CoDeSys OPC-Server V2.0

Installation and Usage

Document Version 1.7

CONTENT

1 Brief Introduction to OPC

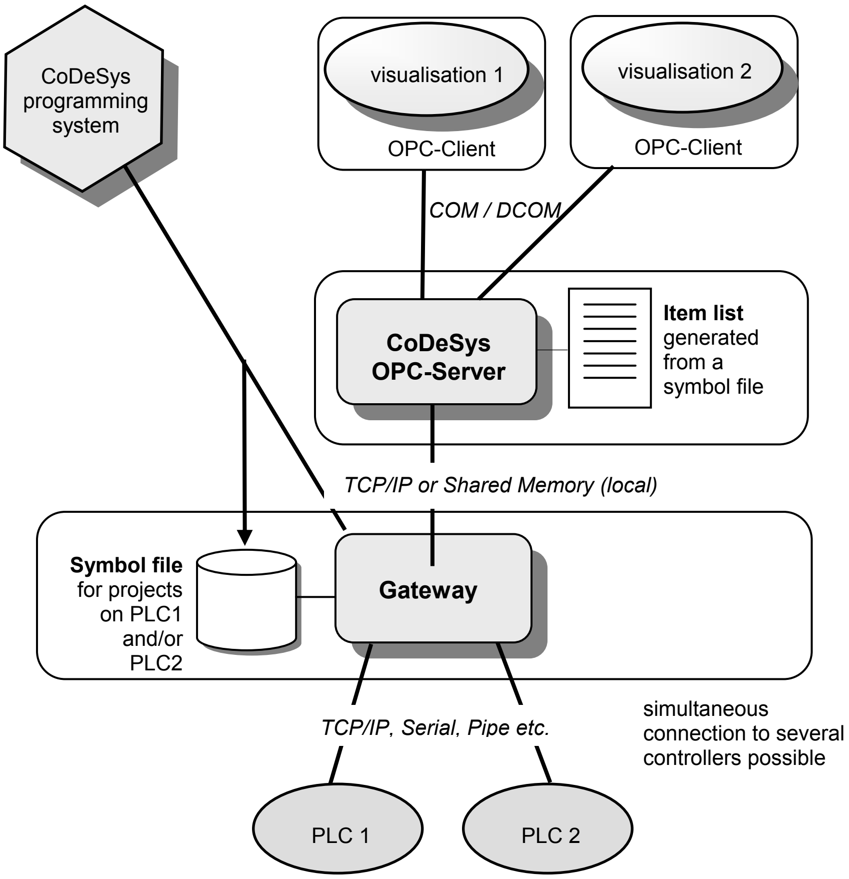
OPC[1] is a standardised interface giving access to process data. It is based on the Microsoft standard COM/DCOM[2] and was extended to meet the needs of data access in the automation process where it is mostly used to read and write data from the controller. The typical OPC clients are visualisations and programs for recording operational data etc.. PLC systems and field bus cards are usually supplied with an OPC server.

The OPC server is not a passive subprogram but an executable program which is started as soon as a connection is established between the client and the controller. It can therefore inform the client whenever the value of a variable or its status has changed.

Due to the features of DCOM it is possible to gain access to an OPC server which is installed on a remote PC. Via OPC it is also possible for several clients to gain access to a source of data simultaneously. Another advantage offered by OPC due to COM technology is programming language independence (C++, Visual Basic, Delphi, Java). Nonetheless OPC substantially increases the resource consumption (memory and CPU time).

Bitte beachten: The CoDeSys OPC-Server V2.0 is able to communicate with all controllers programmed with CoDeSys. It complies with the requirements of the OPC standard V2.0.

Architecture of the CoDeSys OPC-Server V2.0:


CoDeSys programming system
visualisation 1
OPC-Client
visualisation 2
OPC-Client
COM / DCOM
CoDeSys OPC-Server
Item list generated from a symbol file
TCP/IP or Shared Memory (local)
Symbol file for projects on PLC1 and/or PLC2
Gateway
TCP/IP, Serial, Pipe etc.
simultaneous connection to several controllers possible
PLC 1
PLC 2

The CoDeSys OPC server is based on the CoDeSys Gateway Server (Gateway).

tech_doc_e.dot, / V1.0

[1] OPC = OLE for Process Control; OLE = Object Linking and Embedding

For further information see www.opcfoundation.org and www.opc_europe.org

[2] COM = Component Object Model (Basis for OLE); DCOM = Distributed Component Object Model

Please regard: **The symbol file is the basis for the exchange of data via the CoDeSys OPC server:**

At the same time that a project is loaded form the CoDeSys programming system onto the controller a **symbol file** (*.sym or *.sdb) can be generated and placed in the Gateway resp. possibly on the controller. The symbol file contains so called items. One item (data object) corresponds exactly with one variable in the control application program. The variables on the controller can be reached via these items. (More information on the symbol file see in document symbolfile_e.pdf !)

The OPC server requests the Gateway to submit the contents of the symbol files and then produces an **item list** using the received information. As the contents of the item list is determined by the accessibility of variables in the controller it can not be influenced by the OPC client. In each case the OPC server reads the symbol file which was last loaded over the Gateway channel for a project. Items of different projects, meaning of different application programs, can be reached simultaneously if the symbol file was provided by the same Gateway channel.

The item list in the OPC server is updated with the data from the controller or controllers at fixed intervals. In comparison to direct access to the controller, reading and writing of variables via this cache list leads to an increase in access time (max. approx. 1ms per item).

> Figures drawn from past experience show that the OPC can well cope with quantities of around 15 000 items and symbol files with an approximate size of 1,5 MB when loading into the item list. The capacity utilisation when reading and writing the variables depends on how many of the items are active and therefore also have to be taken into consideration when updating the data.

The OPC server allows for the grouping of the data. When doing so we differentiate between public groups which are provided by the OPC server and private groups which the client sets up himself. If the corresponding option is activated in the configuration, the OPC server groups the items i.e. the variables of a project according to the POUs (Project Organization Units) where they are used. One public group is set up per POU.

The private groups can be put together in the client from different items arbitrarily. Initially they have no influence on the groupings in the OPC server but can be transformed into public groups when necessary. It is sensible to work with private groups in order to be able to activate or deactivate certain groups of variables with only one single command depending on whether they are to be accessible or not.

Grouped files should be read from the OPC server consistently which means that all variables are read simultaneously. It is, however, important to know that this is not always possible when working with targets with limited communication buffers.

Please regard: **It is possible to gain access to an OPC server located on a different computer in the network if there is also an OPC server on the local computer.**

What is new compared to CoDeSys OPC-Server V1.0?

- It is possible to update the symbol table without having to stop a client.
- A multi-plc configuration is also possible which means that the client can be connected to several controllers at once.
- It is possible to export and import the OPC configuration (text file*.ini).

"Offline Load" of the symbols, possible with new versions of OPC-Server (V2.3.8.4 and higher) and Gateway-Server (V2.3.3.3 and higher):

Up to now: If at start of the OPC-client the corresponding controller was not running resp. no communication could be built up, then no symbol information at all was shown, although usually already a symbol file was available in the local upload directory (defined in the CoDeSys project options, category Directories) or in the Gateway upload directory, due to previous communications with the target.
Now: Now in such cases a symbol file will be searched, first in the project upload directory and if nothing is found there, in the Gateway upload directory, and the symbols at least will be displayed in the client ("Offline Load") even if of course with quality "Bad". This will only work however, if in the OPC configuration (see chapter 3.2) the respective **project name** is defined !

2 Installation of the CoDeSys OPC server

2.1 Required Files

1. Gateway files:
   Gateway.exe, GClient.dll, GUtil.dll, GDrvBase.dll, GDrvStd.dll und GSymbol.dll

   The Gateway SP10 or a higher version is needed for the OPC server 2.0!

   The Gateway files have to be in a directory which can be reached via the specified directory path e.g. under WINNT/system32 (Windows NT) or under Windows/system (Windows95):

   After the first start of the Gateway the path for the directory “Gateway Files” will be set up in the registry (default: C:\WINNT\Gateway Files). The symbol files generated beforehand by CoDeSys and placed in the project directory are copied to the Gateway directory when the connection with the controller is established. Depending on the Gateway version there are symbol files with the extension *.sym or with the extension *.sdb. *.sdb symbol files are the binary version of the *.sym files and can be read faster by the OPC server.

2. OPC server files: for setup and operational usage
   These can be placed in any directory, whereby the DLLs and the OPCenum.exe have to be located in a subdirectory called REDIST.

   - CoDeSysOPC.EXE (OPC server)

   - OPCCommonSetup.EXE (setup for files listed below)

   Subdirectory REDIST (standard OPC 2.0 files):

   - OPCPROXY.DLL
   - OPCCOMN_PS.DLL
   - CALLRPROXY.DLL
   - OCSDAAuto.DLL
   - OCSSpy_PS.DLL
   - OPCenum.EXE

3. OPCConfig.exe: configuration tool

   Set/check the communication parameters or server settings (registry entry)

4. DCOMCNFG.EXE (C:\WinNT\System32): Needed to establish a connection to an OPC server which is installed on another computer (see chapter 4).

2.2 Installation and Registration

1. Simultaneous Installation and registration

The command

"CoDeSysOPC /Install“ (if necessary give directory path of EXE)

starts a setup (OPCCommonSetup.exe) in the “execute” dialogue which sees through the installation of the OPC files. Then the OPC server registers automatically and a message informs the user whether the registration was successful or not.

2. Registration after successful Installation

The command

„CoDeSysOPC /RegServer“ (if necessary give directory path of EXE)

only registers the OPC server. The installation of the necessary files must have been done beforehand. A message informs the user whether the registration was successful or not.

This separate-registration-feature is important for updates or the creation of setups.

(for registry entries see also 3.2.5)

3. Uninstalling the OPC server

The OPC server is registered with the command CoDeSysOPC /UnRegServer or CoDeSysOPC /DeInstall. The registry entries are deleted but the installed files are not removed.

3 Operating the CoDeSys OPC Servers

After the OPC server is installed it should be offered for selection by the OPC client (e.g. visualisation).

In order to be able to offer data objects of a project to the OPC server from the CoDeSys programming system and to set up a connection via OPC please conduct the steps 3.1 to 3.3 listed below:

3.1 CoDeSys programming system : Generating data objects for a project

Step1: Select 'Projekt' 'Optionen' 'Symbolkonfiguration': 'Symboleinträge erzeugen' :

The Gateway works not based on a project file but on a symbol file with the same name and the extension ".sym" or "sdb" (binary) which includes the symbol entries for the project variables (see 2.1, Required files). In order for this symbol file to be generated automatically with every translation the dialogue "Create Symbol entries" has to be active.

Step 2: Configuration of the symbol file:

The modules for which the symbol options are to be set can be marked in the tree control of the project in the dialogue 'Set object attributes' which can be found under 'Options \ Symbol configuration \ Configure symbol file.

The following options are supported:

- 'Export variables of object': The variables of the chosen object are issued in the symbol file.
- 'Export data entries': Entries giving access to the aggregate variable are generated for the structures and arrays of the object. Condition: 'Export variables of object' active.
- 'Export structure components': For the structure of the object an individual entry is generated for each component of a variable. Condition: Condition: 'Export variables of object' active.
- 'Export array entries': For the arrays of the object an individual entry is generated for each component of a variable. Condition: 'Export variables of object' active.
- 'Write access': The variables of the object may be changed by the OPC server.

Step 3: Setting the communications parameters in the project:

The channel of the used Gateway is set in the dialogue 'Online' 'Communication parameter' (for the configuration of the Gateway also see CoDeSys user manual). These settings have to correspond with those configured in the OPC server. The OPC server starts automatically with the communication settings last used (entry in the registry). These settings can be checked or changed in OPCconfig.exe. For the OPC configuration see chapter 3.

! for multi plc configuration: Should the user want access to variables from several different controllers using the OPC server the individual projects have to be loaded onto these controllers via the same Gateway channel.

Step 4, optional: Save project -> Communication parameters are saved with the project, (after logging the same applies for the Gateway)

Step 5: Creating the symbol file and transfer to the Gateway 'Online' 'Login' -> 'Download'

When the project is translated the symbol file is generated and placed in the project directory. If the project is downloaded ('Online' 'Login' -> 'Download') the symbol file is also loaded onto the Gateway with the current settings (directory 'Gateway Files').

Please note: The number of the symbols should not exceed approx. 15000 which corresponds to a symbol file *.sym with a size of 1.5 MB.

The declaration of variables in the symbol file is done in the same way as in the watch window.

e.g. :

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

There is no access to direct addresses.

3.2 Information on which symbol files are to be edited and configuration of the OPC server with OPCconfig.exe

As an OPC server has to deal with data of several different clients it has to be absolutely clear which symbol entries of which project are to be transferred first. There is no interface to transfer this information in the OPC standard. This is why the project identification is placed in the registry (entries see below) every time a project is downloaded. The OPC server reads this identification and searches for the symbol file with the corresponding id.

⇨ It is advisable to download the desired project onto the relevant target controller directly before starting the OPC server.

⇨ If variables are added to or deleted from the project the item list can be updated by a renewed download (the option 'Dump symbol entries' has to be activated) without the OPC server and the clients having to stop. Should the client try to gain access to a deleted variable for example a corresponding message will appear.

⇨ Please note that an existing boot project no longer corresponds with project that has been altered and downloaded anew!

Start the configurator OPCConfig.exe to configure the OPC server and to edit the corresponding registry entries (see 3.2.5).

! It is very important to configure the communication channel which is used by the OPC server and which corresponds with the one used in the CoDeSys project.

The commands of the menu 'File' and 'Edit' as well as the special dialogues for the single PLC and the multi PLC configuration are explained below.

Remark: Regard the possibility to load configuration settings to the configurator via command line. See on this the description of 'File' ‚Import' in 3.2.1.

3.2.1 Menu 'File'

'Open'	This command opens the configuration last saved with 'File' 'Save'.
'Save'	If the configuration of the server, the PLC(s) and connections is completed it has to be saved with this command to become valid.
'New'	This command opens a configuration which is to be set up anew. This means that the dialogues contain the standard entries or empty fields.
'Export' 'Import'	The current configuration can be exported into a text file using this command. The standard dialogue 'File Save as' appears for this purpose and enables the user to enter a file name which is automatically given the extension *.ini. The ini file contains the configuration parameters and can be edited and re-imported with the command 'File' 'Import'. The available ini files are offered in the standard dialogue 'File open' for importation.

See an example of a ini file in chapter 6.

Remark: Import of a configuration file via command line: As from OPC-Server V.2.3.6.0 with the following command line a configuration ini file (<Export-File.ini>), which has been created before ('File' 'Export'), can be imported into the configurator:

```
OPCConfig.exe /auto <Export-File.ini>
```

By this command the configurator automatically gets opened, the given ini file gets imported and safed and the configurator finally gets reclosed.

3.2.2 Menu 'Edit'

The following commands are at the users disposition depending on whether 'Server', 'PLC' or 'Connection' is marked in the configuration tree or whether a single PLC or a multi PLC configuration is being done:

'Append PLC' Multi PLC, setting for the OPC server
A further controller is added to the configuration, see 3.2.4

'Rename PLC' Multi PLC, setting for the OPC server:
A controller added in the configuration is renamed, see 3.2.4

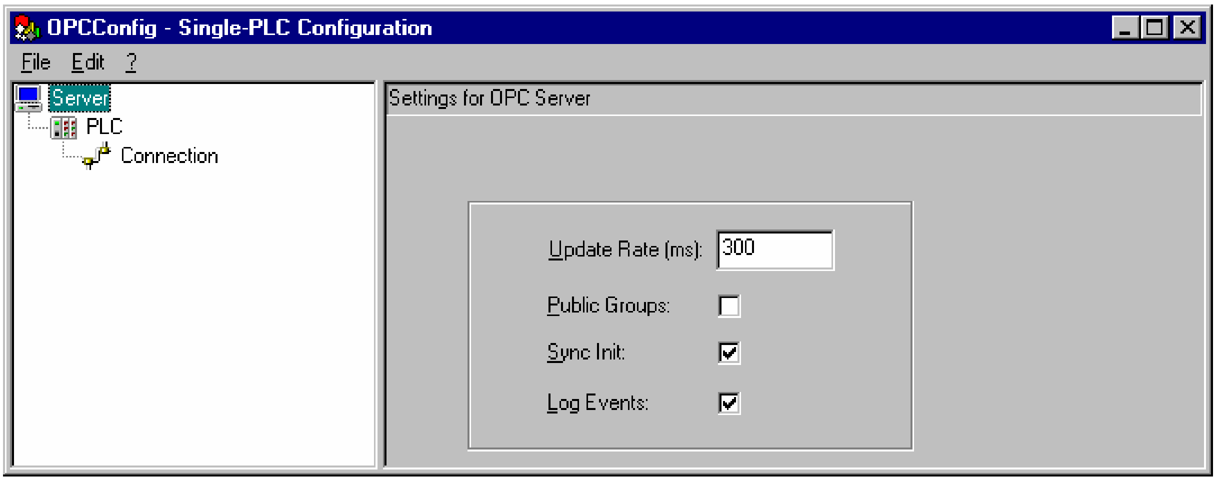
'Delete PLC' Multi PLC, setting for the OPC server:
A controller added in the configuration is deleted, see 3.2.4

'Reset PLC' Single PLC and Multi PLC, setting for PLC:
The configuration parameters set under 'Edit' 'PLC Default Settings' are loaded from the registry.

'PLC Default Settings' Always available:
The dialogue 'PLC Default Settings' opens. Here self-defined data for the PLC configuration can be entered. The dialogue corresponds with the one 'Settings for PLCx' (see 3.2.3). The data placed here is saved in the registry after confirming the dialogue with OK. They can be reloaded for a PLC marked in the configuration tree with the command 'Edit' 'Reset PLC'.

3.2.3 Single PLC Configuration:

After starting the program the following dialogue appears:



Update Rate (ms): 300

Public Groups: □

Sync Init: ☑

Log Events: ☑

A single PLC configuration is set by default this means that only one controller is connected to the server (for multi PLC see 3.2.4). The following possible settings apply for the single PLC configuration:

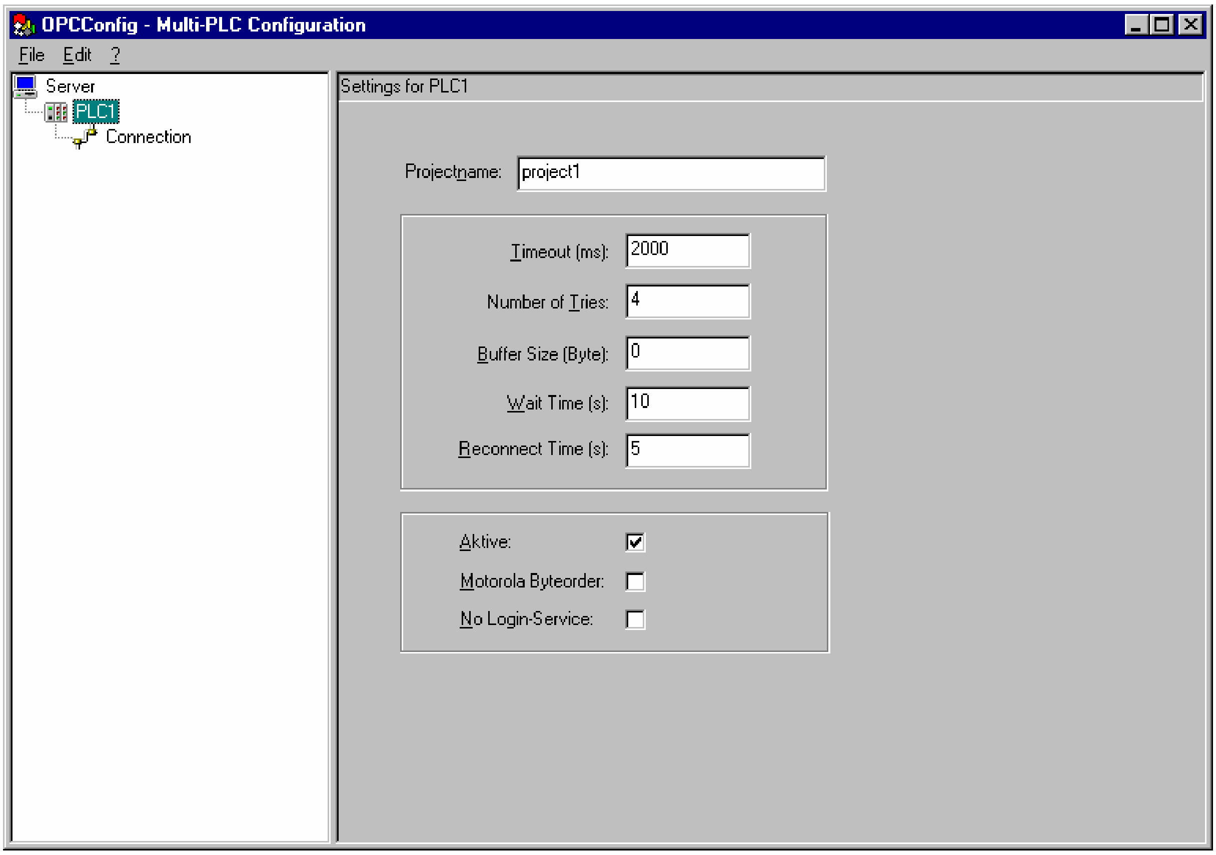
Server, Controller (**PLC**) and **Connection** are displayed in a tree control in the left hand corner of the window. They can be selected with the mouse. The dialogue to configure the entry marked in the tree control appears on the right hand side . The menu 'Edit' is not needed for the single PLC configuration.

If the uppermost entry ‚server' is marked in the tree control the following settings for the server can be made:

- **Update Rate (ms):** Default: 200
  Basic update rate of the OPC-Servers in milliseconds = cycle time with which all item data are read from the controller (this data is written into the cash with which the client communicates with a separately defined update rate). Note: With the status information the server provides also a parameter ‚Bandwidth Usage' giving the ratio of the actual update rate to the configured update rate in percents.

- **Public Groups**: Default: active (tick)
  The OPC server sets up a public group for every IEC module as well as for the global variables.
- **Sync Init**: Default: active (tick)
  Synchronous initialisation: Only when the symbol configuration has been loaded does the OPC server react when starting.
- **Log Events**: Default: active (tick)
  The actions that take place on the OPC server as well as the mistakes that appear are recorded and saved in a log file. This file is set up in the project directory and receives the name OPCServer.log. After shutting down the OPC server the log file can be looked into. The messages of several OPC sessions are lined up in one log file until it reaches the size of 1 MB. When this happens the current date will be added in the filename (OPCServer<date>.log, e.g. OPCServer12.10.2000.log) and the file will be saved. A new log file OPCServer.log will be created.

The following Settings for PLCx can be made on the right hand side of the window if the entry PLCx under the entry server in the tree control is marked:



Settings for PLC1

Projectname: project1

Timeout (ms): 2000

Number of Tries: 4

Buffer Size (Byte): 0

Wait Time (s): 10

Reconnect Time (s): 5

Aktive: ☑

Motorola Byteorder: □

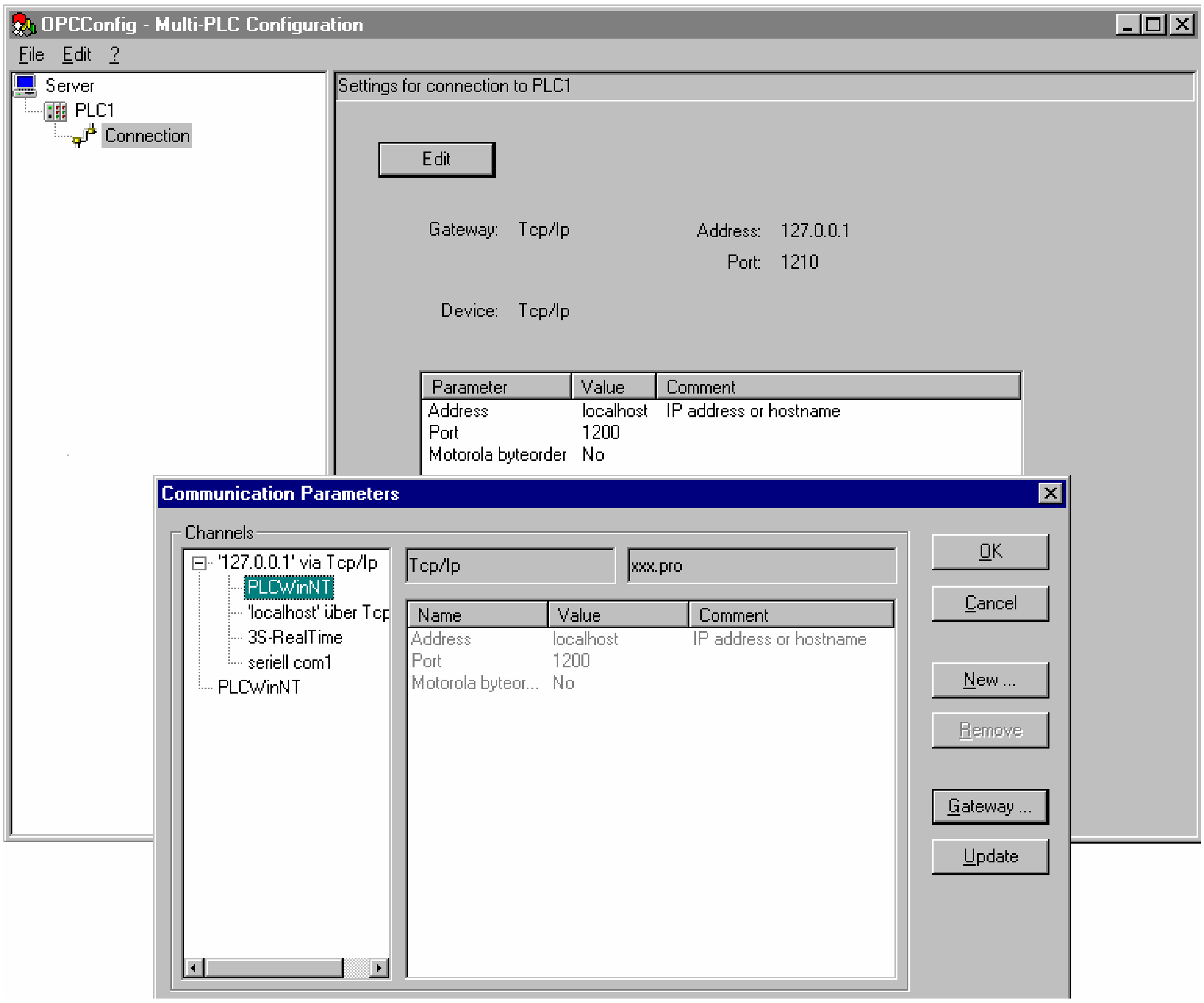
No Login-Service: □

- **Projectname**: Name of the loaded project. This entry is not obligatory when working with single PLC, except you want to make possible an "Offline Load" of the symbols (see note a the end of the chapter 1).
- **Timeout (ms):** 3
  Should the OPC server not receive any feedback on a sent service from the controller after this period of time it will close down automatically.
- **Number of Tries:**
  Number of attempts which the gateway communication driver will make to transfer a block correctly. As soon as the last attempt has failed, a communication error will be produced. (this entry is only relevant for drivers which support a communication via blocks, Level 2).
- **Buffer Size (Byte):** 0
  Size of the communication buffer on target system; if the entry is “0” the attempt is made to

get this information from the device driver. If there is no such information in the device driver it is assumed that the buffer is unlimited.

- **Wait Time (s):** 60
  Time in seconds the OPC server waits until the controller is ready (important during the automatic start of the controller)
- **Reconnect Time (s):** 5
  Time interval, in which the OPC server repeatedly tries to reconnect with the controller via the gateway.
- **Active**: not available (only available in the multi PLC sector)
- **Motorola Byteorder**: not active
  The target system of the project does not use the Motorola byte order (this aspect is to be given special regard when working with 68K, 8051, Power PC !).
- **No Login Service**: active
  This option has to be activated for specific target systems which necessitate the sending of a login service.

If the entry 'Connection' is marked, the 'Settings for the connection to PLCx' can be defined on the right hand side of the window.



Settings for connection to PLC1

Gateway: Tcp/Ip
Address: 127.0.0.1
Port: 1210

Device: Tcp/Ip

Parameter	Value	Comment
Address	localhost	IP address or hostname
Port	1200	
Motorola byteorder	No	

Tcp/Ip
xxx.pro

Name	Value	Comment
Address	localhost	IP address or hostname
Port	1200	
Motorola byteor...	No	

If there already is a configuration this is stated in the specifications behind 'Gateway' and 'Device' as well as in the table. Should the connection to the Gateway not be “local”, specifications concerning the address and the port of the target computer will additionally be mentioned above the table. These specifications have to be equivalent with those made in CoDeSys under 'Online' 'Communication Parameters' for the corresponding project.

If the settings have not jet been specified or if the specified settings are to be changed the dialogue **Communication Parameters** is opened via the button **Edit**. This dialogue is identical with the one available in CoDeSys under Online. For information on how to use this dialogue please see chapter 3.1, step 3 or the user manual for CoDeSys.

3.2.4 Multi PLC Configuration

If more than one controller is to be configured the option 'Single-PLC' in the menu file has to be deactivated.

At first only the entry “Server” will appear on the left hand side of the window. The desired controls are added with the command 'Edit' 'Append PLC' and can then be individually defined along the same lines as the single PLC configuration.

The differences compared to the single PLC configuration:

- The following commands found under the menu **Edit** are additionally available for the PLC currently marked in the tree control:

  'Append PLC' A new entry “PLC” is added. An additional PLC is always added beneath the one marked. The name “PLC<n>” is always given automatically, n starts off as 1 and counts on for every new PLC.

  'Delete PLC The entry is deleted from the overall configuration.

  'Rename PLC' A self defined name can be given to each PLC. To do so first select the button PLC and then the command 'Edit' 'Rename'. An editor field into which a name can be entered will then open.

- The name of the project which was loaded onto this controller via the Gateway must be entered behind **Projectname** in the dialogue 'Settings for PLCx'.
- By turning on or off the option **Active** in the dialogue 'Settings for PLCx' it can be defined whether the controller is to be considered by the OPC server or not.
- **Please regard: OPC servers older than version V2.3.8.2 cannot communicate over different Gateways!**

3.2.5 Registry entries (Example)

[HKEY_CURRENT_USER\SOFTWARE\3S-Smart Software Solutions GmbH\CoDeSys OPC\Config]
* corresponds with the settings made in the OPC configurator in dialog 'Settings for OPC Server' (see Chapter 3.2.3) *

"Check config"="yes"

"LogEvents"="yes"

"Multi"="no"

"Updaterate"="0x0000012c (300)"

"Wait for target"="0x0000000a (10)"

"Sync startup"="yes"

"Public groups"="yes"

"Reconnect Interval"="0x00000005 (5)"

"Notify on quality change"="yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions GmbH\CoDeSys OPC\PLCDefaults]   * corresponds with the data entered under 'Edit' 'PLC Default Settings' in the OPC configurator (see Chapter 3.2.2) *

```
"Motorola"="false"

"Timeout"="2000"

"Tries"="4"

"Waittime"="10"

"Reconnecttime"=5"

"Nologin"="true"

"Buffersize"="1"

"Activ"="true"
```

[HKEY_CURRENT_USER\Software\3S-Smart Software Solutions GmbH\CoDeSys OPC\Connections]
* corresponds with the settings made in the dialog 'Settings for PLC' in the OPC Configurator see Chapter 3.2.3 and 3.2.4) *

"Project0"=hex (Current communication parameters for the OPC server with project identification)

```
"Motorola0"="No"

"Timeout0"="0x000007d0 (2000)"

"Tries0"="0x00000003 (3)"

"NoLogin0"="No"

"Buffersize"="0x00000000 (0)"
```

3.3 Starting CoDeSysOPC.exe

The OPC server (+ Gateway server) is automatically started by the operating system as soon as the client sets up a connection. It does not have to be started explicitly. The client is connected automatically with the running server if the OPC server was already started manually.

As soon as all the clients have broken off their connection to the server it shuts down automatically.

The following symbol appears on the status bar in the right hand corner of the screen as long as the OPC server is running: OPC

If you click on this symbol with the right mouse button you will receive information on the version of the OPC server under 'About'.

Starting in the test mode:

The connection between the OPC server and the client can also be tested if there is no controller connected. For this purpose the OPC server has to be started explicitly with the following command:

```
CoDeSysOPC.exe /TestMode
```

It then operates in the test mode and automatically generates an array of test items with the help of which the communication with the client can be tested.

The OPC symbol also appears in the status bar in this simulation mode.

4 Connection to an OPC Server on a different computer / DCOMCNFG.EXE

A connection to an OPC server on another computer can be established using the DCOMCNFG.EXE which can be found in the root directory. To do so please switch onto 'OPC-Server for CoDeSys..' in the index card 'Applications' and enter the desired computer under 'Properties' 'Location' 'Execute application on the following computer'.

In order for this connection to function there must also be an OPC server running on the local computer.

5 Service Packs for Windows 95 / Windows 98

The following service packs have to be installed in order to run the current OPC and Gateway versions under Windows 95 or 98.

DCOM98CFG.EXE

DCOM98.EXE

6 Example of an ini file for the configuration of the OPC-Server

The ini file of the OPC server configuration can be exported from the OPCConfig.exe and then set up and edited as a text file and then re-imported into OPCConfig.

The following is an example of the entries for the corresponding parameters which can be set up in OPCConfig:

Entry	**Meaning**
[Server]	Settings for the OPC server
updaterate=200	Update Rate
publicgroups=0	Public Groups
syncinit=1	Sync Init
PLCs=2	Connection to two controllers is planned
PLC0=PLC_A	Name of the first controller: PLC_A
PLC1=PLC_B	Name of the second controller: PLC_B
[PLC:PLC_A]	Settings for PLC_A
active=1	Active
motorola=0	Motorola Byteorder
nologin=1	Login Service
timeout=2000	Timeout
tries=3	Number of tries
waittime=10	Waiting time
reconnecttime=5	Reconnecttime
buffersize=3	Buffer size
project=opc1.pro	Project name
gateway=Tcp/Ip	Settings for the connection to PLC_A (Connection client to local Gateway over TCP/IP)
gatewayaddress=localhost	
gatewayport=1210	
device=Tcp/Ip	Parameter for connection to the target computer where the controller PLC1 is running (Dialogue communication parameters)
parameters=3	
parameter0=Address	
value0=127.0.2.1	
parameter1=Port	
value1=1200	
parameter2=Motorola byteorder	
value2=No	

Entry	Meaning
[PLC:PLC_B]	Settings for PLC_B
active=1	See above
motorola=0	
nologin=0	
timeout=2000	
tries=3	
waittime=10	
reconnecttime=5	
buffersize=0	
project=opc_test_s.pro	
gateway=Tcp/Ip	Connection to the gateway on the local computer over TCP/IP (has to correspond to the one already used for PLC-A!)
gatewayaddress=localhost	
gatewayport=1210	
device=Tcp/Ip	Parameter for connection to target computer where controller PLC2 is running
parameters=3	
parameter0=Address	
value0=Sabine	
parameter1=Port	
value1=1200	
parameter2=Motorola byteorder	
value2=No	

7 Error Diagnosis

7.1 Means for error diagnosis:

1. OPC-Server log-file
2. Second (test-)client for checking error behaviour
3. Gateway Inspector with trace and logging (for sophisticated communication errors)

7.2 Brief checklist

Basically check the following points:

4. Gateway installed and running ? Symbol in task bar in the right hand corner active ?
5. Installation and Registration of the OPC-Server OK ?
   takes place automatically by executing: „CoDeSysOPC /Install" (Installation + Registration) or .
   „CoDeSysOPC /RegServer" (only registration)
6. Preparation of the project or settings in the CoDeSys programming system OK?
   - Project options: 'Dump symbol entries' selected?
   - communication parameters: selected Gateway channel OK?
   - Project saved, translated, download executed?
7. Should you want to use OPC on another computer (only possible, if the OPC server is also operating locally and the user is not additionally logged into the CoDeSys project on the other computer):
   Connect to the other computer via DCOMCNFG.EXE (in the system directory):
   - select OPC server in 'Applications'
   - enter desired computer under 'Properties' 'Location' 'Execute application on the following computer'
8. Check the connection and the server settings: OPCConfig.exe:
   - connection: (Entry has to correspond with the valid Gateway set up in the CoDeSys communication parameters)
   - normal server settings: Motorola Byteorder: no (Attention when using 68K, 8051, PowerPC), set up public groups, synchronous initialisation, Notification on quality change, Update Rate 200ms, waiting time target system 60s, Reconnect-Interval 5s, communication timeout 3s, buffer size: 0 or fitting for target system?
   - Save -> new configuration becomes effective when new connection to the OPC server is established
9. Check, whether there are symbol entries disposable: DiagnosticOPCClient.exe (this diagnostic client is not part of the installation but can be delivered on demand)
   Establish connection (Connect OPC-Server) -> click on Private Groups in right hand side window with right mouse button, confirm dialogue with OK, click on the group entry which appears now with right mouse button -> Add all items -> all generated symbol entries (variables) should appear

7.3 Particular error cases

Error: No connection to the PLC (Status NO_CONFIG):

To check:

- PLC is running?
- Communication parameters in OPC configuration (OPCconfig.exe) ok?:
  - Valid communication protocol on PLC?
  - PLC available on the configured communication channel (test by „ping“ on TCP/IP)
  - Cable to PLC ok?
  - In case of using a remote gateway, ist this gateway started?

Error: No symbols available (Status NO_CONFIG):

- Is option „Symbol configuration“ in the CoDeSys programming system (Project/Options) activated and are variables selected there? (check *.sym-file)
- Is a symbol file (*.sdb) available in the Gateway-Files directory or on the PLC?
- Does the project ID of the project in the programming system match with that used in the symbol file? (Maybe a download has been done with a new project version without also creating a corresponding bootproject)

Error: Items Quality „BAD“

Possible reasons:

- Communication error with PLC while OPC-Server is running
- Communication error during download of a new program on the PLC
- Group or Item inactive

Please regard:

- **The OPC server shuts down automatically when all clients have cut the connection to the server.**
- **The OPC server appears as a process in the task manager.**

Change History

Version	Description	Editor	Date
V1.0	Issued	AH, MN	20.12.00
V1.1	Multi-PLC Configuration not only via TCP/IP, but also local	AH, MN	04.10.02
V1.2	Chapter 3.2.3: "Number of Tries" Description corrected	AH, MN	28.11.02
V1.3	Chapter 7: Item 5: "OPCConfig.exe", Item6: Diagnostic Client only available on demand	AH, MN	25.03.04
V1.4	Chapter 1: Note concerning "Offline Load" of the symbol file; reference on documentation for symbol file; Chapter 3.2.3: Remark added at 'Projectname'	AH, MN	05.10.04
V1.5	Extension and Release: Bandwidth usage, Chapter 3.2.3; formal: adaption to new document template	MN	22.02.05
V1.6	Erweiterung Kap. 3.2 und 3.2.1 (Konfigurationsdatei über Kommandozeile importieren)	MN	15.07.2005
V1.6	Review OK	AH	10.08.2005
V1.6	Release, (OPC V2.3.9.1)	MN	10.08.2005
V1.7	Chap.7, Error diagnosis extended (OPC V2.3.9.2)	MN	21.11.2005
V1.7	Release (OPC V2.3.9.2, with CoDeSys V2.3.6.0)	MN	31.01.2006