, the following parameters must be set high enough.

- Updaterate
- Timeouttime
- Waittime
- Reconnecttime

### 4.2 Time response of the OPC server

For serial connection it is recommended to use the Single-PLC mode.

For Ethernet connections generally the Multi-PLC mode could be used (this depends on the function of the automatic configuration adaptation, refer to section 3.3 "Configuration of the OPC server using OPCconfig.exe"). The more subscribers and items are configured, the slower is the communication.



**Caution:** If the transmission rate is set too low in the configuration, the communication becomes considerably slower or is no longer possible. The transmission rate should be set approximately to the value which can be reached.

	Serial				Ethernet			
	Single-PLC		Multi-PLC		Single-PLC		Multi-PLC	
	Up- date rate [ms]	Measured time approx. [s]	Up- date rate [ms]	Measured time approx. [s]	Up- date rate [ms]	Measured time approx. [s]	Up- date rate [ms]	Measured time approx. [s]
<b>1 subscriber</b>								
4 Items	50	0.1	50	0.2	xx	xx	xx	xx
100 Items	500	1	500	1	xx	xx	xx	xx
500 Items	1000	3	1000	3	xx	xx	xx	xx
800 Items	1000	7	1000	7	xx	xx	xx	xx

## 5 Connection to an OPC server on another PC

### 5.1 DCOMCNFG.EXE

Using DCOMCNFG.EXE from the system directory, a connection to an OPC server on another computer can be established. For this purpose, select the entry 'OPC server for CoDeSys...' in the 'Applications' tab and then click on the 'Properties' button. In the properties dialog, open the 'Location' tab, activate the option 'Run application on the following computer' and then enter the desired computer.

Precondition for such a connection is that an OPC server is also running on the local computer.

## 6 Example of an ini file for the OPC server configuration

### 6.1 Example of a multi-PLC configuration

The ini file of the OPC server configuration can be exported from OPCConfig.exe to a text file and then edited and re-imported again to OPCConfig.

The following is an example of ini file entries (Multi-PLC mode with 1 subscriber):

Entry	Meaning
[Server]	Settings for the OPC server
updaterate=200	Transfer rate [200 ms]
publicgroups=0	Public Groups [deactivated]
logevents=1	Create log file [activated]
syncinit=1	Sync Init [activated]
PLCs=1	Number of subscribers [1]
PLC0=PLC1	PLC name subscriber 1 [PLC1]
[PLC:PLC1]	Settings for PLC1
active=1	Subscriber active [activated]
motorola=1	Motorola byte order [activated]
nologin=1	No login service [deactivated]
timeout=10000	Timeout value [10s]
tries=3	Number of attempts to reconnect [3]
waittime=10	Wait time [10s]
reconnecttime=10	Reconnect time [10 s]
bufferize=4800	Buffer size [4800]
project=PLC1.pro	Project name [PLC1.pro]

gateway=Local	Gateway channel [local]
device=ABB TCP/IP Level 2	Parameter for gateway connection to PLC1 (driver setting): - Gateway driver [ABB TCP/IP Level 2] - Driver name [TCP_IP_251] - IP address 192.168.0.251
instance=TCP_IP_251	
parameters=11	
parameter0=Address	
value0=192.168.0.251	
parameter1=Port	
value1=1200	
parameter2=Receive Timeout	
value2=2000	
parameter3=Routing levels	
value3=0	
parameter4=Coupler (Level 1)	
value4=0	
parameter5=Channel (Level 1)	
value5=0	
parameter6=Address (Level 1)	
value6=0, 0, 0, 0, 0	
parameter7=Coupler (Level 2)	
value7=0	
parameter8=Channel (Level 2)	
value8=0	
parameter9=Address (Level 2)	
value9=0, 0, 0, 0, 0	
parameter10=Block size	
value10=128	

## 7 Short checklist

### 7.1 Short checklist

Please check the following if the communication via OPC does not work:

1. Gateway installed and running? Icon in the taskbar (bottom right) active?
2. Installation and registration of the OPC server ok?  
Is automatically performed when "CoDeSysOPC /Install" (installation + registration) or "CoDeSysOPC /RegServer" (registration only) are executed.
3. Project preparation / settings in the programming system CoDeSys ok?
  - Project options: 'Dump symbol entries' selected?
  - Communication parameters: Selected gateway channel ok?
  - Project saved, compiled and download performed?
  - The symbol files \*.sym and \*.sdb must exist in the path WinNT\Gateway Files.
4. If OPC shall be used on another computer (only possible if the OPC server is also running on the local computer and if the CoDeSys project is not additionally accessed (currently logged in) on the other computer):  
Establish a connection to the other computer using DCOMCNFG.EXE (file is located in the system directory):
  - select the entry 'OPC server for CoDeSys...' in the 'Applications' tab,
  - click on the 'Properties' button, open the 'Location' tab, activate 'Run application on the following computer' and then enter the desired computer.
5. Verify the connection and the server settings: OPCCFG.EXE:
  - Connection: (entry must correspond to the valid gateway channel defined in the CoDeSys communication parameters)
  - Usual server settings: Update Rate 200 ms, Public Groups: no, Sync Init: no
  - Usual PLC settings: Communication timeout 10s (10000), Number or reconnect attempts: 3, buffer size: 4800, wait time target system 10s, reconnect interval: 10s, Active: yes, Motorola Byteorder: yes, No Login Service: yes
  - ..Connection settings for PLC: Warning: For Multi-PLC mode, local and TCP/IP cannot be mixed, verify driver settings
  - Save -> The new configuration becomes effective when a new connection to the OPC server is established
6. Verify whether symbol entries are available: DiagnosticOPCClient.exe  
Establish connection (Connect OPC-Server) -> in the right-hand window, right-click on Private Groups, confirm the dialog with OK, right-click on the created group entry -> add all items -> all created symbol entries (variables) should be displayed now.

The OPC server is automatically exited as soon as all clients have cleared the connection to it.

The OPC server is displayed in the Task Manager as a process.

## **8 Index OPC**

### **B**

Behaviour of the OPC server 26

### **C**

Connection to an OPC server on another PC 27

### **E**

Example of an ini file for the OPC server configuration 27

### **G**

Glossary OPC 31

### **I**

Installation of the CoDeSys OPC server 9

### **O**

Overview of OPC 3

### **S**

Short checklist 29

### **U**

Use of the CoDeSys OPC server 12

## 9 Glossary

OPC = OLE for Proceess Control

OLE = Object Linking and EMBEDDING

For further information see also <http://www.opcfoundation.org> and <http://www.opceurope.org>

COM = Component Object Model (Basis für OLE)

DCOM = Distributed Component Object Model







---

**ABB STOTZ-KONTAKT GmbH**

Eppelheimer Straße 82 69123 Heidelberg, Germany  
Postfach 10 16 80 69006 Heidelberg, Germany  
Telephone (06221) 701-0  
Telefax (06221) 701-240  
Internet <http://www.abb.de/stotz-kontakt>  
E-Mail [desst.help@de.abb.com](mailto:desst.help@de.abb.com)