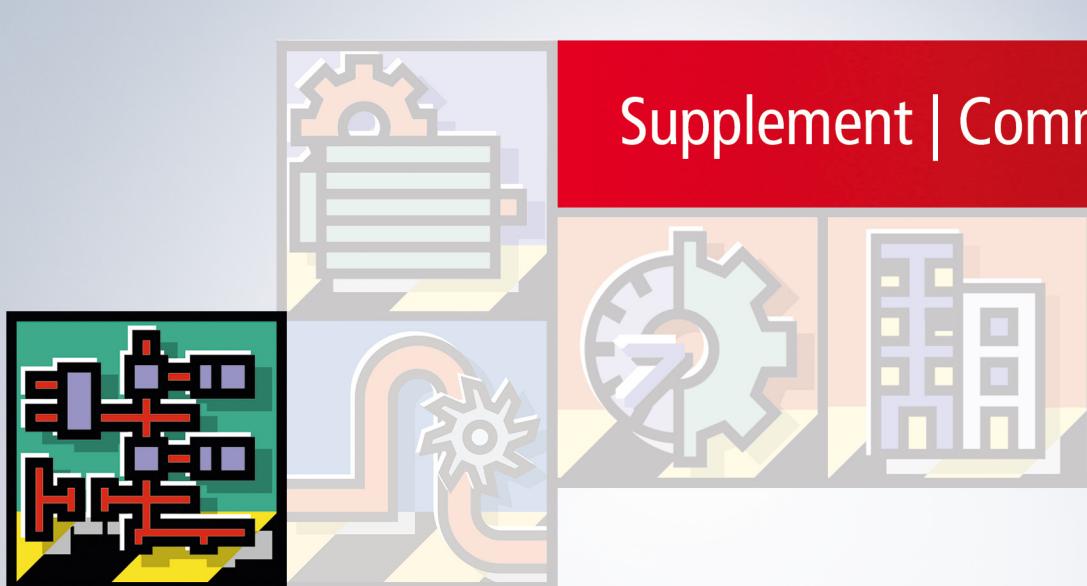


Manual | EN

TS6120

TwinCAT 2 | OPC Server





# Table of contents

<b>1 Foreword .....</b>	<b>5</b>
1.1 Notes on the documentation .....	5
1.2 For your safety .....	6
1.3 Notes on information security .....	7
<b>2 OPC General .....</b>	<b>8</b>
<b>3 Overview .....</b>	<b>9</b>
<b>4 Installation .....</b>	<b>10</b>
<b>5 Licensing .....</b>	<b>12</b>
<b>6 Configuration.....</b>	<b>13</b>
6.1 OPC DA Server .....	13
6.1.1 General .....	13
6.1.2 Data Access .....	18
6.1.3 Conversion .....	25
6.1.4 Simulation .....	28
6.1.5 Item Properties.....	31
6.1.6 Data exchange over the network .....	32
6.2 OPC XML-DA Server .....	34
6.2.1 Configuration OPC XML-DA on Windows XP .....	36
6.2.2 Configuration OPC XML-DA on Windows 7 .....	40
6.2.3 Status information .....	45
6.2.4 TwinCAT OPC Server CE : Optimization .....	46
6.2.5 OPC Client (Diagnostic & Recipe) .....	47
<b>7 OPC Client sample code.....</b>	<b>61</b>
7.1 OPC .NET client : API sample for read / write .....	61
7.2 OPC .NET client : API sample for subscriptions .....	62
7.3 OPC client : AJAX (Asynchronous Java XML) .....	63
<b>8 Appendix.....</b>	<b>64</b>
8.1 OPC Compliance Certification .....	64
8.2 DCOM .....	65
8.2.1 Prerequisites .....	66
8.2.2 Client .....	69
8.2.3 Server .....	76
8.2.4 DCOM Permission .....	82



# 1 Foreword

## 1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

### Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

### Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

### Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702  
with corresponding applications or registrations in various other countries.



EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

## 1.2 For your safety

### Safety regulations

Read the following explanations for your safety.

Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

### Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

### Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

### Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

### Personal injury warnings

#### DANGER

Hazard with high risk of death or serious injury.

#### WARNING

Hazard with medium risk of death or serious injury.

#### CAUTION

There is a low-risk hazard that could result in medium or minor injury.

### Warning of damage to property or environment

#### NOTICE

The environment, equipment, or data may be damaged.

### Information on handling the product



This information includes, for example:  
recommendations for action, assistance or further information on the product.

## 1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our <https://www.beckhoff.com/secguide>.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at <https://www.beckhoff.com/secinfo>.

## 2 OPC General

### OPC Definition

OPC stands for "Openness Productivity and Collaboration" and describes an initiative for standardisation of data exchange in automation.

Beckhoff is a active member of the OPC Foundation. (More information about OPC may be obtained under [www.opcfoundation.org](http://www.opcfoundation.org) )

### OPC Interface

Formerly, applications such as operating and monitoring software were forced to use the differing access procedures of each controller manufacturer.

OPC is the specification of a universal software interface that is totally independent of the manufacturer. This opens up the possibility of gaining access in a unified manner to process data from any manufacturer.

### OPC Server

An OPC server is a software component that offers a specified OPC interface to other applications.

### 3 Overview

- OPC Specification 1.0a and 2.0
- Free choice of TwinCAT devices (ADS devices): PLC / IO / NC / CAM / ...
- Access via TwinCAT PLC / TwinCAT I/O or bus controller to all the fieldbusses supported by TwinCAT (Lightbus / Profibus DP / Interbus-S / CANopen / DeviceNet / ...)
- Access to TwinCAT variables by address / by name / by generic ADS IndexGroup and IndexOffset
- Optimal access to data from the OPC server on TwinCAT variables
- Simulation mode (sinus, ramp, random) for variables  
(i.e. testing an OPC client/server communication without access to physical I/O)
- Easy, one-time only configuration by XML

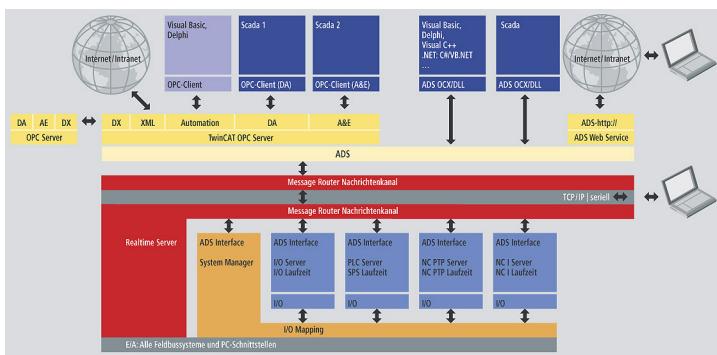
#### Features of the evaluation version

The evaluation version is a fully functional version with a 30 days time limitation. The OPC server then switches to its internal simulation mode. All the process values are then simulated, and there is no longer any access to the ADS devices.

The time limit in the evaluation version can be removed by purchasing a [TwinCAT OPC software licence](#) [►12], and by the entry of a valid registration key.

#### Integration into the TwinCAT system

The OPC server creates the connection to the TwinCAT system as a gateway via ADS, and offers OPC clients the specified OPC-DA / AE interface.



#### System requirements

- Microsoft Windows NT 4 / Windows 2000 / XP
- TwinCAT v2.6 (or higher)

Please use the latest Service Packs.

#### What is included

The installation version includes

- TwinCAT OPC server (DataAccess and Alarm & Events)
- Documentation
- OPC Test client (for local access to the OPC server)

## 4 Installation

This part of the documentation gives a step-by-step explanation of the TwinCAT OPC setup process for Windows XP based operating systems. The following topics are part of this document:

- Downloading the setup file
- Starting the installation

### Downloading the setup file

Like many other TwinCAT Supplement products, TwinCAT OPC is available for download via the Beckhoff homepage. The download represents the most current version, which can be licensed either as a 30-Day Demo or as a full version. To download the setup file, please perform the following steps:

- Navigate on [www.beckhoff.com](http://www.beckhoff.com) to "Download" TwinCAT Supplement and select "TS6120 TwinCAT OPC Server".
- Select "Start Download", "Download as Guest".
- (Optional) Transfer the downloaded file to the TwinCAT runtime system, where you would like to install the Supplement.

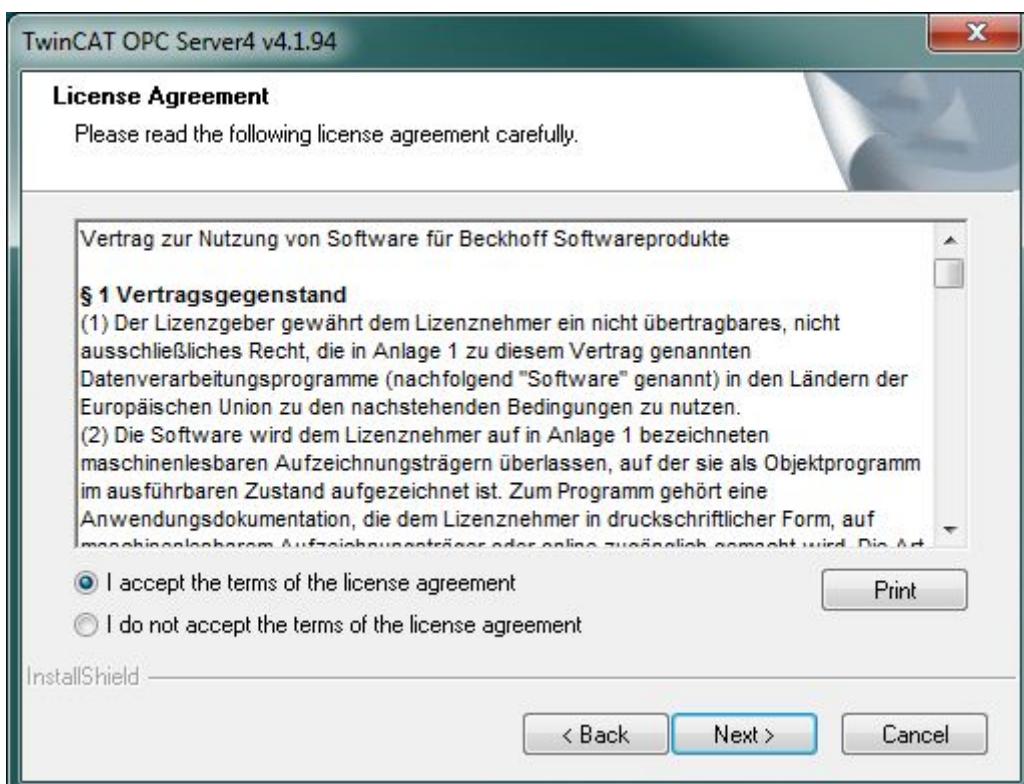
### Starting the installation



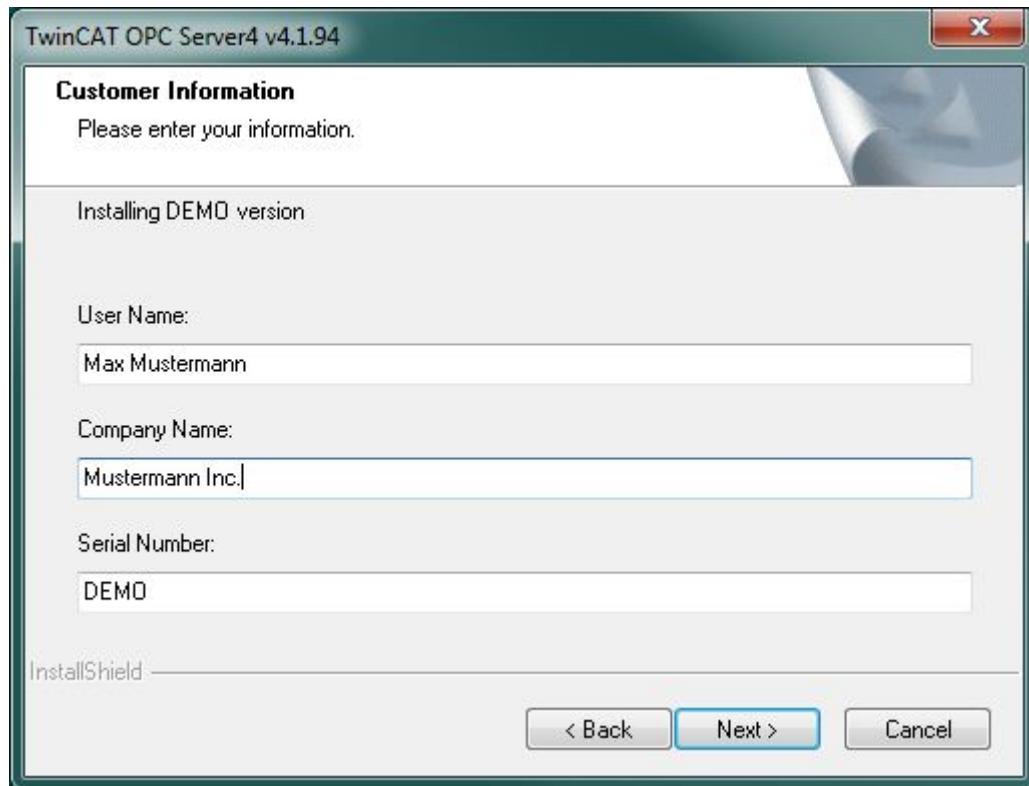
Under Windows 7 32-bit/64-bit, please start the installation with "Run as Administrator" by right-clicking the setup file and selecting the corresponding option in the context menu.

To install the Supplement, please perform the following steps:

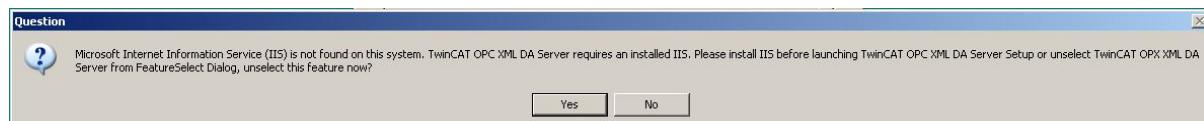
- Double-click the downloaded setup file "**TcOpcSvr410.exe**".
- Select an installation language
- Click on "Next" and accept the license agreement



- Enter your user information. All fields are mandatory. If you would like to install a 30-Day demo, please enter "DEMO" as a license key.



- The currently available Setup installs by default **all** TwinCAT OPC components. This also includes the feature TwinCAT OPC XML DA, which needs the IIS Server. However, if this service is not available, the Setup generates a corresponding warning message. If you don't want to use OPC XML DA, you can safely accept this error message by clicking on "Yes". Otherwise, select "No" and install IIS before starting the Setup again.



## 5 Licensing

The following information applies to the OPC server running on a PC platform. This information is not relevant when running on a CE platform.

TwinCAT OPC can be used free of charge as an evaluation version without a registration key for 30 days. Registration is necessary for general use. A licence for use of the software is obtained with this registration.

### Registration instructions

#### During the installation

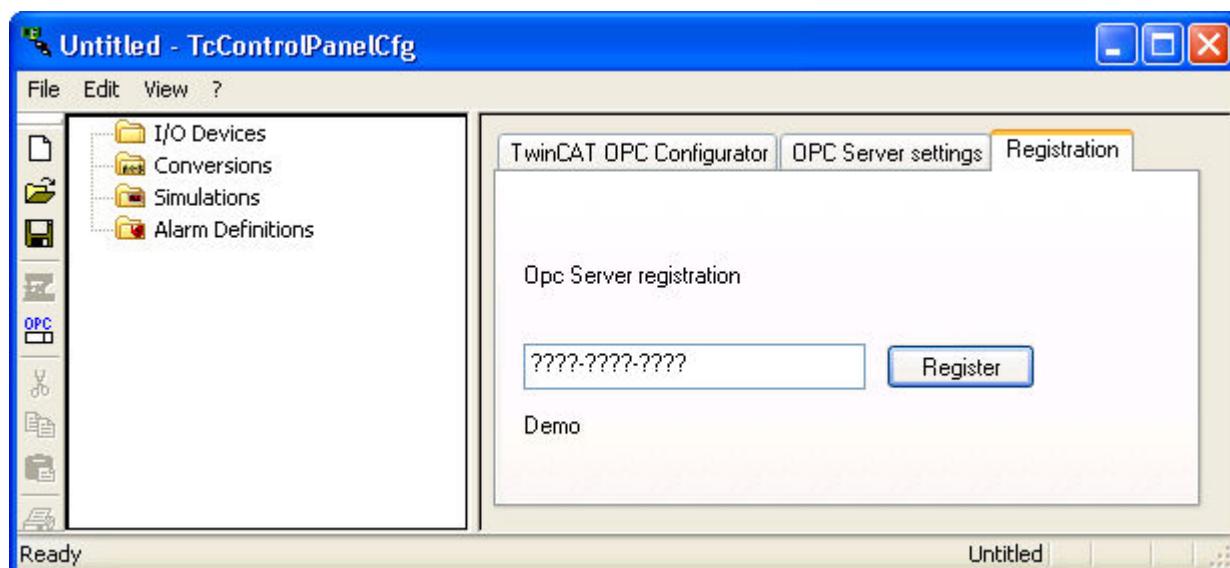
When installing TwinCAT OPC, the registration key (e.g. K123-4567-8901) can be entered into the appropriate field in the setup program. This removes the time restrictions placed on the evaluation version.

If no key is available at the time of installation, please enter "DEMO" as the key. The time-limited (30 days) evaluation version of the TwinCAT OPC will be installed.

#### Subsequent registration of an evaluation version

Start the program TwinCAT OPC Configurator TcOpcCfg.exe and select tab "**Registration**".

Please enter a valid registration number into the appropriate field and press "**Register**". The time limit is removed from the evaluation version.



## 6 Configuration

### 6.1 OPC DA Server

#### 6.1.1 General

##### 6.1.1.1 TwinCAT OPC Server : Optimisations

In case of trouble with OPC-Server performance please check following steps :

###### 1. Minimise OPC-Namespace in TwinCAT-OPC-Server :

- OPC-Server should provide required variables in his namespace which are requested by an opc-client.

It makes no sense to provide 700.000 PLC variables in the OPC-namespace but the opc-client just exchange values from 10.000 variables.

The size of OPC-namespace will cause a longer opc-server-startup-time and will cause a high memory consumption .

###### Option 1 :

Export just part of PLC variables required for OPC :

In the TwinCAT PLC programming environment, select "Project"->"Option"->"Symbolconfiguration". Activate "Create Symbol", and use the mouse to select under "Configure Symbolfile..." the areas that you want to export (e.g. POU's and global variables).

After compiling the PLC-project the new symbolfile "<yourProjectName>.sym" will be in the same folder as your plc-project.

###### REMARK:

- Each compiling of the PLC-project will automatically update the symbolfile.
- The OPC-Server imports structure and array-elements out of \*.SYM files.

###### Option 2 :

Mark PLC-variables in the PLC-code to be relevant for OPC

This way is based on the TwinCAT export file \*.tpy.

Detailed info [► 46]

###### 2. OPC-Server : Running as EXE or DLL ?

- Arguments for working with EXE :
  - One single OPC-client will access data from TwinCAT-OPC-Server
  - The OPC-Client is installed on a different PC , the network has to be crossed via DCOM.
- Arguments for working with DLL :
  - More than one OPC-Client will communicate with the OPC-Server : Instead one single OPC-Server shares his CPU time to multiple opc-clients, the OPC-Server should work as a DLL : As a result each OPC-Client will have "his personal" instance of the OPC-Server with each opc-server having a separate ADS-communication channel into the ADS-device like the PLC.

###### 3. OPC-Server via DCOM

- The installation of BECKHOFF OPC-components consist of two parts : The native OPC-Server from Beckhoff and so called "OPC common components" which comes from the OPC-Foundation.

The installation of BECKHOFF-OPC-products have been redesigned to integrate the official OPC-components and install them in the recommended way.

Change of installation : 07/2004, new installation for TwinCAT OPC-Server V4.1.0.37

### 6.1.1.2        OPC Server as EXE / DLL (out-of-process / in-process)

The following information applies to the OPC server running on a PC platform. This information is not relevant when running on a CE platform

TwinCAT OPC V4.1 is available in the form of an "out-of-process" server (with EXE as the file extension) and as an "in-process" server (with DLL as the file extension).

#### ProgID

The OPC client requires the ProgID to startup communication with TwinCAT-OPC-Server. For TwinCAT-OPC-Server V4.1 this ProgID is "**BECKHOFF.TwinCATOpcServerDA**".

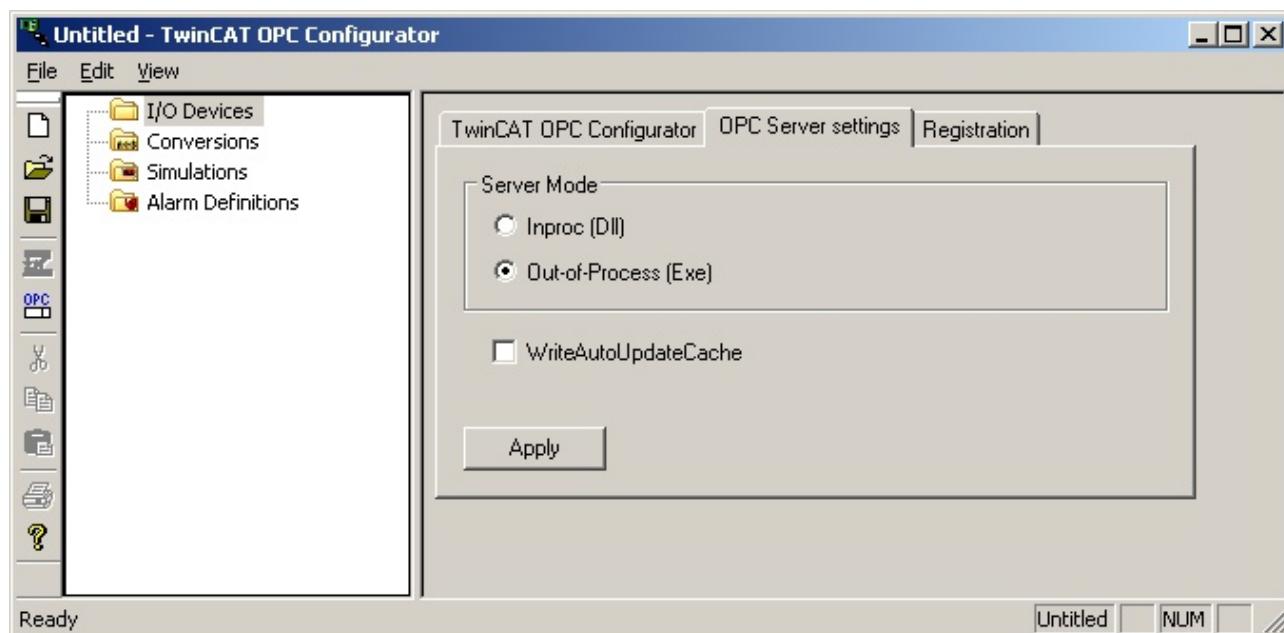
**The ProgID is identical for both types, the EXE or the DLL version.**

**Diagnose : Which type (EXE or DLL) is active ?**

Start the TwinCAT-OPC-Configurator "Start - All Programs - TwinCAT System - TwinCAT OPC - TwinCAT OPC Configurator"



Select the tab "OPC Server settings". See current active type of "Server Mode" .



#### Activating the TwinCAT OPC server as an in-process server (DLL)

Select "Server Mode" "Inproc (DLL)" and press "Apply".

Result :

The TwinCAT OPC server now runs as a DLL in the process space of the respective OPC client. The TwinCAT OPC server is therefore not visible as an independent process in the task manager.

The main advantage of this version is seen when a number of OPC clients are working at the same time with TwinCAT OPC server. In that case, each OPC client is given its own OPC server for its "personal use". Each OPC client then has its own independently operating OPC channel to the TwinCAT devices. Most advantage will be noticed for write-requests via OPC-Server into ADS-device.

### NOTICE

When using Windows 7 and a OPC-Client which runs as a Windows Service, you need to disable User Account Control (UAC) for the OPC-Server to run properly.

#### **Activating the TwinCAT OPC server as an out-of-process server (EXE)**

Select "Server Mode" "**Out-of-Process (EXE)**" and press "**Apply**".

Result :

The TwinCAT OPC server now operates as an EXE program, and each OPC client works with the single, shared instance of the OPC server.

In this version the TwinCAT OPC server is visible as an independent process in the task manager.

If a number of OPC clients are working at the same time with the TwinCAT OPC server, all the queries are handled by a single instance of the OPC server.

This single instance eliminates or minimize double requests of variables for one single ADS device.

#### **DCOM:**

1. We do not advise the use of DCOM connections.

2. If, however, a DCOM connection is essential, it is implemented with the type "Out-of-process" OPC server.

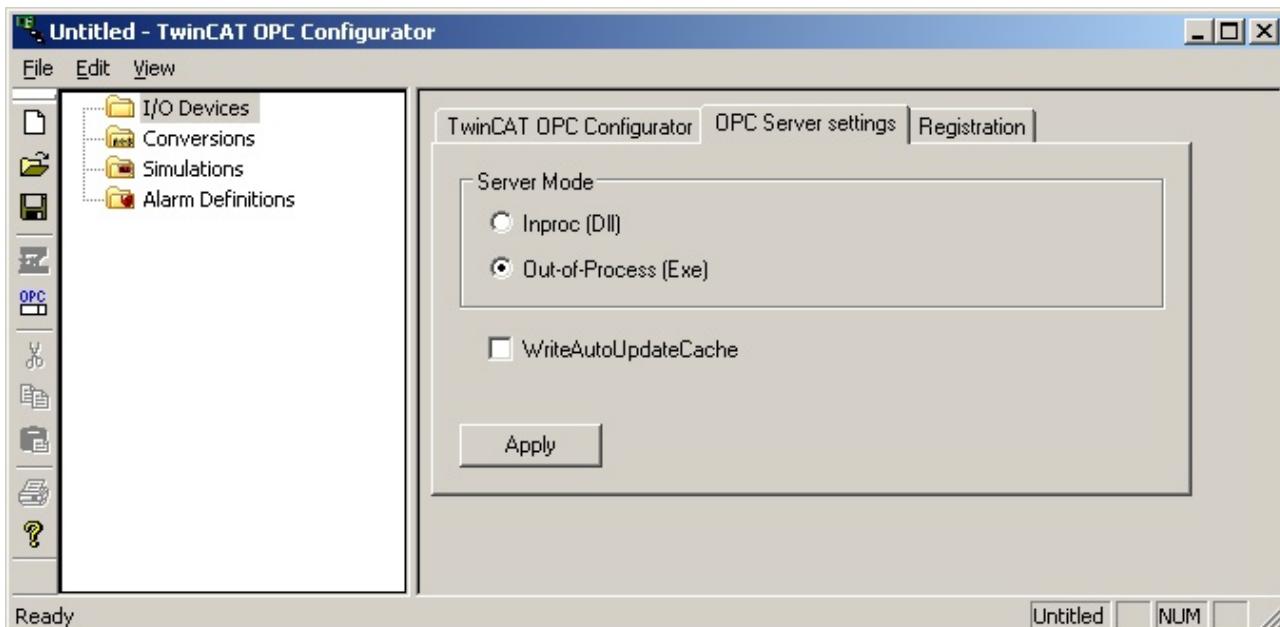
#### **6.1.1.3 Automatic Cache Update**

**The following information applies to the OPC server running on a PC platform. This information is not relevant when running on a CE platform**

This option causes the OPC Server to automatically update its cache after write operations.

According to OPC Specification an OPC Server shall not update its cache automatically after writing a new value. After a successful write operation the cached value remains the same and is not updated until the next scan cycle. As this behaviour causes problems in some applications the Beckhoff OPC Server provides an option to change this behaviour.

This option is turned off by default, so that the Beckhoff OPC Server operates correctly according to OPC Specification. To change the cache update behaviour start the TwinCAT OPC Configurator, Open the "OPC Server Settings" tab and check "AutoUpdateWriteCache"



## 6.1.1.4 Description of the TwinCAT-OPC-Server XML configuration

### Schemes

Namespace

Devices

Device <see [Device-Parameters \[► 16\]](#)>

Device <see [Device-Parameters \[► 16\]](#)>

SimulationSignals

Simulation <see [Signal-Parameters \[► 17\]](#)>

...

Conversions

Conversion <see [Conversion-Parameters \[► 17\]](#)>

...

### Device-Parameters

Device-Parameters

Type	Req./Opt.	Description
Name	Req	Name for device like "PLC1". OPC-Client browse this name.
Description	Optional	
AdsPort	Req	Number of ADS-PortID, like 801 for first PLC-runtime system.
AdsNetId	Optional	specific ADS-NetId, like "174.12.15.45.1.1" <b>If not specified or like "0.0.0.0.0", the OPC-Server will always communicate to AdsNetId of local ADS-router.</b>
AdsTimeout	Optional	ADS timeout in ms for this ADS device
AdsTimeSuspend	Optional	ADS suspend time in ms for this ADS device, if the ads communication fails.
AutoCfg	Optional	0 : do not include symbol-file defined under <AutoCfgSymFile> 1 : Include sym-file of TwinCAT-PLC defined under <AutoCfgSymFile> 2 : Include sym-file of TwinCAT-BCxxxx-project defined under <AutoCfgSymFile> 5 : Upload symbolic from ADS-device 7 : Include tpy-file of TwinCAT28-project defined under <AutoCfgSymFile> 8 : Include tpy-file of TwinCAT28-project defined under <AutoCfgSymFile>, but import only symbols with symbol-property "OPC=1"

Type	Req./Opt.	Description
AutoCfgSymFile	Optional	full path and name of symbol-file to be included like "C:\Test\demo.sym" or "\User1\Test\demo.sym" or "C:\User1\Test\demo.tpy"

## Signal-Parameters

Signal-Parameters

Type	Req./Opt.	Description
SignalID	Req	unique ID-number which identifies this simulation-signal
SignalDesc	Optional	
SignalType	Req	<p><b>0 : Read Count</b>  <b>1 : Write Count</b>  <b>2 : Random</b>  <b>3 : Ramp</b>  <b>4 : Sine</b>  <b>5 : Square</b>  <b>6 : Triangle</b>  <b>7 : Step</b>  <b>8 : Reserver</b>  <b>9 : Step</b></p> <p><b>Read Count</b> : incremented by one every time when the item is read</p> <p><b>Write Count</b> : incremented by one every time when the item is written</p> <p><b>Random</b>: generates random value within the Amplitude range starting with Position</p> <p><b>Ramp, Sine, Square, Triangle, Step</b>: (periodical signals) Their time behavior is influenced by Period and Phase parameters. Period specifies the signal frequency, while Phase moves the signal origin on the time axis</p> <p>Square and Triangle signal types have one more parameter: Ratio. Ratio defines Triangle signal steepness, or Square signal H/L proportions. NumSteps parameter of the Step signal defines a number of steps signal amplitude will be divided into.</p>
NumSteps	Optional, depends on <SignalType>	
Amplitude	Optional, depends on <SignalType>	
Period	Optional, depends on <SignalType>	
Phase	Optional, depends on <SignalType>	
Position	Optional, depends on <SignalType>	
Ratio	Optional, depends on <SignalType>	

## Conversion-Parameters

Conversion-Parameters

Type	Req./Opt.	Description
ConversionID	Req	unique ID-number which identifies this conversion
ConversionDesc	Optional	
ConversionType	Req	0 : NoConversion 1 : LinearConversion
Clamping	Optional	0 : No clamping 1 : Clamp on EU 2 : Clamp as specified  If clamping is active, the data value will be limited to its High clamp/EU value, when it exceeds the upper limit, and similarly with Low clamp parameter.
HighClamp	Optional, depends on <ConversionType>/<Clamping>	1.0 (Default)
LowClamp	Optional, depends on <ConversionType>/<Clamping>	0.0 (Default)
HighEU	Optional, depends on <ConversionType>/<Clamping>	engineering unit (client scale) 1.0 (Default)
LowEU	Optional, depends on <ConversionType>/<Clamping>	engineering unit (client scale) 0.0 (Default)
HighIR	Optional, depends on <ConversionType>/<Clamping>	instrument range (device scale) 10000 (Default)
LowIR	Optional, depends on <ConversionType>/<Clamping>	instrument range (device scale) 0 (Default)

## 6.1.2 Data Access

An OPC server represents a standardised interface for the management of process data. The process data available in the TwinCAT system must therefore be known to the OPC server, or must be made known to it at the time of configuration.

To represent this "hierarchical process space" clearly, the "Devices" are subdivided into subsidiary items.

The OPC client can browse through this representation and use it for the server's actual configuration. The TwinCAT OPC server supports the optional browser OPC interface.



This hierarchical display is not to be confused with the configuration of the OPC server that exists at run-time. The run-time configuration of the OPC server, i.e. the creation of groups, specification of the refresh time, the insertion of tags etc. is performed dynamically by the OPC client.

### General

The advantage of configuration via XML with links to the symbols of the ADS devices is that this configuration only must be carried out once. Even if, for instance, the PLC project changes (e.g. if variables are deleted, or if new variables are defined) it is not necessary to change the configuration in the OPC server: each time the OPC service starts, the namespace of the OPC server is correctly built up.

### Device Definition

A device represents a TwinCAT ADS device (e.g. a PLC run-time system) that is uniquely identified by the <AdsAmsServerNetID> and the <AdsAmsPortID>.

Further information about the concept of devices in TwinCAT can be found in the general TwinCAT ADS documentation.

## Configuration of the TwinCAT OPC server

- <https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162179979/.htm>: Configuration by variable import from TwinCAT PLC control
- [Receiving Data from the TwinCAT I/O task ▶ 22](#): Configuration by variable upload from the TwinCAT I/O task

## Configuration of the TwinCAT OPC Server CE

In general, the configuration of the TwinCAT OPC Server CE is like the configuration of the TwinCAT OPC Server. The most important information is given here:

- Set the active OPC XML configuration:  
PC: The TcOpcStartup.exe tool saves the current OPC XML configuration file in the registry  
CE: By default, the file "\hard disk\TwinCAT\OPC\CurrentConfigOpc.xml" is entered in the registry as the current OPC XML configuration file
- The OPC XML file for CE works with the same XML scheme
- The OPC XML file must allow for the fact that paths are written using a different notation under CE:  
PC example: <AutoCfgFile>C:\Project123\Test.tpy</AutoCfgFile>  
Example CE: <AutoCfgFile>\hard disk\project\Test.tpy</AutoCfgFile>

### 6.1.2.1 Receiving data from the TwinCAT PLC 2.8 and higher

#### 1st Step TwinCAT PLC

##### TwinCAT 2.8 (or higher):

With TwinCAT 2.8 or higher an TwinCAT OPC-Server 4.1.0.79 (or higher) the OPC-configuration has become very easy and has to be done just once.

After the PLC project has been compiled, there is always a file in the PLC project directory with the name structured as "<project name>.tpy".

- The \*.tpy file is automatically replaced each time the PLC project is compiled.  
In the PLC Control navigate to "Projects.. Options... TwinCAT" and select "Download symbol description". With each compile the <projectname>.typ file is now transported to target system and stored as e.g. "\TwinCAT\Boot\CurrentPLC\_1.tpy" for first PLC-runtime system, or "\TwinCAT\Boot\CurrentPLC\_2.tpy" for second PLC-runtime system etc.
- Which PLC variable is available via OPC ?  
With adding simple information done by the declaration of the PLC-variable you can mark a PLC variable to be known by the OPC-Server. This helps to reduce the OPC-namespace.  
The OPC-DA Server will automatically connect to PLC runtime and provide PLC-variables which are marked for OPC-DA within the PLC-Control.  
Mark the PLC-variables with a comment to enable them being accessed by OPC-DA Server.

Extra comments in PLC-IEC1131 Editor will modify the configuration of opc-server:

Sample1:

PLC variable "bMemAlarm"1 will not be available via OPC-DA-Server

```
bMemFlag1      AT%MX10.0   :  BOOL;  /* (OPC:1 : available for OPC-clients) */
bMemFlag2      AT%MX10.1   :  BOOL;  /* (OPC:1 : available for OPC-clients) */
bMemAlarm1     AT%MX10.2   :  BOOL;
bMemAlarm2     AT%MX10.3   :  BOOL;  /* (OPC:1 : available for OPC-clients) */
iReadOnly       :  INT;    /* (OPC:1 : available for OPC-clients)
                           (OPC_PROP[0005]:1:OPC_PROP_RIGHTS, here ReadOnly) */
```

Sample2:

Two instances fbTest and fbTest2 of FB\_BLOCK1.

Due to fbTest is marked with (OPC:1), all members of FB\_BLOCK1 are available via OPC-DA

Instance fbTest2 is not marked, but three member variables (ni1 / no1/ nx1) are marked to be available in all instances - the OPC-DA client will browse only that three variables.

```

PROGRAM MAIN
VAR
    fbTest :FB_BLOCK1; (*^ (OPC:1:bla) *)
    fbTest2 :FB_BLOCK1;
END_VAR

FUNCTION_BLOCK FB_BLOCK1
VAR_INPUT
    ni1 :INT; (*^ (OPC:1:bla) *)
    ni2 :INT;
END_VAR
VAR_OUTPUT
    no1 :INT; (*^ (OPC:1:bla) *)
    no2 :INT;
END_VAR
VAR
    nx1 :INT; (*^ (OPC:1:bla) *)
    nx2 :INT;
END_VAR

```

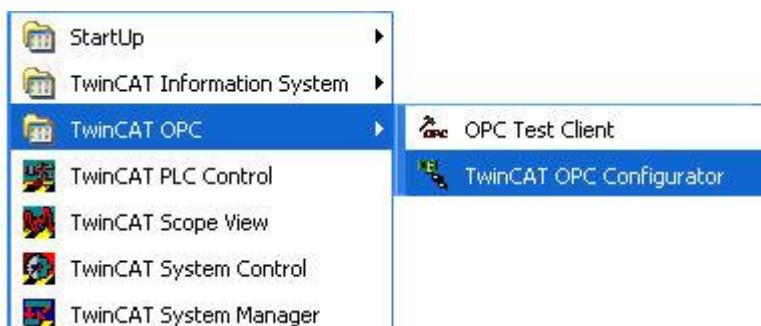
The screenshot shows the PLC code editor with a function block definition. To the right, there are two separate windows showing object hierarchies and their corresponding variable lists:

- Top Window:** Objects > PLC1 > MAIN > fbTest. Variable list: PLC1.MAIN.fbTest.ni1, PLC1.MAIN.fbTest.no1, PLC1.MAIN.fbTest.nx1, PLC1.MAIN.fbTest.no2, PLC1.MAIN.fbTest.nx2.
- Bottom Window:** Objects > PLC1 > MAIN > fbTest2. Variable list: PLC1.MAIN.fbTest2.ni1, PLC1.MAIN.fbTest2.no1, PLC1.MAIN.fbTest2.nx1.

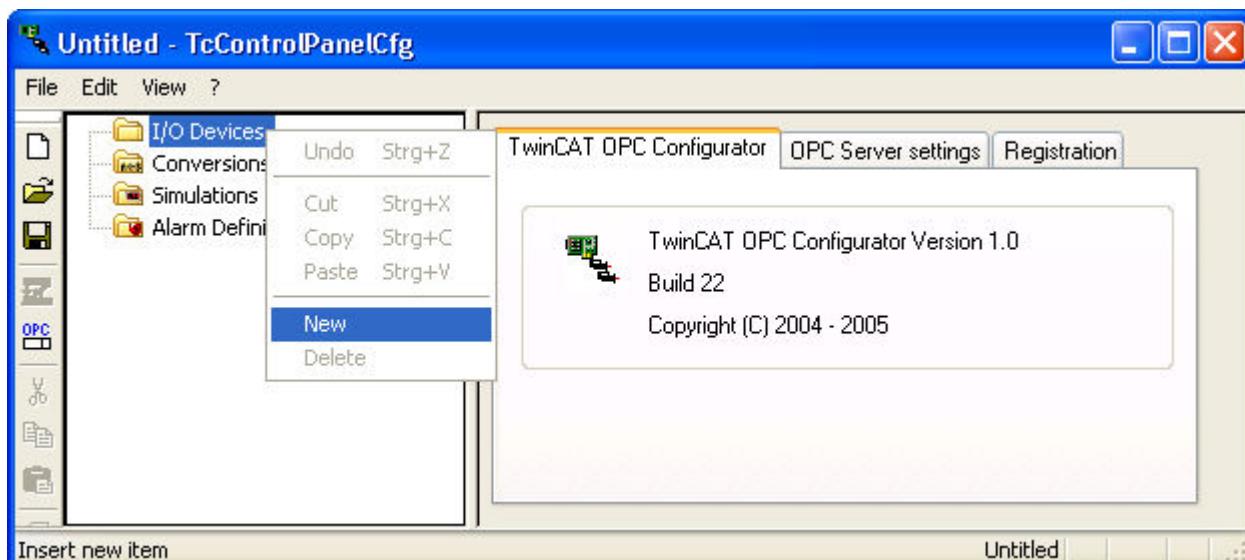
- To configure a PLC variable for readonly-access, the comment needs to be set with OPC\_PROP[0005], as shown above for the variable iReadOnly. Please make sure to write this property exactly as shown above!

## 2nd step: Configuration of TwinCAT OPC Server - to be done just once

Start the TwinCAT-OPC-Configurator "Start - All Programs - TwinCAT System - TwinCAT OPC - TwinCAT OPC Configurator"



Select "I/O Devices" with right click and then menu "New".



For "Name" please enter a name for this alarm definition.  
This name has to be OPC conform, no special characters allowed.

Sample: The TwinCAT-PLC runtime-system 1 has a PLC variable "temperatur".  
Definition of devicename with "Plc1" will publish the PLC variable via OPC as "Plc1tTemperatur" later for OPC-Clients.



After applying with "OK" you see the next dialog with settings of the new device "Plc1"  
For option "AutoCfg" select option "8 Link and filter to generic TwinCAT symbol file"  
For option "AutoCfgFile" click to "Search-buton" and navigate to e.g. "C:\TwinCAT\Boot\CurrentPLC\_1.tpy"

If you work on a small PLC project and all PLC variables should be available for OPC you can select option "AutoCfg" "7 Link to generic TwinCAT symbol file"

In this case all PLC variables are imported, the filter comment in the PLC Control is ignored.

Following settings have to be checked:

## Requirements

Type	Req./Opt.	Description
Name	Req	Name for device like "PLC1". OPC-Client browse this name
Description	Optional	
AutoCfg	Optional	1 : Include sym-file of TwinCAT-PLC defined under <AutoCfgSymFile> 2 : Include sym-file of TwinCAT-BCxxxx-project defined under <AutoCfgSymFile>  7 : Include tpy-file of TwinCAT28-project defined under <AutoCfgSymFile>  8 : Include tpy-file of TwinCAT28-project defined under <AutoCfgSymFile>, but import only symbols with symbol-property "OPC=1"
AutoCfgSymFile	Optional	full path and name of symbol-file to be included like " C:\Test\demo.sym" or " \\User1\Test\demo.sym" or "C:\TwinCAT\Boot\CurrentPLC_1.tpy"
AdsPort	Req	Number of ADS-PortID, like 801 for first PLC-runtime system 811 for first PLC-runtime system 821 for first PLC-runtime system 831 for first PLC-runtime system 800 for PLC-runtime system on BC
AdsNetId	Optional	specific ADS-NetId, like "174.12.15.45.1.1" <b>If not specified or like "0.0.0.0.0", the OPC-Server will always communicate to AdsNetId of local ADS-router.</b>
AdsTimeout	Optional	ADS timeout in ms for this ADS device. If ADS device will not replay within this time, the OPC-Server will set the requested variables to "BAD QUALITY"

Type	Req./Opt.	Description
AdsTimeSuspend	Optional	ADS suspend time in ms for this ADS device, if the ads communication fails. If an ADS device is running, it does not make sense to communicate with high frequency to that device. In case of ads-communication trouble the OPC-Server will wait till next ads-request to that device.
Disable	Optional	Disables this device. OPC-Server will ignore this device.

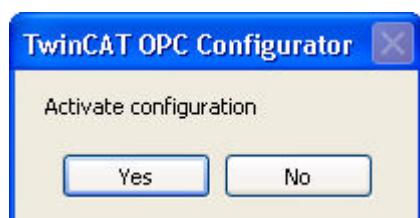
See detailed description in appendix.

### 3 rd step : Check and save configuration

Save the configuration under current name or select menu "File - Save As..." to insert a new name.

The OPC-Configurator will save the XML based file.

After saving the OPC-configurationfile we get the option to activate this configuration to be the configuration for next startup of OPC-Server.



After selecting "Yes" this current configuration will be the start-configuration of the opc-server. With next start of OPC-Server this new configuration will be active.

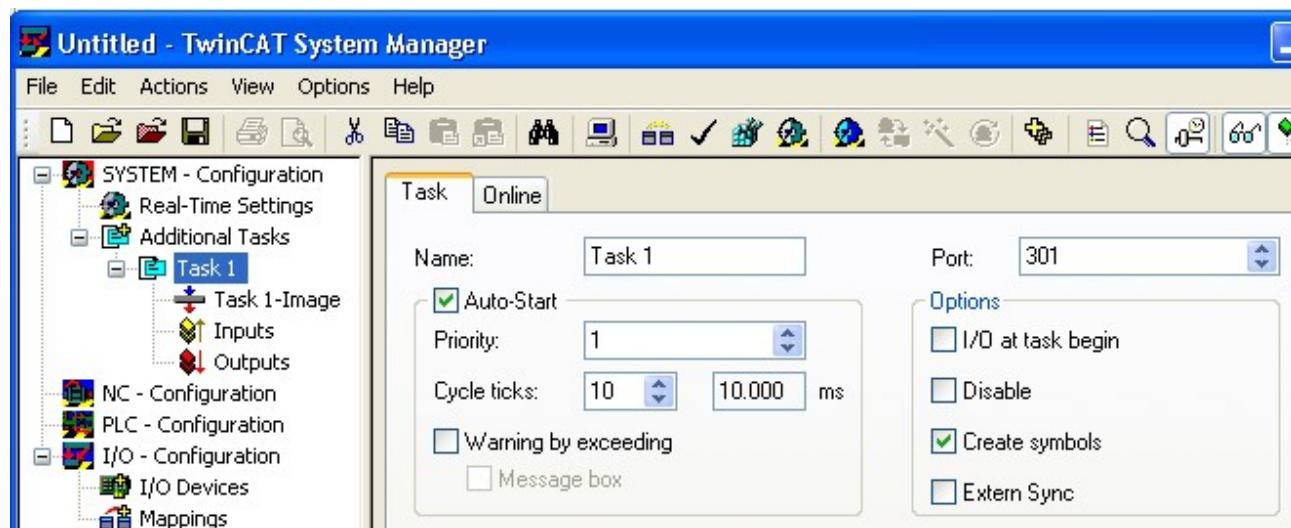


If you activated a new OPC-configuration, but the OPC-Server still has old namespace and settings the reason is that the OPC-Server was not stopped and restarted.

## 6.1.2.2 Receiving Data from the TwinCAT I/O task

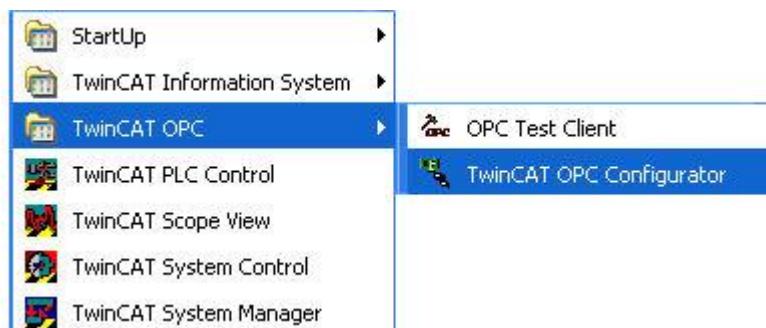
### 1st Step TwinCAT I/O task configuration

In the TwinCAT System Manager navigate through "System configuration ->Additional tasks -> Task x". Once there, activate "Create symbol entry".

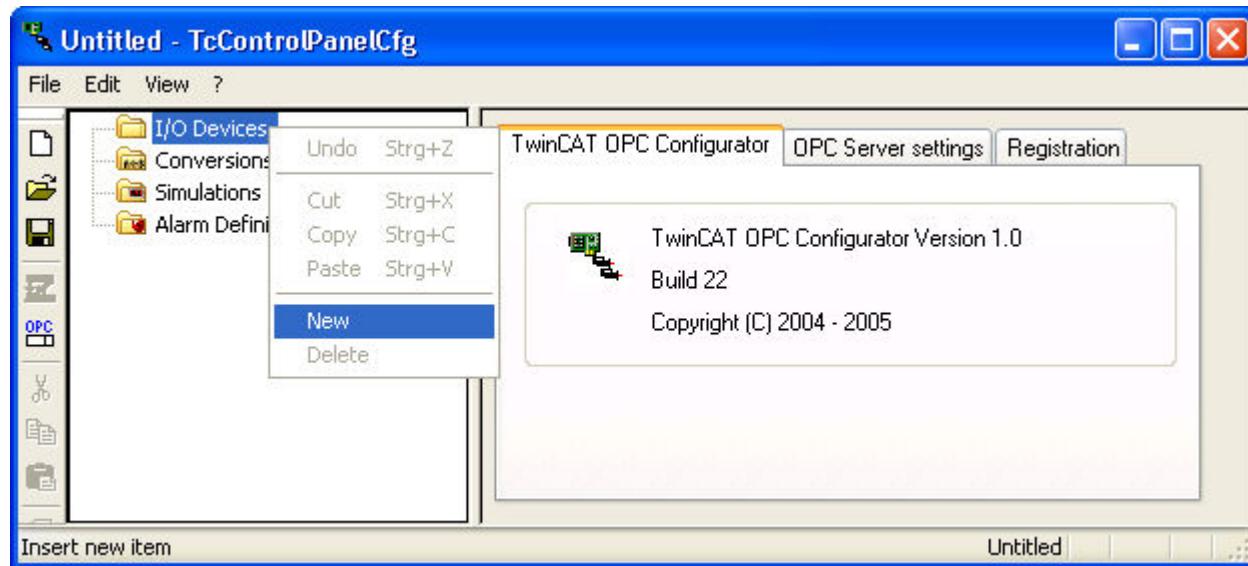


## 2nd Step Import into the TwinCAT OPC server

Start TwinCAT-OPC-Configurator "Start - All Programs - TwinCAT System - TwinCAT OPC - TwinCAT OPC Configurator"



In the left navigation tree please select "I/O Devices" with a right click the menu item "**New**".



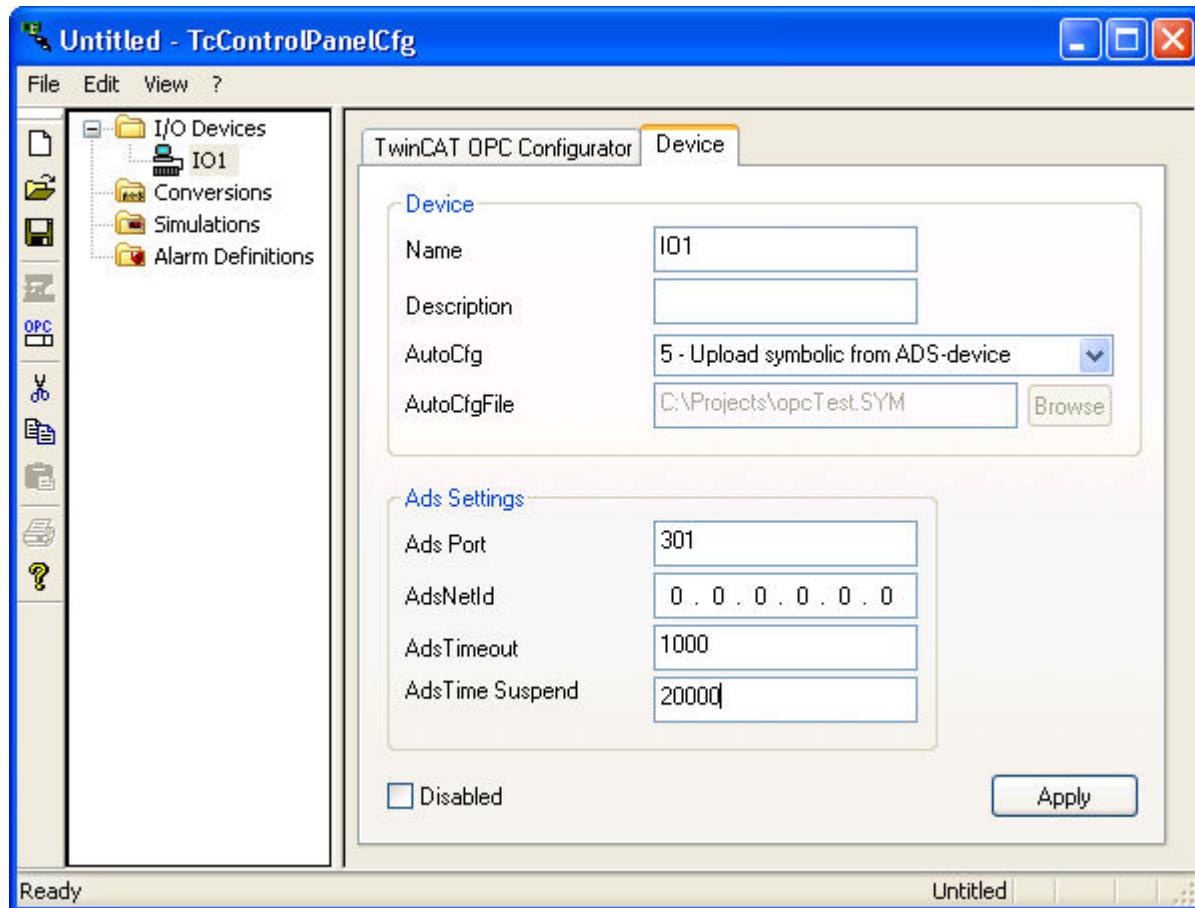
Enter a name for that ADS device in field "**Name**".

The name has to be OPC conform. Special characters are not allowed.

The device-name will be part of the ItemID later.

Sample: TwinCAT-PLC was defined with a variable "Temp". Choosing the devicename as "IO1" the opc-server present the variable as "IO1.Temp".

With click on "OK" we get the new dialog for further settings of device :



Following settings have to be checked :

Type	Req./Opt.	Description
Name	Req	Name for device like "IO1". OPC-Client browse this name
Description	Optional	
AutoCfg	Optional	5 : Upload symbolic from ADS-device.
AutoCfgFile	Optional	will be ignored
AdsPort	Req	Number of ADS-PortID, like 301 for first IO task 302 for next IO task ...
AdsNetId	Optional	specific ADS-NetId, like "174.12.15.45.1.1" <b>If not specified or like "0.0.0.0.0", the OPC-Server will always communicate to AdsNetId of local ADS-router.</b>
AdsTimeout	Optional	ADS timeout in ms for this ADS device. If ADS device will not replay within this time, the OPC-Server will set the requested variables to "BAD QUALITY".
AdsTimeSuspend	Optional	ADS suspend time in ms for this ADS device, if the ads communication fails. If an ADS device is running, it does not make sense to communicate with high frequency to that device. In case of ads-communication trouble the OPC-Server will wait till next ads-request to that device.
Disable	Optional	Disables this device. OPC-Server will ignore this device.

See detailed description in appendix.

### 3 rd step : Check and save configuration

Save the configuration under current name or select menu "File - Save As..." to insert a new name.  
The OPC-Configurator will save the XML based file.

After saving the OPC-configurationfile we get the option to activate this configuration to be the configuration for next startup of OPC-Server.



After selecting "Yes" this current configuration will be the start-configuration of the opc-server. With next start of OPC-Server this new configuration will be active.



If you activated a new OPC-configuration, but the OPC-Server still has old namespace and settings the reason is that the OPC-Server was not stopped and restarted.

### 6.1.3 Conversion

The OPC server offers the feature to online convert process data.

The OPC-Server takes care of online-conversion in both communication directions ::

- Processvalue --> Communication to OPC-Server --> conversion within OPC-Server --> Communication to OPC-Client --> OPC-Client
- OPC-Client --> Communication to OPC-Server --> conversion within OPC-Server --> Communication to process --> Processvalue

#### Sample :

- KL3202 offers process value of temperatures in unit 1/10 Grad Celsius, e.g. 200.
- The opc-server is configured with a conversion "Factor 10" which is linked to this plc-variable.
- The opc-client gets the process value 20 Grad.
- If opc-client writes a new value 25 Grad celsius to OPC-server, the server will convert this to 250 and writes this value to the PLC.

#### Sample

The <https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162181387.zip> contains a PLC project and an OPC conversion definition.



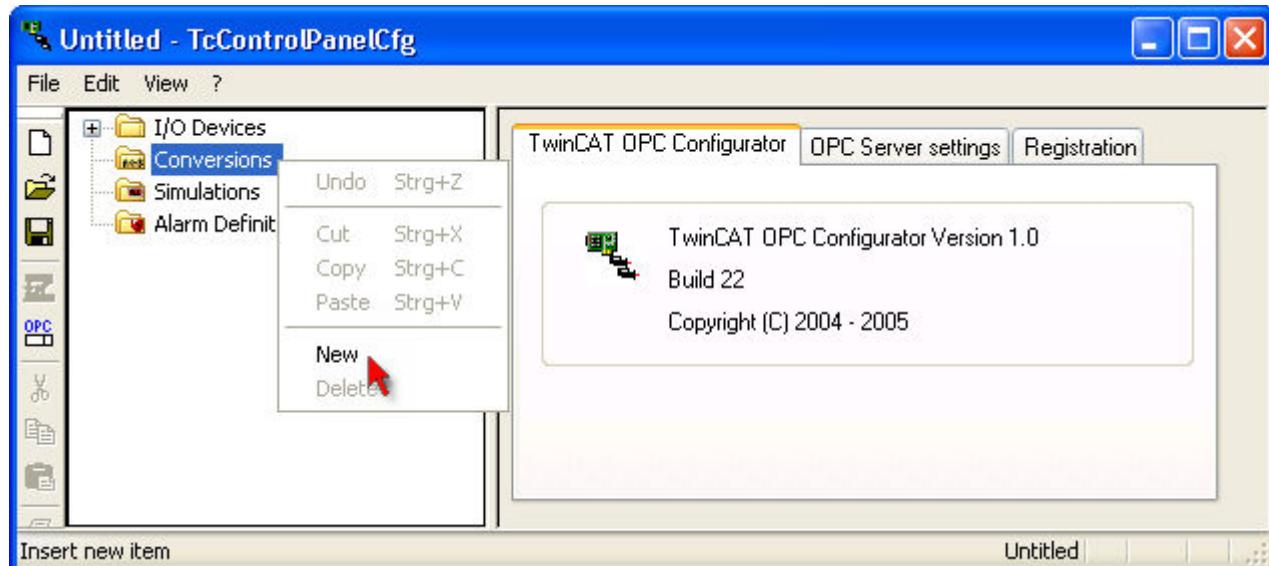
OPC-Server presents a converted variable as Datatyp "VT\_R8 / Double" to OPC-client - independend of numerical datatyp in the PLC.

#### 1. Step: Configuration of conversion for TwinCAT OPC Server

Start TwinCAT-OPC-Configurator "Start - All Programs - TwinCAT System - TwinCAT OPC - TwinCAT OPC Configurator"

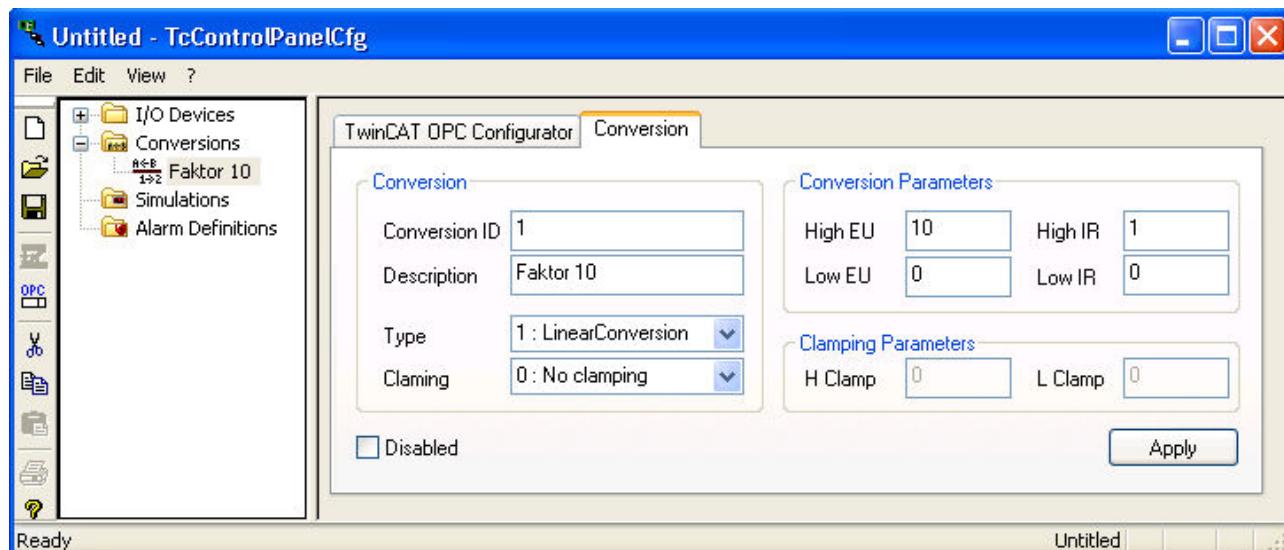


Navigate to "**Conversions**", right click and select menu "New".



For "**Name**" please enter a name for this alarm definition.  
This name has to be OPC conform, no special characters allowed.

With "OK" you see the dialog to configure in detail your new "**Conversion**".



Following settings can be done:

Type	Req./Opt.	Description
Conversion ID	Required	
Type	Required	
Clamping		

Type	Req./Opt.	Description
High EU	Req	
Low EU	Optional	
High IR	Optional	
Low IR	Optional	
H Clamping		
L Clamping		
Disable	Optional	Disables this conversion. OPC-Server will ignore this conversion.

### Step: Check and save configuration

Save new configuration by selecting menu "File - Save As...".

After successfully creating and saving of new OPC-configurationfile the next dialog offers to activate this new configuration.



Select "OK" to activate the configuration. With next startup the OPC-Server will work with new settings.



In case of opc server offers old previous configuration to opc-clients but a new configuration was activated this means, that OPC-server was not restarted to load fresh configuration.

### 3. Step: Link of PLC-Variable with Conversion-template

With previous steps we defined a "conversion-Template" with detailed information about conversion handling.

Now we define, which PLC variable should be handled with conversion.

This link is configured in PLC-Control as and additional information for IEC1131-variable declaration.

Take care on following settings :

Add a comment behind PLC-variable declaration. This comment will be interpreted by the opc-server and links this plc-variable to a previous defined conversion-template.

The conversion-template with the unique conversion-ID was configured in step 1.



Today just global PLC-variables can be linked to conversion-templates. Elements of structures or arrays can not be configured as an OPC-conversion.

Future versions of OPC-Server will not have this limitation.

Sample :

- PLC-Variable of Typ INT **intMemCounterConversion** is an OPC-DA variable which is converted by the opc-server as defined in conversion template 102.

The syntax is important :

```
intMemCounterConversion : INT;      (* ~
                                     (OPC : 1 : available for
                                     OPC-clients)
                                     (OPC_PROP[6010] : 1      :
                                     OPC_PROP_CONV_ENABLE)
                                     (OPC_PROP[6011] : 102    :
                                     OPC_PROP_CONV_ID)
                                     *)
```

Each time compiling the PLC-project the PLC-Control will create the file <PLC-projectname>.tpy. This XML based file contains information about PLC-variables and their link to the OPC-server.

The OPC server will analyze this information of the TPY file., so configure the OPC-Server to know the file <PLC-projectname>.tpy <https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162179979.htm>

## 6.1.4 Simulation

The OPC-Server offers to simulate process-values. In this case no communication to the ADS-device (like PLC) is done instead simulated values are send to OPC-client.

Different simulation templates like sinus, ramp, random etc. are available to be configured with detailed behaviour (amplitude, start /stop values).

The feature "Simulation" ist available in OPC Server version V4.0.x.x.

OPC-Server version V4.1.x.x offers a multiple more performant access to TwinCAT-devices, the simulation-mode is under construction and not supported today.

The OPC-Server offers

- simulation of all process values
- simulation of some specific process values

It is not possible to combine both simulated and real-process values.

**Sample**

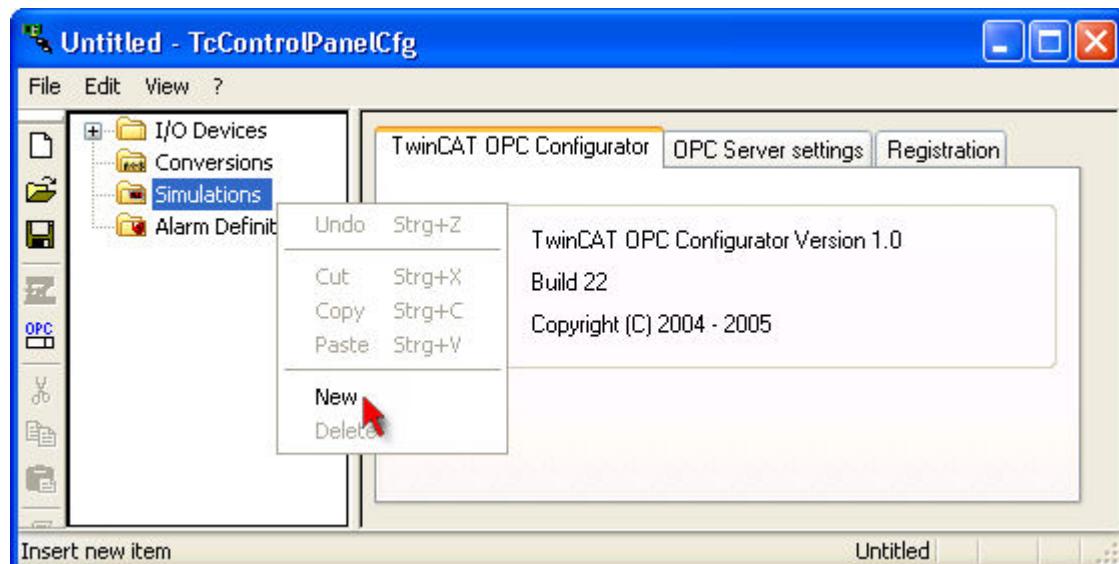
The <https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162182795.zip> contains both, the PLC project and the definition of OPC-simulation.

### 1. Step: Configuration of simulation-templates for TwinCAT OPC Server

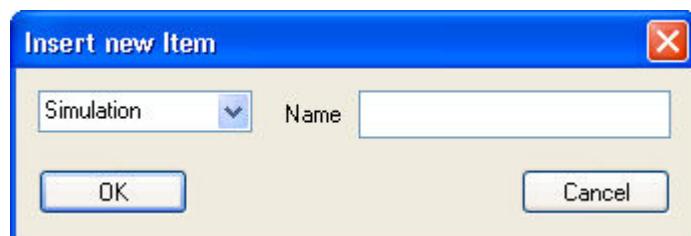
Start TwinCAT-OPC-Configurator "Start - All Programs - TwinCAT System - TwinCAT - TwinCAT OPC Configurator"



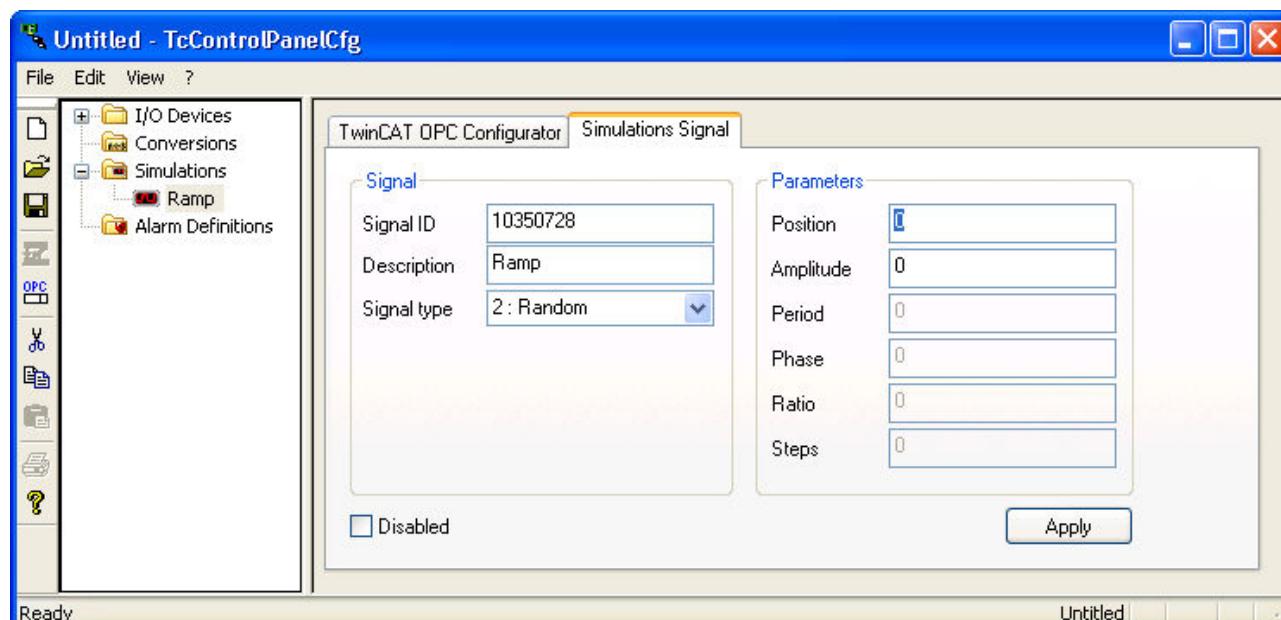
Navigate to "**Simulations**", right click and select menu "**New**".



For "**Name**" please enter a name for this simulation template.  
This name must be OPC conform, no special characters allowed.



With "OK" you get into the dialog to configure in details your new "**Simulations Signal**".



Following settings can be done :

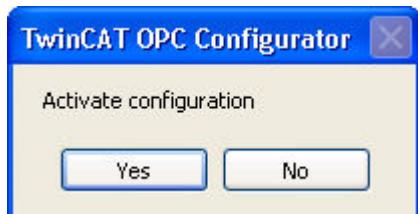
Signal Type	Req./Opt.	Description
0 : ReadCount	Optional	
1 : WriteCount	Optional	
2 : Random	Optional	
3 : Ramp	Optional	
4 : Sine	Optional	

Signal Type	Req./Opt.	Description
5 : Square	Optional	
6 : Triangle	Optional	
7 : Stepp	Optional	
Disable	Optional	Disables this signal. OPC-Server will ignore this signal.

## 2. Step: Check and save configuration

Save new configuration by selecting menu "File - Save As..." .

After successfully creating and saving of new OPC-configuration file the next dialog offers to activate this new configuration.



Select "OK" to activate the configuration. With next startup the OPC-Server will work with new settings.



In case of opc server offers old previous configuration to opc-clients but a new configuration was activated this means, that OPC-server was not restarted to load fresh configuration.

## 3. Step: Link of PLC-Variable with Alarm-Type

With previous steps we defined a "simulation-template" with detailed information about simulation behaviour. Now we define, which PLC variable should be handled like a simulation.

This link is configured in PLC-Control as and additional information for IEC1131-variable declaration.

Take care on following settings :

Add a comment behind PLC-variable declaration. This comment will be interpreted by the opc-server and links this plc-variable to a previous defined simulation-template.

The simulation-template with the unique Simulation-ID was configured in step 1.



Today just global PLC-variables can be linked to Simulation-Templates. Elements of structures or arrays can not be configured as an simulated variable.

Future versions of OPC-Server will not have this limitation.

Sample :

- PLC-Variable of Typ LREAL **dblMemAlarmCounter** is a OPC variable linked to simulation template 6. The OPC-Server will not access this PLC variable, but instead give simulated values back to OPC-client.

The syntax is important:

```

dblMemAlarmCounter      :      LREAL;    (* ~
                                (OPC      : 1 :
available for OPC-clients)
                                (OPC_PROP[6006]  : 1 :
OPC_PROP_SIMU_ENABLE)
                                (OPC_PROP[6007]  : 6 :
OPC_PROP_SIMU_ID)
                                *)

```

Each time compiling the PLC-project the PLC-Control will create the file <PLC-projectname>.tpy. This XML based file contains information about PLC-variables and their link to simulation templates. The OPC server will analyze this information of the TPY file., so configure the OPC-Server to know the file <PLC-projectname>.tpy <https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162179979.htm>

## 6.1.5 Item Properties

OPC-specification allows to add additional information to single OPC-items (opc-item means : a TwinCAT variable). This optional functionality is named "**OPC-Item-Properties**" in the OPC-spec and allows an opc-client to browse and read these additional properties.

Sample:

As an option the PLC variable "temperature" can hold addition informaton like **comment** (OPC-Property-ID 101) and also **UNIT** (OPC-Property-ID 100) which can be offered by the opc-server.

### Configuration of OPC Property ID in the PLC

Add a comment behind PLC-variable declaration. This comment will be interpreted by the opc-server and will define OPC-Item-Properties.



Today only global PLC-variables can contain item-properties. Elements of structures or arrays can not be configured.

Future versions of OPC-Server will not have this limitation.

Sample of PLC variable-declaration:



All item-properties are optional.

```

lTemperatur : DWORD;
(* ~
(OPC : 1 : Make variable visible for
OPC-Server)
(OPC_PROP[0005] : 3 :
OPC_PROP_RIGHTS, here Read AND Write Access)
(OPC_PROP[0100] : Grad F      :
OPC_PROP_UNIT)
(OPC_PROP[0101] : Demovariable      :
OPC_PROP_DESC)
(OPC_PROP[0205] : We are the champions :
OPC_PROP SND)
(OPC_PROP[0206] : ..\..\info.html      :
OPC_PROP_HTML)
(OPC_PROP[0207] : ..\..\service.avi      :
OPC_PROP_AVI)
*)

```

Each time compiling the PLC-project the PLC-Control will create the file <PLC-projectname>.tpy. This XML based file contains information about PLC-variables and their link to the OPC-server.

The OPC server will analyze this information of the TPY file., so configure the OPC-Server to know the file <PLC-projectname>.tpy <https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162179979.htm>

List of supported OPC-item properties of OPC-Specification :

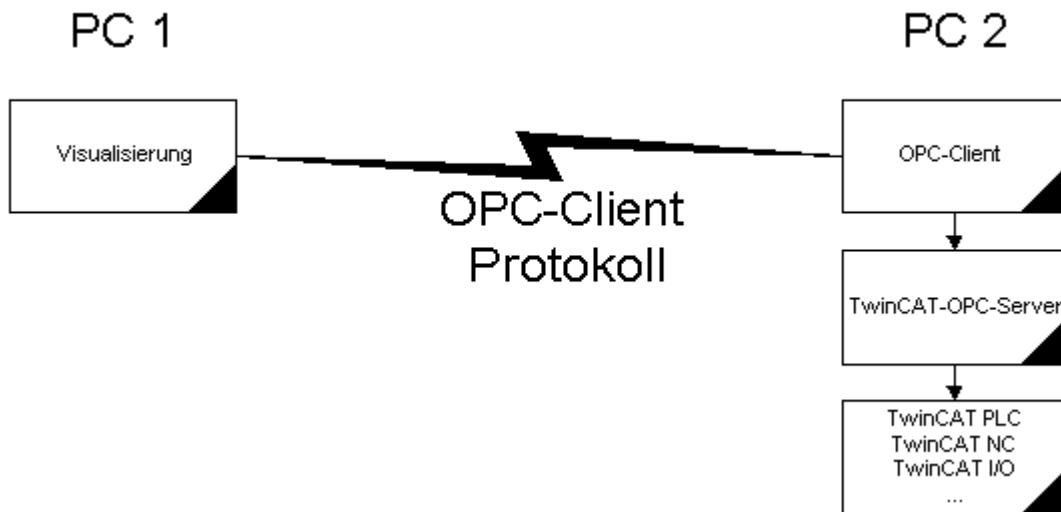
<b>Property ID laut OPC Spec</b>	<b>Req./Opt.</b>	<b>Description</b>
OPC_PROP[0005]	Optional	<p>OPC_PROP_RIGHTS</p> <p>1 : OPC-Server will publish this variable with access : OPC_READABLE</p> <p>2 : OPC-Server will publish this variable with access : OPC_WRITEABLE</p> <p>3 : OPC-Server will publish this variable with access : OPC_READABLE AND OPC_WRITEABLE</p> <p>Default : 3 : ReadWrite Zugriff</p>
OPC_PROP[0100]	Optional	OPC_PROP_UNIT : Unit
OPC_PROP[0101]	Optional	OPC_PROP_DESC : Description
OPC_PROP[0102]	Optional	OPC_PROP_HIEU
OPC_PROP[0103]	Optional	OPC_PROP_LOEU
OPC_PROP[0106]	Optional	OPC_PROP_CLOSE
OPC_PROP[0107]	Optional	OPC_PROP_OPEN
OPC_PROP[0200]	Optional	OPC_PROP_DSP
OPC_PROP[0201]	Optional	OPC_PROP_FGC
OPC_PROP[0202]	Optional	OPC_PROP_BGC
OPC_PROP[0203]	Optional	OPC_PROP_BLINK
OPC_PROP[0204]	Optional	OPC_PROP_BMP
OPC_PROP[0205]	Optional	OPC_PROP SND
OPC_PROP[0206]	Optional	OPC_PROP_HTML
OPC_PROP[0207]	Optional	OPC_PROP_AVI
OPC_PROP[6007]	Optional	BECKHOFF Defined : Simulation ID
OPC_PROP[6008]	Optional	BECKHOFF Defined : Alarm enabled
OPC_PROP[6009]	Optional	BECKHOFF Defined : Alarm ID
OPC_PROP[6010]	Optional	BECKHOFF Defined : Conversion enabled
OPC_PROP[6011]	Optional	BECKHOFF Defined : Conversion ID

## 6.1.6 Data exchange over the network

### 6.1.6.1 Network via client protocol

Visualisation systems sometimes offer their own protocol to bridge the network. An OPC client from the visualisation system supplier is also installed on the TwinCAT controller PC for this purpose.

The data flow proceeds as follows:



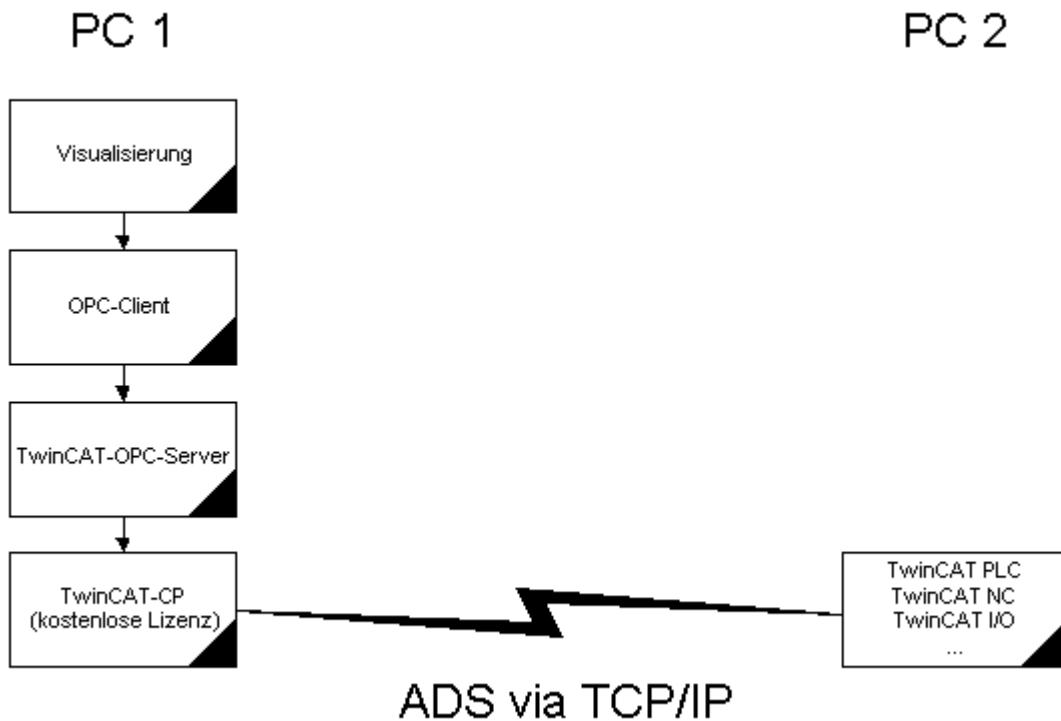
#### Performance advantage

The primary advantage of this configuration is that considerably less loading is placed on the network, since only the process values that have altered are exchanged there: the OPC server checks the whole process cyclically, but only passes the changes in the process on to the OPC client.

#### 6.1.6.2 Network via TwinCAT ADS

The TwinCAT OPC server performs the communication with the TwinCAT components (PLC run-time systems, bus terminal controllers etc.) via TwinCAT ADS. Since these services are also available in the network, the TwinCAT OPC server can also communicate with TwinCAT components that exist in the network.

You will find the [settings needed for ADS communication over a network](#) in the ADS Reference.



## Settings in the OPC server

An ADS communication partner is always specified by two parameters: These are what is known as the "AdsAmsNetId" (e.g. 1.2.3.4.5.6) and the "PortId" (e.g. 801 for the first PLC run-time system).

If ADS is to be used to exchange data with, for example, the first PLC run-time system on another PC, then the AdsAmsNetId of the target PC, and "801" as the PortId, must be entered in the OPC server under "Device settings".



If possible, the option of using the "Network via client" protocol should be selected for performance reasons.

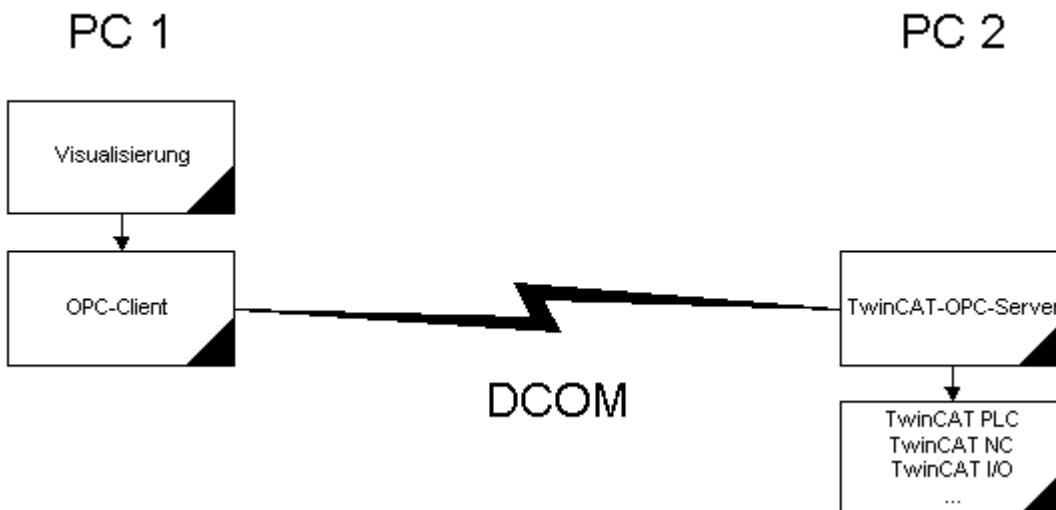
## Also see about this

- █ Network via client protocol [▶ 32]

### 6.1.6.3 Network via DCOM

If you wish (or need) to allow data exchange between an OPC client and the TwinCAT OPC server over a network via DCOM, this DCOM access type must be configured beforehand.

The data flow is then as follows:



The following steps describe the necessary settings for the PC on which the TwinCAT OPC server is installed.

(Use step 1 for the PC on which the OPC client is installed.)

#### Requirement

It is assumed that all the PCs which are to participate via DCOM in an OPC client/server connection are already set up for network operation (i.e. they have configured network cards running the TCP/IP protocol)

## 6.2 OPC XML-DA Server

This chapter describes how to configure the OPC XML DA Server on BECKHOFF PC based systems.

#### Summary :

**The OPC-XML DA server is configured with default xml-configuration file. Just activate once a checkbox in PLC-Control to download the symbol information to target device.**

With TwinCAT 2.10 Build 1240 (or higher) there is no manual configuration required. Please refer to section "Zero configuration" on this page.

With previous TwinCAT versions please refer to section "**Manual or specific configurations**" on this page.

### Zero configuration (requires TwinCAT 2.10 on host PC)

#### Details :

- The OPC-XMLDA configuration file "...\\wwwroot\\TcOpcXmIDa\\TcOpcXmISvrCfg.xml" contains the list of ADS-devices which are available via OPC-XML-DA interface.
- By default this list is pre-configured for one local PLC-runtime system (first PLC runtime-system on ADS-port 801)
- By default the required symbol-information is linked to **C:\\TwinCAT\\Boot\\CurrentPlc\_1.tpy**

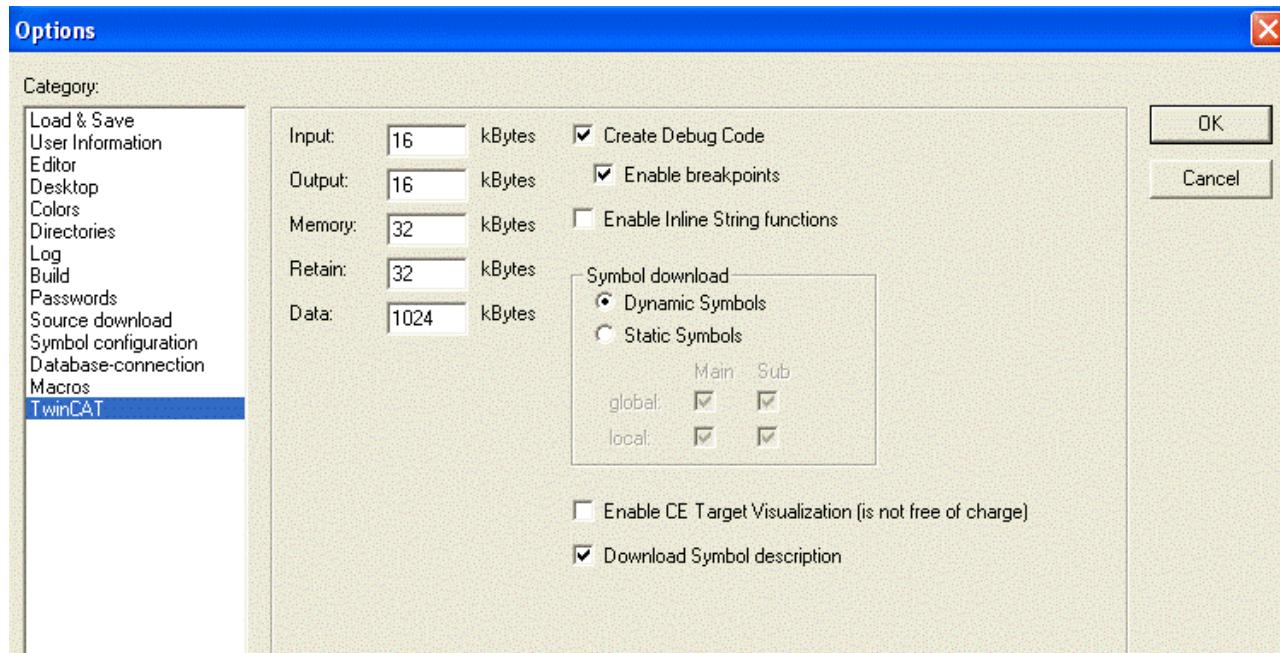
```
<?xml version="1.0"
encoding="utf-8" standalone="no"?>
<TcOpcXmlSvrConfiguration>
<Namespace>
  <Devices>
    <Device>
      <Name>PLC1</Name>
      <AdsNetId>0.0.0.0.0</AdsNetId>
      <AdsPort>801</AdsPort>
      <AdsTimeout>2000</AdsTimeout>
      <AdsTimeSuspend>20000</AdsTimeSuspend>
      <AutoCfg>7</AutoCfg>
      <!-- NOTE: CurrentPlc_1.tpy will be downloaded from
PLC-Control -->
      <AutoCfgSymFile>C:\\TwinCAT\\Boot\\CurrentPlc_1.tpy</AutoCfgSymFile>
      <Disabled>0</Disabled>
    </Device>
  </Devices>
</Namespace>
</TcOpcXmlSvrConfiguration>
```

With TwinCAT Version 2.10 Build 1240 the PLC Control allows to download the symbol file <plc-project-name>.TPY to the target TwinCAT machine. The TPY file contains information about all available PLC-variables in the PLC-project.

OPC-XML DA CE Server requires this TPY-information.

To download the current symbol information in a TPY file navigate in the PLC-Control to "Project-->Options-->TwinCAT" and check the option "**Download Symbol description**"

With each download of the PLC-code now also the <project-name>.TPY file is downloaded as "**C:\\TwinCAT\\Boot\\CurrentPlc\_1.tpy**" containing information about symbols of first run time system. (Second runtime system would be "**C:\\TwinCAT\\Boot\\CurrentPlc\_2.tpy**" etc.)



#### Summary:

The OPC-XML DA server is configured with default xml-configuration file. Just activate the checkbox to download the symbol information within PLC-Control environment.



**OPC-XML DA Server checks configuraton on startup. See notes to [reload configuration \[► 45\]](#) today (without reboot).**

### Optional manual or specific configurations ?

If the default configuration is not available you have to configure the opc-xml-da configuration file "...\\wwwroot\\TcOpcXmlDa\\TcOpcXmlSvrCfg.xml" manual.

- If required the opc-xml configuration file can be done with TwinCAT OPC COnfigurator

Sample for OPC XML DA server accessing multiple ADS devices

```
<?xml version="1.0"
encoding="utf-8" standalone="no"?>
<TcOpcXmlSvrConfiguration>
<Namespace>
  <Devices>
    <Device>
      <Name>PLC1</Name>
      <AdsNetId>0.0.0.0.0</AdsNetId>
      <AdsPort>801</AdsPort>
      <AdsTimeout>2000</AdsTimeout>
      <AdsTimeSuspend>20000</AdsTimeSuspend>
      <AutoCfg>7</AutoCfg>
      <!-- NOTE: CurrentPlc_1.tpy will be downloaded from
PLC-Control -->
      <AutoCfgSymFile>C:\\TwinCAT\\Boot\\CurrentPlc_1.tpy</AutoCfgSymFile>

      <Disabled>0</Disabled>
    </Device>
    <Device>
      <Name>PLC2</Name>
      <AdsNetId>0.0.0.0.0</AdsNetId>
      <AdsPort>811</AdsPort>
      <AdsTimeout>2000</AdsTimeout>
      <AdsTimeSuspend>20000</AdsTimeSuspend>
      <AutoCfg>7</AutoCfg>
      <!-- NOTE: CurrentPlc_2.tpy will be downloaded from
PLC-Control -->
      <AutoCfgSymFile>C:\\TwinCAT\\Boot\\CurrentPlc_2.tpy</AutoCfgSymFile>
      <Disabled>0</Disabled>
    </Device>
  </Devices>
</Namespace>
</TcOpcXmlSvrConfiguration>
```



OPC-XML DA Server checks configuraton on startup. Reconfiguration might require a new start today.

## 6.2.1 Configuration OPC XML-DA on Windows XP

This chapter describes the IIS configuration required for OPC XML DA for operating system XP.



For other Windows-OS the configuration can differ in details.

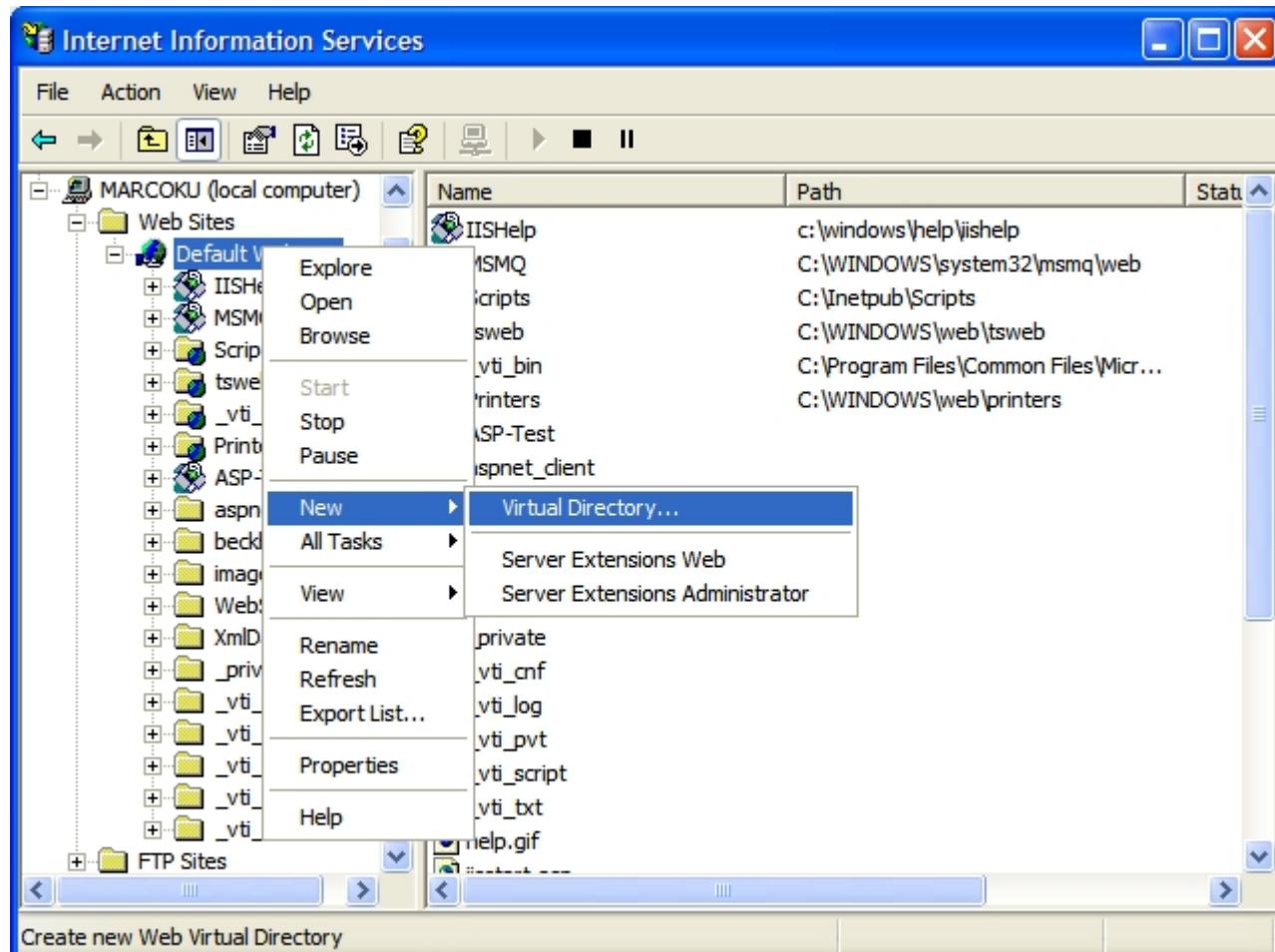
### Requirements :

The TwinCAT OPC XML DA server has to be installed on the PC. This chapter describes how to configure the Microsoft IIS.

#### Step 1: Create "Virtual Directory" in IIS (Internet Information Service)

- Open "Internet Information Services" (under "Control Panel/Administrative Tools/").

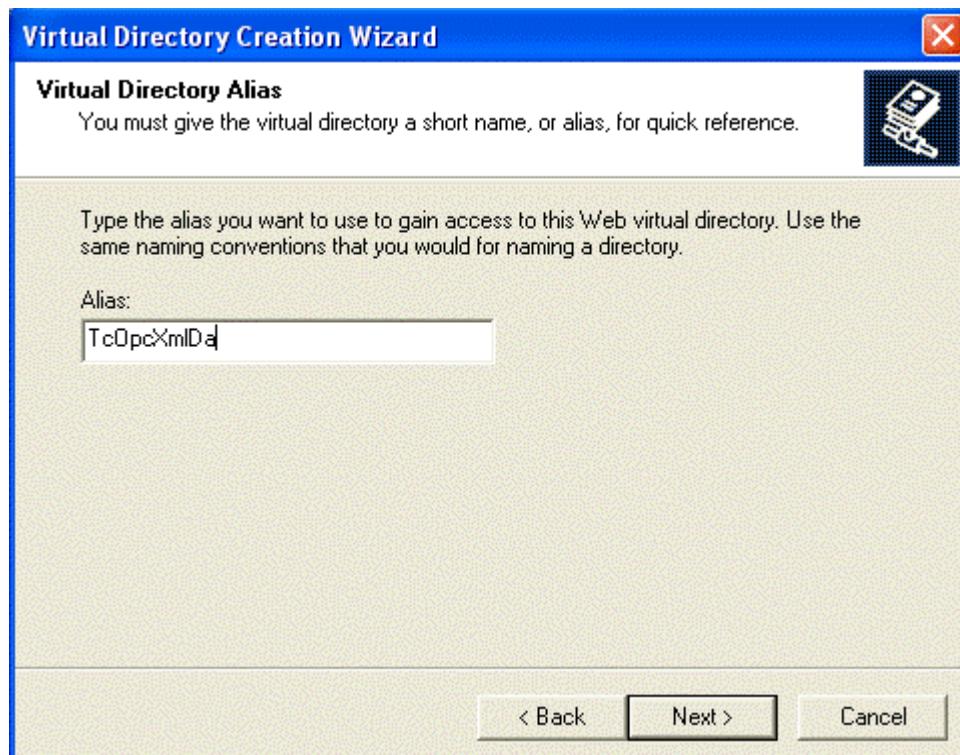
- Right click on "Default Web Site"
- Select "New" and "Virtual Directory..."



Everybody is welcome, so just click next.

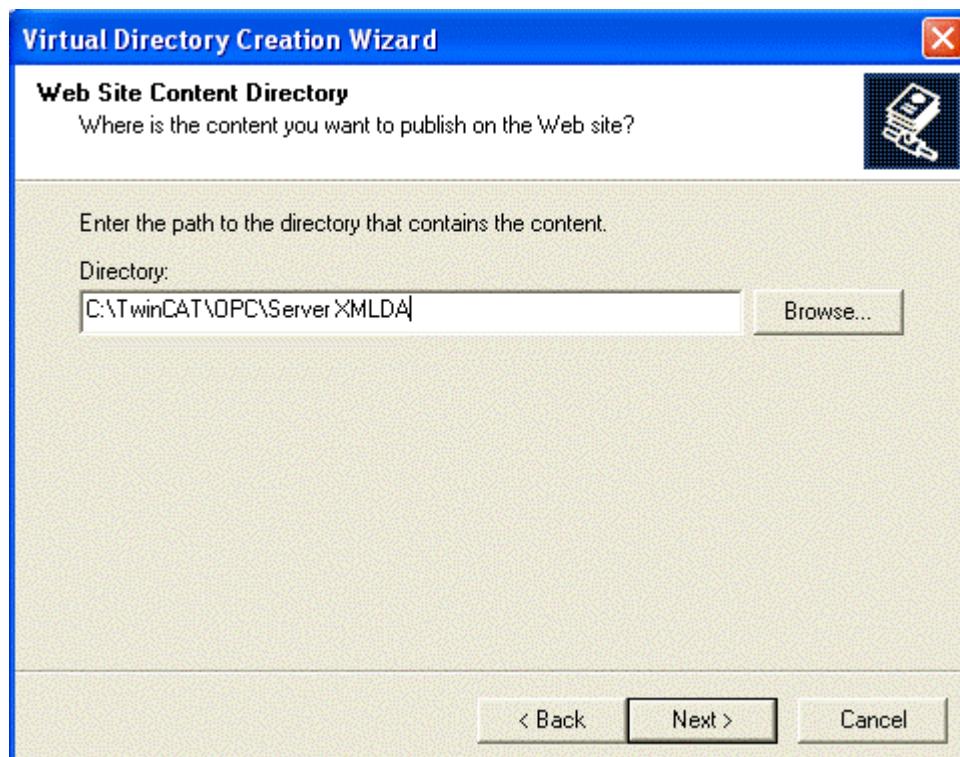


Please enter the alias "TcOpcXmlDa" and click "Next"

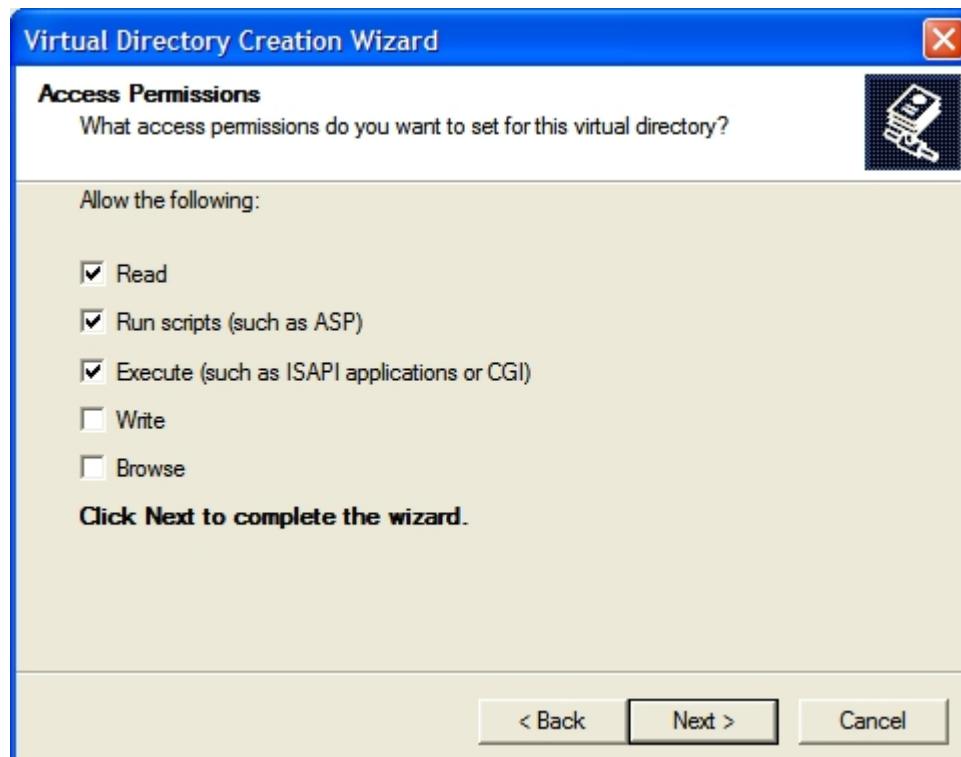


With "Browse..." you have to specify the folder which contains the TwinCAT OPC Xml DA Server. By default the folder should be like "**C:\TwinCAT\OPC\Server XMLDA**".

Select "Next" to proceed.



Check options "Read", "Run scripts" and "Execute" and click "Next".



Select "Finish" to finish the configuration of TwinCAT OPC XML DA Server.



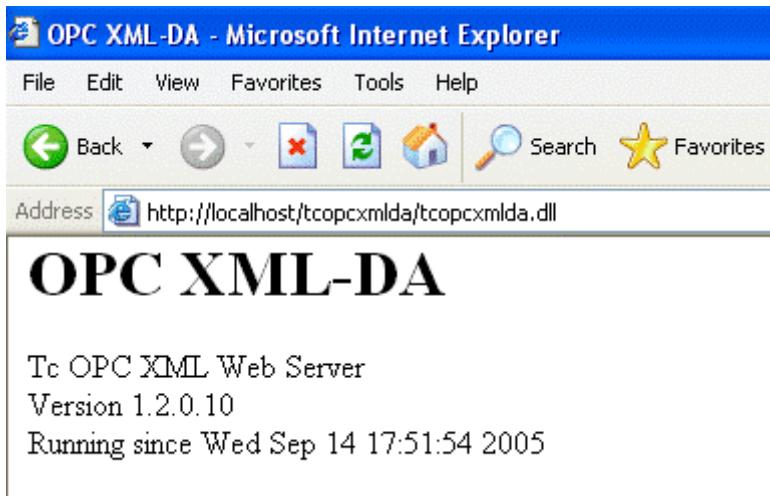
## Step 2 : Test

The URL of the OPC-XML-DA server on the PC system can be checked locally or from a remote PC: In both cases open the Internet explorer and enter the URL of the OPC Server XML DA on PC system like :  
**http://<ip-adress or name of PC device>/tcopxmlida/tcopxmlida.dll**

Sample :

<http://192.16.17.5/tcopcxmlida/tcopcxmlida.dll>  
or  
<http://localhost/tcopcxmlida/tcopcxmlida.dll>

The TcOpcXmlDa server will reply with a status page containing the product version :



In case of problems (like receiving no HTML status data) please check if a proxy server is activated on your host PC.

After deactivating the proxy and reloading the URL the Opc XML DA server should reply with status info.

## 6.2.2 Configuration OPC XML-DA on Windows 7

This chapter describes the IIS configuration required for OPC XML DA running on Windows 7.



The configuration can differ for other Windows operating systems.

### Requirements :

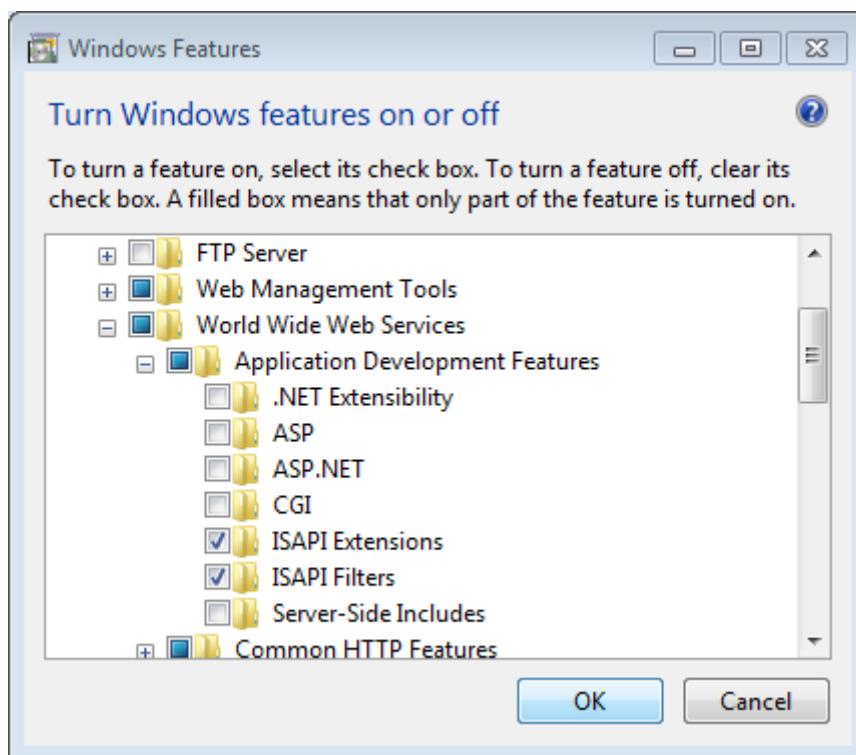
The TwinCAT OPC XML DA server has to be installed on the PC. This chapter describes how to configure Internet Information Service 7.

#### Step 1: Installing IIS7

By default, IIS7 is not part of the Windows 7 installation. Therefore you need to add this functionality manually. For more information see <http://technet.microsoft.com/de-de/library/cc725762%28v=ws.10%29.aspx>.

When installing IIS7, the following extensions need to be activated:

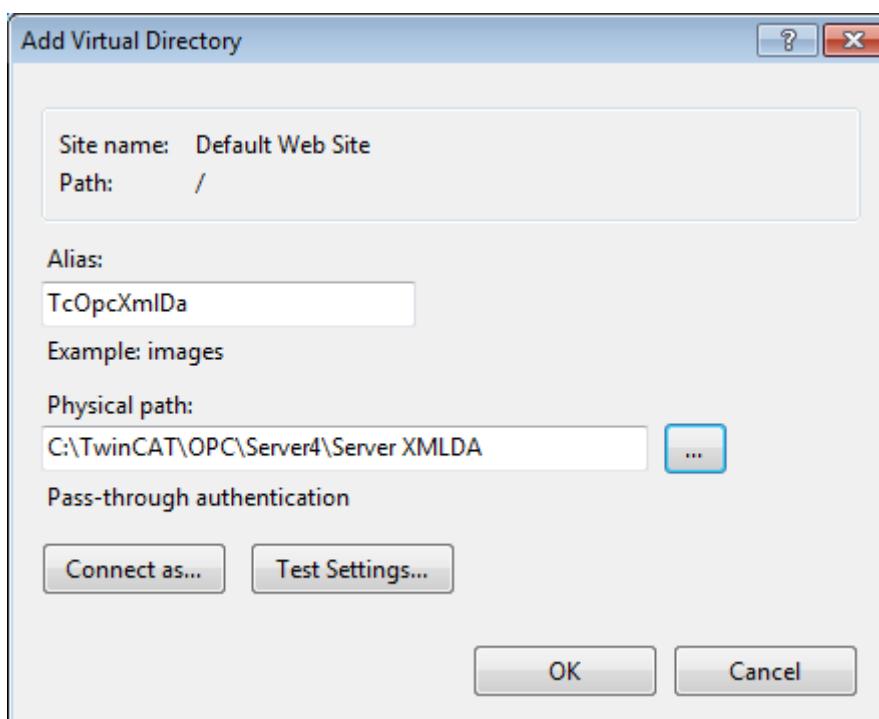
- ISAPI Extensions
- ISAPI Filters



## Step 2: Create "Virtual Directory" in IIS7 (Internet Information Service)

Usually this step will be performed automatically by the Setup.

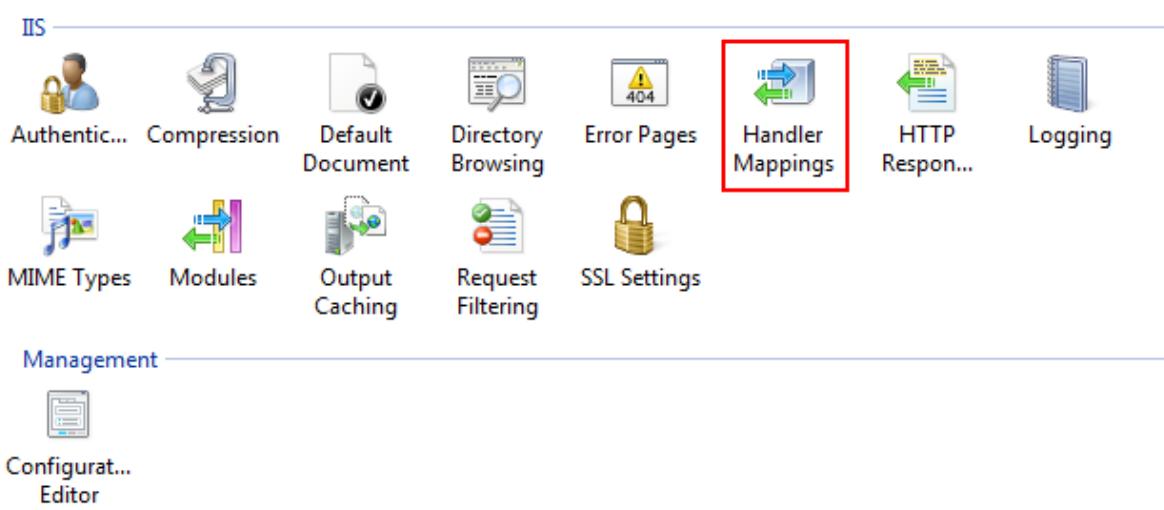
- Open "Internet Information Service (IIS) Manager" which can be found under "Control Panel\Administrative Tools\"
- Right click on "Default Web Site"
- Select "Add Virtual Directory..."
- Please enter the alias "TcOpcXmlDa" and the physical path to your TwinCAT OPC XML DA Server installation. By default, this folder should be under C:\TwinCAT\OPC\Server4\Server XMLDA. Click on "OK".



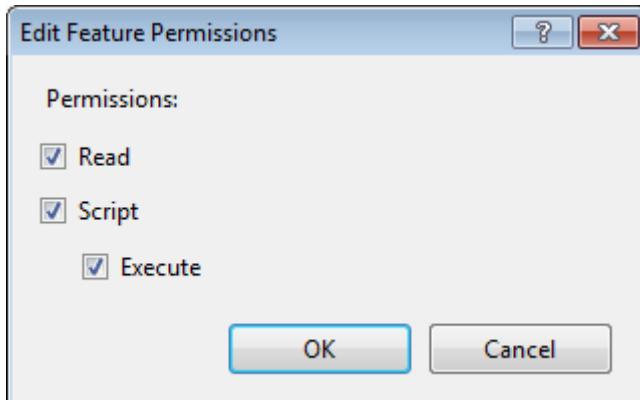
- Double-click the "Authentication" icon, select "Anonymous Authentication" and click on "Edit". Instead of specifying a user account, select the "Application pool identity" and click on "Ok".



- Next, you need to set execute permissions on that virtual directory. Select the added directory and double-click on "Handler Mappings"

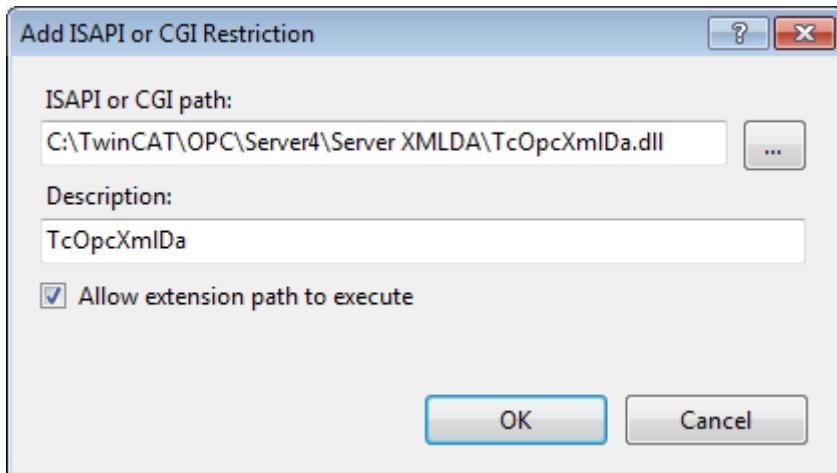


- Click on "Edit Feature Permissions" and select the "Execute" permission. Click on "OK".



- As a next step you need to create an ISAPI allowance. Please select the root node in IIS Manager (named after your computername) and then double-click "ISAPI and CGI Restrictions".

- Click on "Add" to create a new enabled extensions. In the "ISAPI or CGI path" textbox, please specify the path to TcOpcXmlDa.dll, which normally is "C:\TwinCAT\OPC\Server4\Server XMLDA\TcOpcXmlDa.dll". Also select the checkbox "Allow extension path to execute".

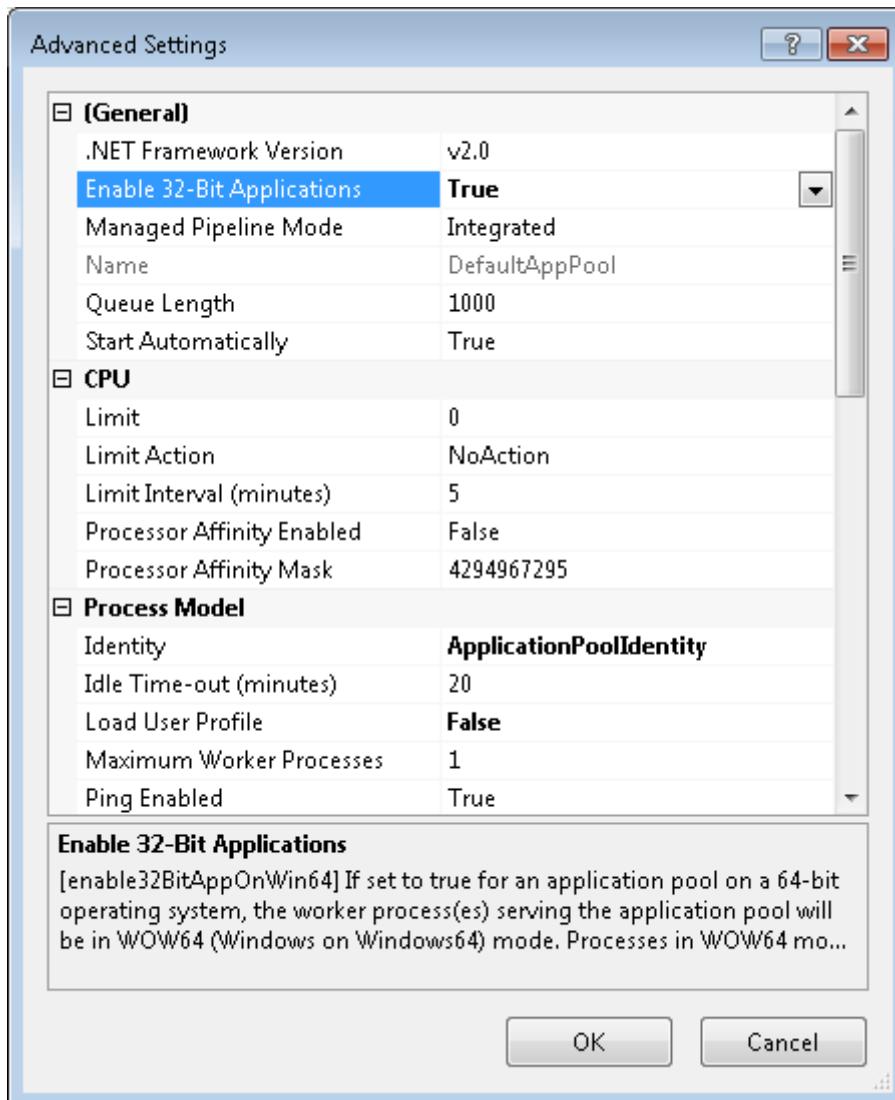


- Restart your system.

If you use Windows 7 64-bit, you need to explicitly enable 32-bit ISAPI-DLLs in IIS. In this case, please perform the following steps:

- Open "Internet Information Service (IIS) Manager" which can be found under "Control Panel\Administrative Tools\"
- Click on "Application Pools"
- Select the "DefaultAppPool" and click on "Advanced Settings..." from the Actions panel

- Set the entry "Enable 32-bit Applications" to "True", then click on "OK" to commit the changes



### Step 3: Testing OPC-XML DA configuration

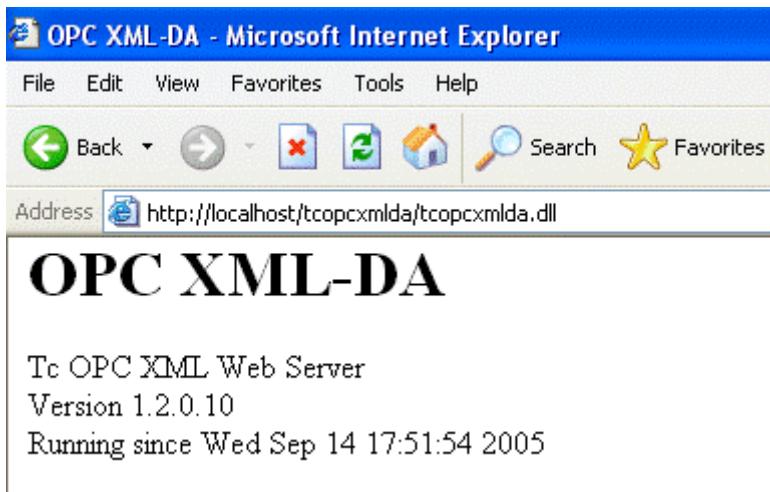
The URL of the OPC-XML DA Server on the PC system can be accessed locally or from a remote computer. In both cases, open the web browser (e.g. Internet explorer) and enter the URL of the OPC-XML DA Server, for example:

**http://<ip-adress or name of PC device>/tcopcxmlda/tcopcxmlda.dll**

Examples:

**http://192.16.17.5/tcopcxmlda/tcopcxmlda.dll**  
**http://localhost/tcopcxmlda/tcopcxmlda.dll**

The OPC-XML DA Server will reply with a status page containing the product version. If you see this page, the installation and configuration of OPC-XML DA has been successful.



In case of problems (like receiving no HTML status data) please check if your system uses a proxy server.

After deactivating the proxy and reloading the URL, the OPC-XML DA Server should reply with the status info above.

### 6.2.3 Status information

This chapter describes nice helpful hints for OPC-XMLDA server (XP and CE)

**OPC specification describes the interfaces which an OPC-XMLDA server have to (MUST) implement.**

**As an addOn the TwinCAT OPC-XMLDA server provides additional "nice to have" functionalities..**

These additional functions are initiated via the URL of the OPC-XML-DA server.

Open the Internet explorer and enter the URL of the OPC Server XML DA on target system like :

**http://<ip-adress or name of device>/TcOpcXmlDa/TcOpcXmlDa.dll<Services>**

**<Services> could be like this :**

- Status information
- Log information
- Reload configuration
- Display configuration
- Restart server
- Stop server

#### 1. Status information

**<Services> = "" (nothing)**

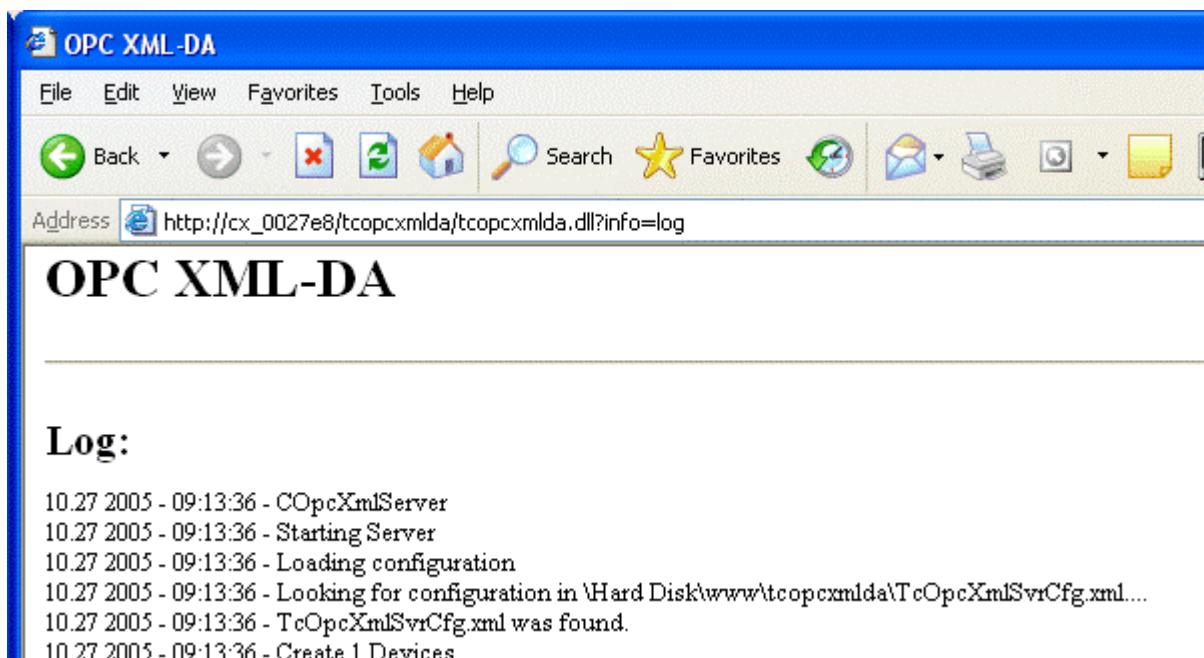
**SAMPLE : http://<ip-adress or name of device>/TcOpcXmlDa/TcOpcXmlDa.dll**

The TcOpcXmlDa server will reply with a status page containing the product version :

#### 2. Log information

**<Services> = ?info=log**

**Sample : http://<ip-adress or name of device>/TcOpcXmlDa/TcOpcXmlDa.dll?info=log**



The screenshot shows a web-based OPC XML-DA configuration tool. The title bar says "OPC XML-DA". The menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". Below the menu is a toolbar with icons for Back, Forward, Refresh, Home, Search, Favorites, and other utilities. The address bar shows the URL "http://cx\_0027e8/tcopcxmlida/tcopcxmlida.dll?info=log". The main content area is titled "OPC XML-DA" and contains a "Log:" section with the following text:

```

Log:
10.27 2005 - 09:13:36 - COpcXmlServer
10.27 2005 - 09:13:36 - Starting Server
10.27 2005 - 09:13:36 - Loading configuration
10.27 2005 - 09:13:36 - Looking for configuration in \Hard Disk\www\tcopcxmlida\TcOpcXmlSvrCfg.xml...
10.27 2005 - 09:13:36 - TcOpcXmlSvrCfg.xml was found.
10.27 2005 - 09:13:36 - Create 1 Devices

```

### 3. Reload configuration file

<Services> = ?info=reload

**Sample :** `http://<ip-adress or name of device>/TcOpcXmlDa/TcOpcXmlDa.dll?info=reload`

### 4. Display configuration

<Services> = ?info=config

**Sample :** `http://<ip-adress or name of device>/TcOpcXmlDa/TcOpcXmlDa.dll?info=config`

Not implemented now, under construction

### 1. Restart of OPC-XMLDA server

<Services> = ?action=restart

**Sample :** `http://<ip-adress or name of device>/TcOpcXmlDa/TcOpcXmlDa.dll?action=restart`

### 2. Stop of OPC-XMLDA server

<Services> = ?action=stop

**Sample :** `http://<ip-adress or name of device>/TcOpcXmlDa/TcOpcXmlDa.dll?action=stop`



Not implemented now, under construction.

## 6.2.4 TwinCAT OPC Server CE : Optimization

This article describes how to minimize the OPC-namespace of TwinCAT-OPC-Server CE.

Take care to NOT load all PLC variables into OPC-Server namespace but instead load that smaller list of PLC variables which are required via OPC in the OPC-client.



CE is designed for small systems with limited resources like CPU and memory. Due to this the optimization of namespace is highly recommended for OPC-Servers running on CE.

In general the opc-server is configured by the OPC-XML configuration file to find the detailed symbol information for each ADS device.

- The option <AutoCfg>7</AutoCfg> is designed to import all variables defined in TPY files  
OPC DA : Limitation today : Only global variables (Strings / ints / Bools..) are imported. Today no structures are imported.  
OPC XMLDA : No limitation
- The option <AutoCfg>8</AutoCfg> is designed to import just a subset of PLC variables marked in the PLC-code to be available for OPC..  
OPC DA : Limitation today : Only global variables (Strings / ints / Bools..) are imported. Today no structures are imported.  
OPC XML : No limitation

In this sample the PLC variable "bMemAlarm"1 will not be available via OPC-Server.

The variable "iReadOnly" is marked to be an OPC-variable and is available for OPC-clients. Additional the OPC-Item Property 5 (defined by OPC-Foundation) is set to "1" which means, the opc-server will handle this variable as "Read-Only". OPC-Server will take care, that OPC-clients can not write to that PLC-variable.

Extra comments in PLC-IEC1131 Editor will modify the configuration of opc-server:

```
bMemFlag1      AT%MX10.0   :  BOOL:  (*~ (OPC:1:available for OPC-clients) *)  
bMemFlag2      AT%MX10.1   :  BOOL:  (*~ (OPC:1:available for OPC-clients) *)  
bMemAlarm1     AT%MX10.2   :  BOOL:  
bMemAlarm2     AT%MX10.3   :  BOOL:  (*~ (OPC:1:available for OPC-clients) *)  
iReadOnly       :  INT:    (*~ (OPC:1:available for OPC-clients)  
                           (OPC_PROP[0005]:1:OPC_PROP_RIGHTS, here ReadOnly) *)
```

## 6.2.5 OPC Client (Diagnostic & Recipe)

This TcOpcClient has been developed to assist in the test and development of our OPC Data Access 1.0 and 2.0 Servers. This text assumes that you are using Windows 95/98 or Windows 2000/NT 4.0 (Service Pack 4 recommended).

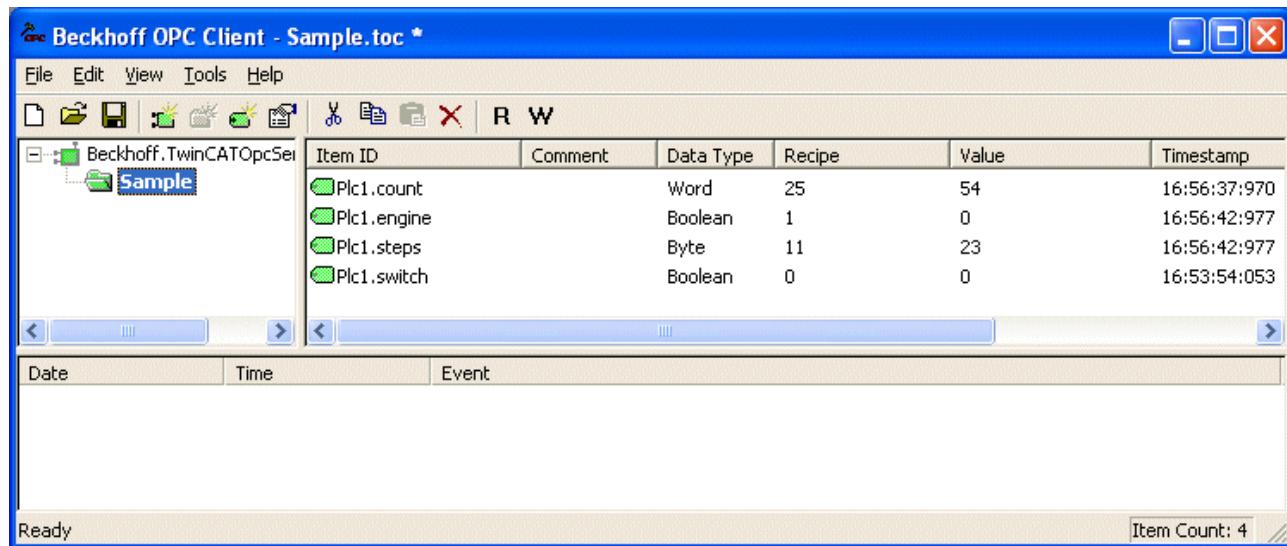
The TcOpcClient supports both local and remote OPC server connections. Remote connections are handled through the operating system's DCOM interface. Windows 95 users should note that Microsoft provides a special update patch that fixes problems with the original COM/OLE logic that shipped with the operating system. In addition, this patch provides the DCOM support necessary for remote server access that was not implemented in the operating system.

### Client Preparation

Begin by creating a [server connection \[► 49\]](#). A server connection is used to connect to an OPC server. Once this connection is made, a [group \[► 52\]](#) can be added to the connection with varying properties such as update rate, dead band, and time bias. Finally, [items \[► 54\]](#) can be added to the individual groups. These items have properties such as an initial active state, data type and access path. Active items will automatically receive data change notifications and will display its current value and quality in the item view.

## 6.2.5.1 The Main Window

Here is a view of the application.

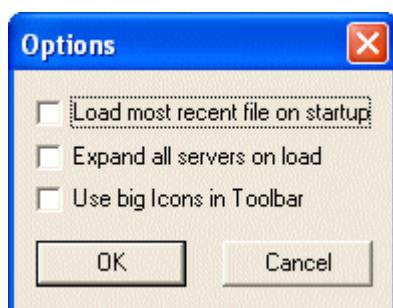


### Title Bar

The application title bar will display the name of the loaded project file. A \* following the name indicates that it is modified.

### Client Configuration

To configure the Beckhoff OPCClient select **Options** from the **Tools** menu:



### Toolbar

- Creates a New project
- Opens an existing project from disk
- Saves the current project to disk
- Create a new server connection [▶ 49]
- Create new group [▶ 52]
- Create new item [▶ 54]
- Edit properties of the selected server, group or item

-  Cut the current selection to the clipboard
-  Copy the current selection to the clipboard
-  Paste a server, group or item(s) from the clipboard
-  Delete the current selection
-  Read recipe values
-  Write recipe values

### Group Window

This window manages a list of [server connections \[▶ 49\]](#) along with their [group \[▶ 52\]](#) objects. Right clicking on a server/group in the list will display a menu of options applicable to the server/group. Double clicking will display the property sheet for the server/group.

### Item Window

This window displays a list of [item \[▶ 54\]](#) information for the selected group. Right clicking on an item name in the list will display a menu of options applicable to the item. Double clicking will display the properties dialog for the item.

Clicking on the Item ID column heading will cause the item list to be sorted based on Item ID. Moving the mouse over the header for the column that is to be changed and dragging the left or right end of the header button until the column is the desired size can change column widths.

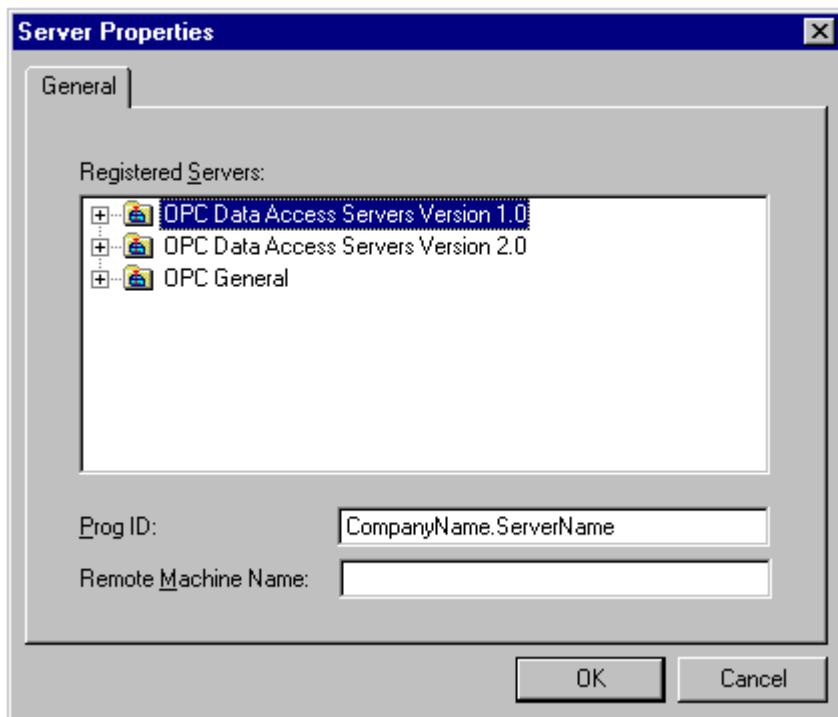
### Message Log

This window logs status messages generated by the application. As with the item window, the column headers may be resized. This window does not support sorting.

## 6.2.5.2 Server Connection

A server connection provides a link between an OPC server and this client. Through this connection, [groups \[▶ 52\]](#) will be added.

Select **NewServer...** from the **Edit** menu or click the toolbar button to create a new server connection using the following property sheet:



## Prog ID

Specify the Prog ID of the OPC Server the client should connect to. You can browse for registered servers by expanding any of the following branches:

OPC Data Access Servers Version 1.0 - Data Access 1.0 servers should self-register and add their Prog ID to this component category. Servers that are properly registered will be listed under this branch.

OPC Data Access Servers Version 2.0 - Data Access 2.0 Servers should self-register and add their Prog ID to this component category. Servers that are properly registered will be listed under this branch.

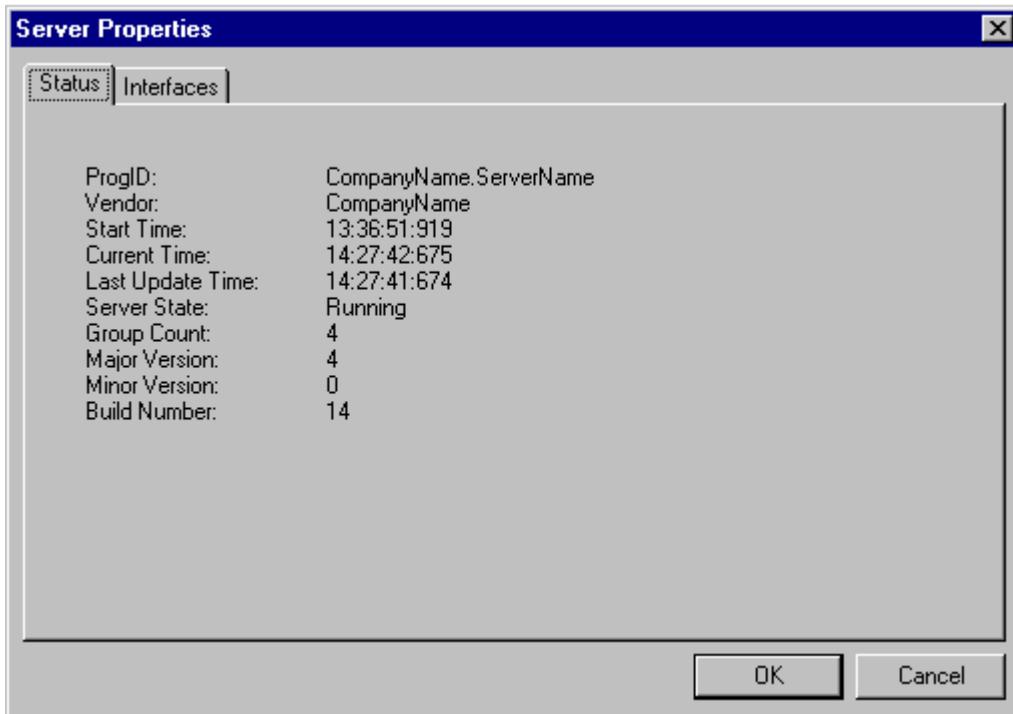
OPC General - All Servers should add a special OPC key in the system registry to be identified as an OPC server. This method allows for older clients and servers to be used (the preferred method is to use component categories). Servers that are registered with this key will be listed under this branch.

Double-clicking on any registered server will automatically update the Prog ID field.

## Remote Machine Name

Specify the name of the machine the OPC server (specified by the Prog ID) resides on. If the server is located on the same machine as the client, leave this field blank.

Once the client makes a successful connection to a server, invoke the server properties by selecting **Properties...** from the **Edit** menu to view the server status and the supported server interfaces.



The server status page details the status of the server. The client queries the status of the server and obtains the following information:

Vendor – A String that specifies vendor specific information.

Start Time - Time the server application was started.

Current Time - Current Time as known by the server.

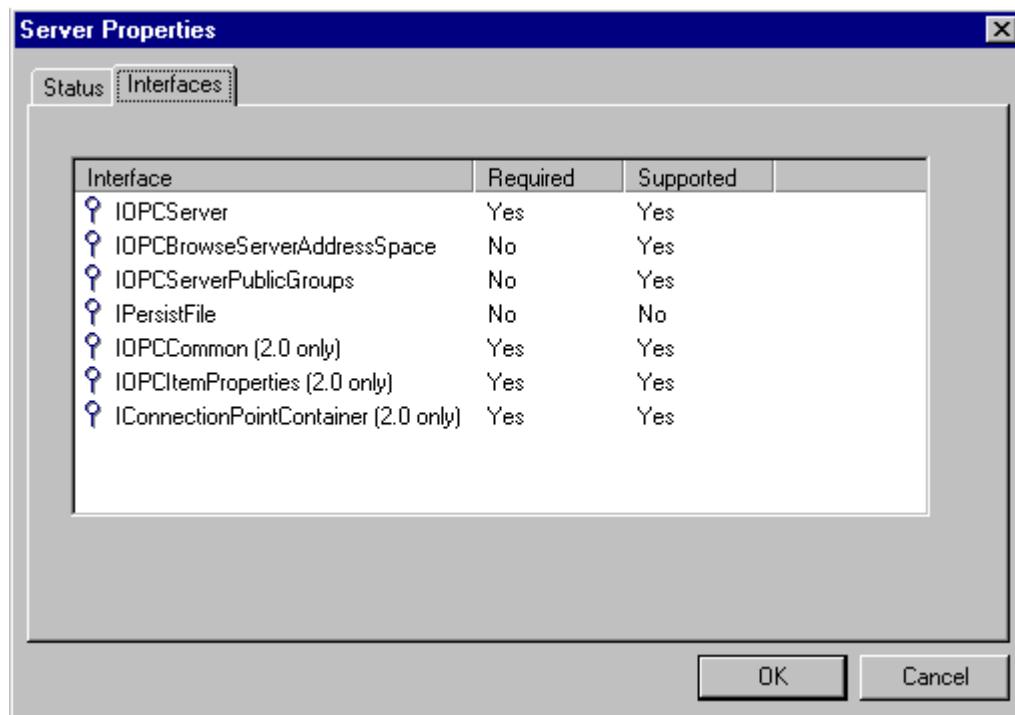
Last Update Time - Time the server sent the last data update to the client on this connection.

Server State - The current state of the server. Valid states are Running, Failure, No Configuration, Suspended, and Test.

Major Version - Major version of the server software.

Minor Version - Minor version of the server software.

Build Number - Build number of the server software.



This interface page lists the interfaces that are required at the server level. Interfaces not required are considered optional. The "supported" column states which interfaces this server supports.

#### Visual/Operational Changes

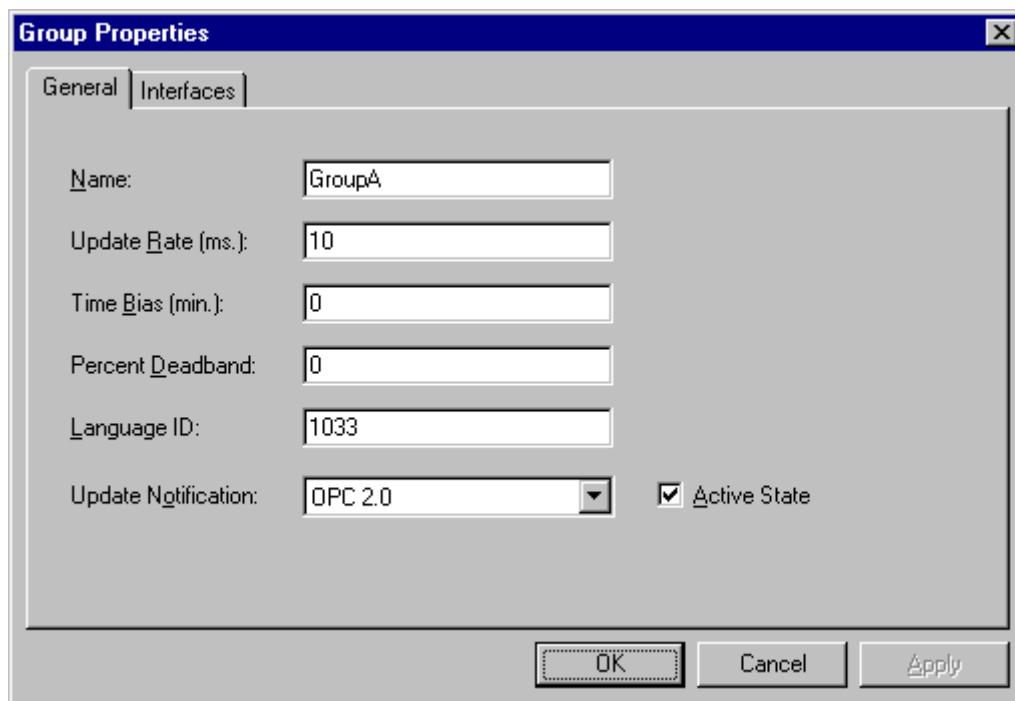
When the TcOpcClient adds, connects, or disconnects a server connection you will notice a change in the appearance of the server connection. The image that appears to the left of each server connection will change to one of the following:

- The client is connected to the server.
- The client is disconnected from the server.
- The client could not connect to the server (the server may not be properly registered).

#### 6.2.5.2.1 Group

A group is used to organize a collection of [items \[▶ 54\]](#) with a common set of properties. The group also specifies the following properties: group name, update rate, time bias, percent dead band, language ID, active state, and the type of data connection that should be made to the server.

Select **NewGroup...** from the **Edit** menu or click the toolbar button to create a new group using the following property sheet:



#### Name

Enter a name to identify the group. If no name is entered, it is the responsibility of the OPC Server to generate a unique name for you.

#### Update Rate

Specifies how often, in milliseconds, the OPC server should provide updates for changing data.

#### Time Bias

Specifies the time difference in minutes between the client/server and the actual device. The information allows the client to convert the timestamp received by the server back to the local time of the device.

#### Percent Dead band

Specifies the percent change in data required to notify the client of a data change.

#### Language ID

Specifies the language to be used by the server when returning values as text for operations on this group.

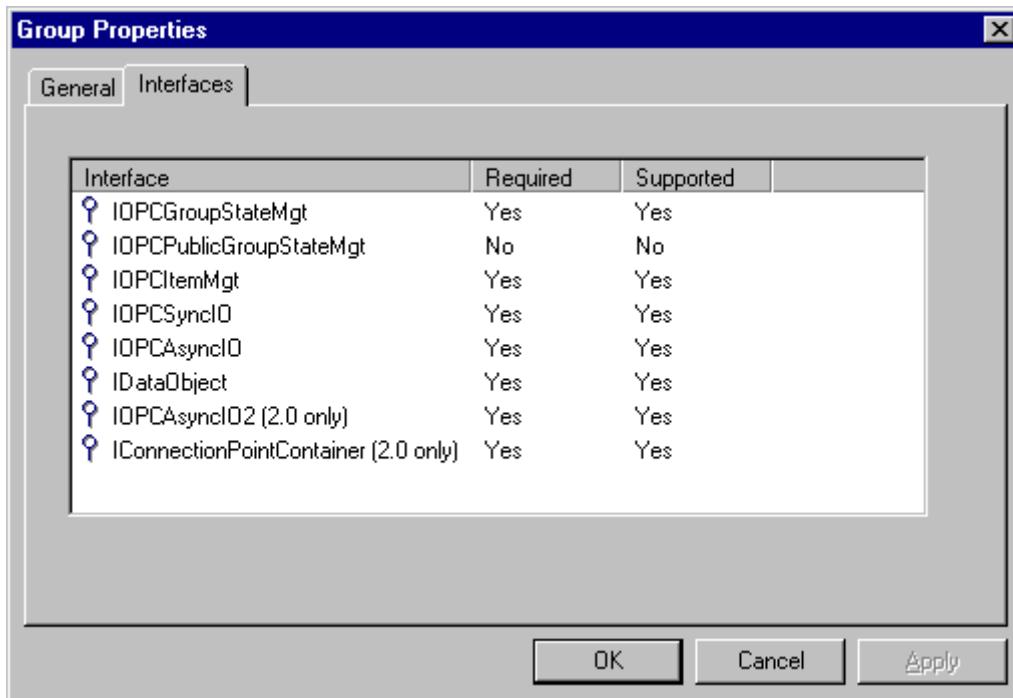
#### Active State

Specifies the active state of the group. When the group is active, the client will receive data change updates for all active items at the specified update rate.

#### Update Notification

Specifies the method to use when receiving data change notifications, or initiating asynchronous requests. Valid selections are OPC 1.0 (No Timestamp), OPC 1.0 (With Timestamp) and OPC 2.0. Data Access 2.0 servers automatically supply timestamps with all data transactions.

When the group has been successfully added to the project, invoke the group properties by selecting **Properties...** from the **Edit** menu to modify group properties and view the supported group interfaces.



This interface page lists the interfaces that are required at the group level. Interfaces not required are considered optional. The supported column states which group interfaces are supported by server.

#### Visual/Operational Changes

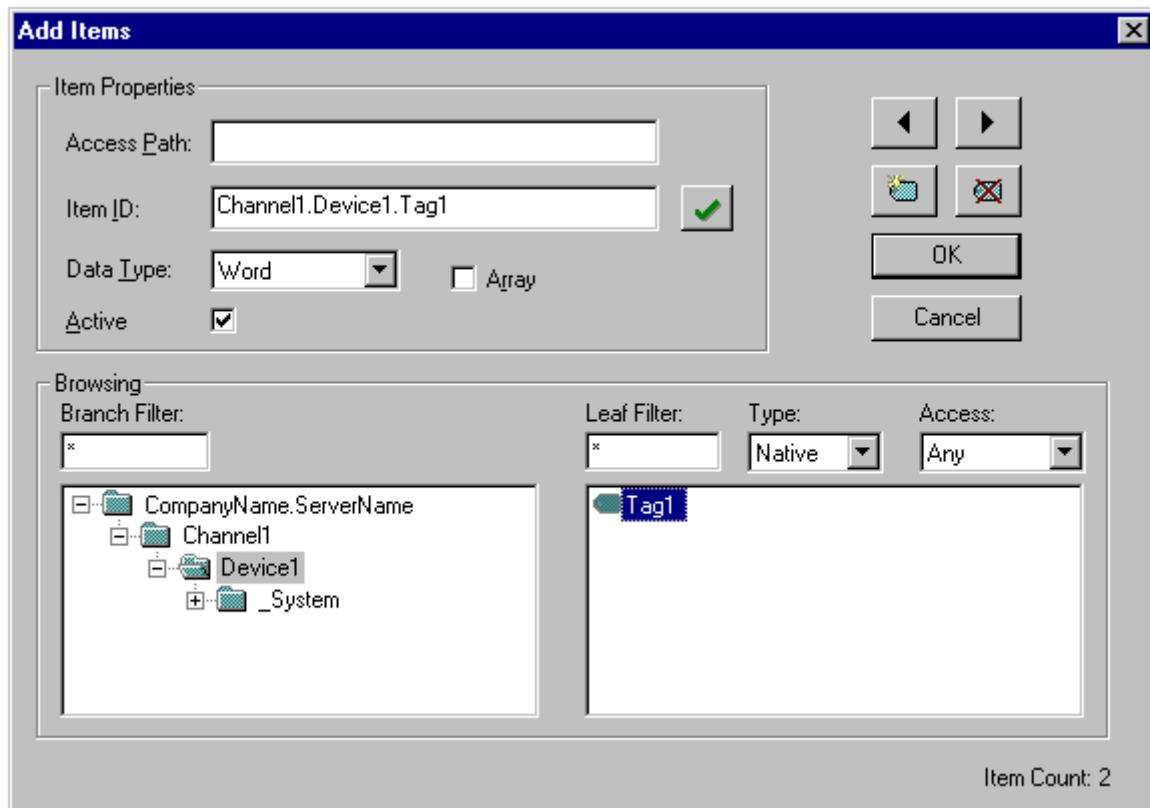
When the TcOpcClient adds, activates or inactivates a group you will notice a change in the appearance of the group. The image that appears to the left of each group will change to one of the following:

- The group is active.
- The group is inactive.
- The client could not add the specified group to the server connection.

#### 6.2.5.2.2 Items

Items represent data that may be accessed via an OPC server. An item specifies the following properties: item ID, access path, requested data type, and active state.

To add an item to a group select **NewItem...** from the **Edit** menu or click on the toolbar button to define an item using the item editor dialog box:



## Access Path

Required by some OPC servers to complete an item definition. Refer to your OPC server's documentation to determine whether you need to specify an access path.

## Item ID

The OPC server item used to reference the data. Refer to your OPC server's documentation to determine valid item names for the location in question. If the server supports tag browsing then you can select an item by using the browse controls.

## Data Type

The requested data type should be used when communicating between the OPC server and the OPC Quick Client. This should be specified to agree with the size and type of the register or memory location that is being addressed. Supported data types are defined as follows:

Data Type	Description
<b>Native</b>	Default as determined by the OPC server
<b>Boolean</b>	Single bit
<b>Char</b>	Signed 8 bit value
<b>Byte</b>	Unsigned 8 bit value
<b>Short</b>	Signed 16 bit value
<b>Word</b>	Unsigned 16 bit value

Data Type	Description
<b>Long</b>	Signed 32 bit value
<b>DWord</b>	Unsigned 32 bit value
<b>Real</b>	Single precision floating point value. (32 bits)
<b>Double</b>	Double precision floating point value (64 bits)
<b>String</b>	Zero terminated character array.

Check the Array box if the data type is an array type.

### Active State

Specifies the initial active state for the item when adding the item. The state can be changed later as described later on in this topic.

### Button Definitions

- ◀ Applies changes to the current tag and fills the dialog with properties of the previous tag in the list.
- ▶ Applies changes to the current tag and fills the dialog with properties of the next tag in the list. If no next tag exists, the client attempts to auto-increment the current item ID.
-  Creates a new item definition.
-  Deletes the current item definition.
-  Validates the current item definition.

### Browsing

If the server supports browsing, the browsing controls will be available for use. Browse the branch list to search for selected items. Apply appropriate filtering to simplify the search.

### Visual/Operational Changes

When the TcOpcClient adds, activates or inactivates an item you will notice a change in the appearance of the item. The image that appears to the left of each item will change to one of the following:

-  The item is active.
-  The item is inactive.
-  The client could not add the specified item to the group.

## Recipes

For each item, a recipe value may be specified, by selecting **Set recipe** from the **Tools|Item** menu. You may also choose to read recipes for a whole group, by choosing **Read recipes** from the **Tools|Group** menu. Later on you may write the whole recipe for one group down to the PLC. Reading and writing of recipes may be accelerated by selecting a group and pressing **<Strg + Leftarrow>** for reading or **<Strg + Rightarrow>** for writing Group recipes.

## Comments

Each Item has associated with it a comment. You can use this value to store any item specific data as you desire.

### 6.2.5.3 Operations

#### Server Connection Operations

##### Connect

Select **Connect** from the **Tools|Server** menu to reestablish a connection to the server. All groups and items will also be added.

##### Disconnect

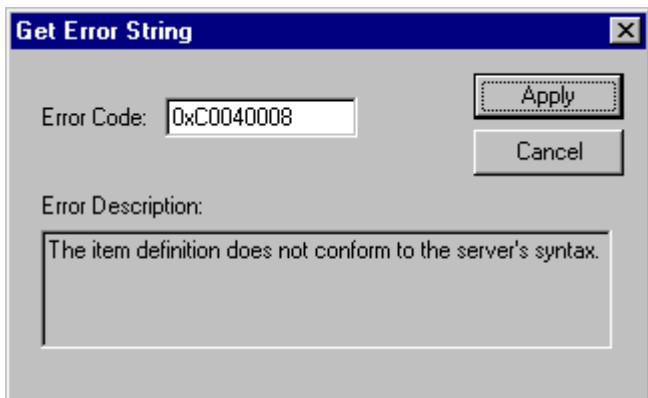
Select **Disconnect** from the **Tools|Server** menu to disconnect from the server.

##### Reconnect

Select **Reconnect** from the **Tools|Server** menu to reestablish a connection to the server.

##### Get Error String

Select **Get Error String...** from the **Tools|Server** menu to obtain an error description for any valid error codes.

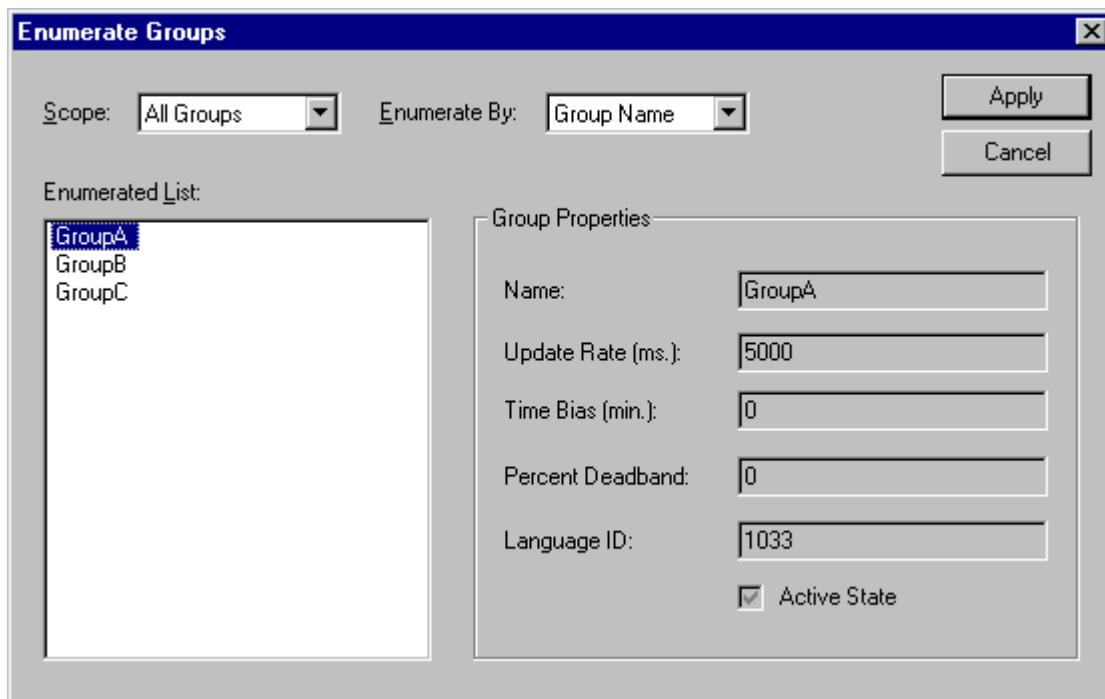


To request an error description from the server, enter the error code and press **Apply**. If the server can translate the error code, the description will appear. An OPC server should be able to translate any OPC error code, as well as Win32 errors.

This operation tests the following server functionality: `IOPCServer::GetErrorString()`.

##### Enumerate Groups

Select **Enumerate Groups...** from the **Tools|Server** menu to enumerate groups available to the server connection.



Groups can be enumerated two different ways. The server can be requested to enumerate groups by name or by object. Enumerating by name, the server passes back an enumeration of strings each containing the name of a group. Enumerating by object, the server passes back an enumeration of group interface pointers (IUnknown), which the client can make use of, by querying for other interfaces. Either way, the client uses both mechanisms to obtain the same results, a list of group names along with the properties of each group. The client can request only private groups (groups this client added to the server connection), public groups (groups that may have been added by another client but made available to us) or all groups be enumerated.

This operation tests the following server functionality: IOPCServer::CreateGroupEnumerator ( ), IOPCServer::GetGroupByName ( ), IOPCGroupStateMgt::GetState ( ).

### Get Group By Name

Select **Get Group ByName...** from the **Tools|Server** menu to obtain group properties by name.

To request group properties by name; enter the name of the group and press Apply. If the server can resolve the group name, the client will proceed to obtain the properties for that group.

This operation tests the following server functionality: IOPCServer::GetGroupByName ( ), IOPCGroupStateMgt::GetState ( ).

### Group Operations

#### Clone Group

Select **Clone Group** from the **Tools|Group** menu to clone the selected group. An OPC server should create a new group, with a unique name, with identical properties to the group that is being cloned. If the cloned group contains items, the items should also be cloned, again with the same properties. The active state of the cloned group and items should be set to an inactive state.

This operation tests the following server functionality: IOPCGroupStateMgt::CloneGroup, SetState ( ), IOPCItemMgt::CreateEnumerator, SetClientHandles ( ), IEnumOPCItemAttributes::Reset, Next ( ).

### Item Operations

#### SetActive/Inactive

Select **SetActive/Inactive** from the **Tools|Item** menu to activate/inactivate the selected item(s). Only active items should receive on data change updates (the group must also be set active)

### Synchronous Read (Cache/Device)

Select **Synchronous CacheRead** or **SynchronousDevice Read** from the **Tools|Item** menu to perform a synchronous read from cache/device, respectively, on the selected items.

### Asynchronous Read (Cache/Device)

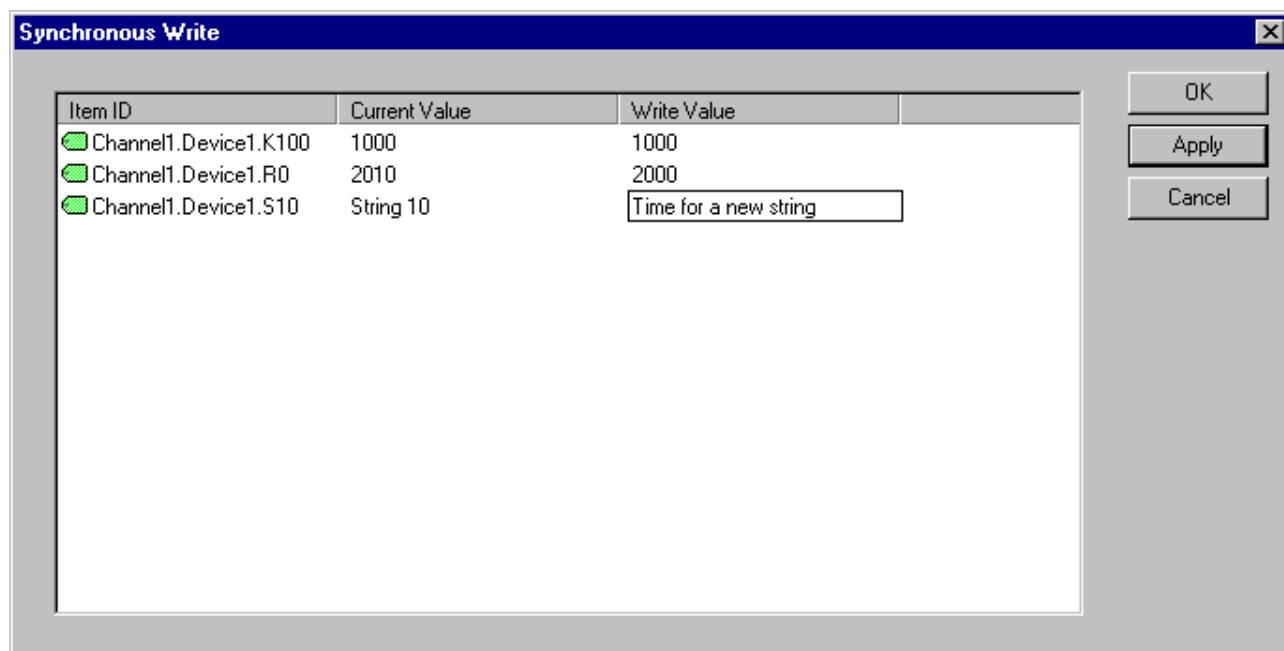
Select **Asynchronous 1.0/2.0 Cache Read** or **Asynchronous 1.0/2.0 Device Read** from the **Tools|Item** menu to perform an asynchronous read from cache/device, respectively, on the selected items. Reads using the older method, 1.0, uses IAdviseSink to return read complete notifications. The newer method, 2.0, uses connection points to exchange data between client and server.

### Asynchronous Refresh (Cache/Device)

Select **Asynchronous 1.0/2.0 Cache Refresh** or **Asynchronous 1.0/2.0 Device Refresh** from the **Tools|Item** menu to perform an asynchronous refresh from cache/device on the respective selected items. Refreshes using the older method, 1.0, uses IAdviseSink to return refresh notifications. The newer method, 2.0, uses connection points to exchange data between client and server.

### Synchronous/Asynchronous Write

Select **Synchronous Write** or **Asynchronous 1.0/2.0 Write** from the **Tools|Item** menu to perform a synchronous/asynchronous write on the selected items.



Both types of writes utilize the same dialog. The dialog displays the current value at a 100-millisecond refresh rate. Enter the data to write in the write value column and press Apply to send the data to the server. Pressing "OK" applies the write and closes the dialog. Press Cancel to abort the operation.

### Security Operations

You may choose to protect administrative operations on a configuration with an admin password. To do so, select tools->SetPwd from the menu. In the Following Dialog you are prompted to enter the current and new Administration passwords.



As soon as you switch to a non-blank password, any administrative operation on the document will be prohibited. To perform administrative tasks select tools->login and enter the password.

## 7 OPC Client sample code

### 7.1 OPC .NET client : API sample for read / write

<https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162184203/.zip>

To read or write variables from Beckhoffs TwinCAT OPC-DA-Server via an C# application perform the following simple steps

- Download and install the Opc.NET Api from the [OPC Foundation website](#)
- Start VisualStudio 2005 and create a new C# console application
- Go to the Solution Explorer, and add references to  
    OpcNetApi.dll OpcNetApi.Com.dll
- Use the following code to read and write variables

```
static void Main(string[] args)
{
    // Create a server object and connect to the TwinCATOpcServer
    Opc.URLurl = newOpc.URL("opcda://localhost/BECKHOFF.TwinCATOpcServerDA");
    Opc.Da.Serverserver = null;
    OpcCom.Factoryfact = newOpcCom.Factory();
    server = newOpc.Da.Server(fact, null);
    server.Connect(url, newOpc.ConnectData(newSystem.Net.NetworkCredential()));

    // Create a group
    Opc.Da.Subscriptiongroup;
    Opc.Da.SubscriptionStategroupState = newOpc.Da.SubscriptionState();
    groupState.Name = "Group";
    groupState.Active = false;
    group = (Opc.Da.Subscription)server.CreateSubscription(groupState);

    // add items to the group
    Opc.Da.Item[] items = newOpc.Da.Item[3];
    items[0] = newOpc.Da.Item();
    items[0].ItemName = "PLC1.Value1";
    items[1] = newOpc.Da.Item();
    items[1].ItemName = "PLC1.Value2";
    items[2] = newOpc.Da.Item();
    items[2].ItemName = "PLC1.Value3";
    items = group.AddItems(items);

    // write items
    Opc.Da.ItemValue[] writeValues = newOpc.Da.ItemValue[3];
    writeValues[0] = newOpc.Da.ItemValue();
    writeValues[1] = newOpc.Da.ItemValue();
    writeValues[2] = newOpc.Da.ItemValue();
    writeValues[0].ServerHandle = group.Items[0].ServerHandle;
    writeValues[0].Value = 0;
    writeValues[1].ServerHandle = group.Items[1].ServerHandle;
    writeValues[1].Value = 0;
    writeValues[2].ServerHandle = group.Items[2].ServerHandle;
    writeValues[2].Value = 0;
    Opc.IRequestreq;
    group.Write(writeValues,
    321, newOpc.Da.WriteCompleteEventHandler(WriteCompleteCallback), outreq);

    // and now read the items again
    group.Read(group.Items, 123, newOpc.Da.ReadCompleteEventHandler(ReadCompleteCallback), outreq);
    Console.ReadLine();
}

static void WriteCompleteCallback(object clientHandle, Opc.IdentifiedResult[] results)
{
    Console.WriteLine("Write completed");
    foreach (Opc.IdentifiedResult writeResult in results)
    {
        Console.WriteLine("\t{0} write result: {1}", writeResult.ItemName, writeResult.ResultID);
    }
    Console.WriteLine();
}

static void ReadCompleteCallback(object clientHandle, Opc.Da.ItemValueResult[] results)
```

```
{
Console.WriteLine("Read completed");
foreach(Opc.Da.ItemValueResultreadResultinresults)
{
Console.WriteLine("\t{0}\tval:{1}", readResult.ItemName, readResult.Value);
}
Console.WriteLine();
}
```

## 7.2 OPC .NET client : API sample for subscriptions

<https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162185611/.zip>

To create a subscription from your C# application perform the following steps:

- Download the Opc .Net API from the [OPC Foundation website](#) (Go to: Resources > Samples and Tools > Classic)
- Start Visual Studio 2005 and create a new C# console application
- Navigate to the Solution Explorer and add references to the  
OpcNetApi.dll OpcNetApi.Com.dll
- Use the following code to create your subscriptions.

```
static void Main(string[] args)
{
// Create a server object and connect to the TwinCATOpcServer
Opc.URL url = new Opc.URL("opcda://localhost/BECKHOFF.TwinCATOpcServerDA");
Opc.Da.Server server = null;
OpcCom.Factory fact = new OpcCom.Factory();
server = new Opc.Da.Server(fact, null);

server.Connect(url, new Opc.ConnectData(new System.Net.NetworkCredential()));

// Create a group
Opc.Da.Subscription group;
Opc.Da.SubscriptionState groupState = new Opc.Da.SubscriptionState();
groupState.Name = "Group";
groupState.Active = true;
group = (Opc.Da.Subscription)server.CreateSubscription(groupState);

// add items to the group.
Opc.Da.Item[] items = new Opc.Da.Item[3];
items[0] = new Opc.Da.Item();
items[0].ItemName = "PLC1.dyn_R8[1]";
items[1] = new Opc.Da.Item();
items[1].ItemName = "PLC1.dyn_R4[1]";
items[2] = new Opc.Da.Item();
items[2].ItemName = "PLC1.dyn_I4[1]";

items = group.AddItems(items);

group.DataChanged += new Opc.Da.DataChangedEventHandler(OnTransactionCompleted);
Console.ReadLine();
}

static void OnTransactionCompleted( object group, object hReq, Opc.Da.ItemValueResult[] items)
{
Console.WriteLine("----->");
Console.WriteLine("DataChanged ...");
for (int i = 0; i < items.GetLength(0); i++)
{
Console.WriteLine("Item DataChange - ItemId: {0}", items[i].ItemName);
Console.WriteLine(" Value: {0,-20}", items[i].Value);
Console.WriteLine(" TimeStamp: {0:00}:{1:00}:{2:00}.{3:000}",
items[i].Timestamp.Hour,
items[i].Timestamp.Minute,
items[i].Timestamp.Second,
items[i].Timestamp.Millisecond);
}
Console.WriteLine("-----<");
}
```

## 7.3 OPC client : AJAX (Asynchronous Java XML)

<https://infosys.beckhoff.com/content/1033/tcopcserver/Resources/12162187019/.zip>

## 8 Appendix

### 8.1 OPC Compliance Certification

The OPC foundation offers members of the organisation an compliance test tool. This tool tests the functionalities and interfaces of the OPC server. For OPC clients is no test tool available.

Testing a server should be done by competent personal only, besides the results could be forged.

**Example:**

**In one test case, the OPC compliance tool writes variables in the OPC Server, reads them afterwards back and compares the values.**

**If these test variables would be PLC variables, which changes cyclically in the PLC, these test case would be failed.**

**Compliance certification**

These and further current OPC certificates are listed on the web site of the OPC foundation:

<http://www.opcfoundation.org>

## OPC Data Access 2.05a Compliance Test Result

<b>Server:</b>	Beckhoff.TwinCATOPCServerDA (Local Server)
<b>Compliance Test Result:</b>	Compliant
<b>Date:</b>	11/6/2003 11:59:31 PM
<b>Company:</b>	Beckhoff Industrie Electronic
<b>OS version:</b>	Windows NT 5.1.2600 Service Pack 1
<b>Version info:</b>	<i>Compliance Test Tool: V 2.0.2.1105 OPC Sever Major Version: 4 OPC Sever Minor Version: 1 OPC Sever Build Number: 27 OPC Sever Vendor Info: Beckhoff TwinCAT OPC Server4</i>
<b>Optional interfaces (supported):</b>	IOPCBrowseServerAddressSpace IPersistFile
<b>Optional interfaces (unsupported):</b>	IOPCServerPublicGroups IOPCPublicGroupStateMgt
<b>Data types for logical test (available):</b>	VT_BOOL VT_I2 VT_I4 VT_R4 VT_R8 VT_BSTR VT_UI1 VT_I1 VT_UI2 VT_UI4 SAFEARRAY OF VT_I2 SAFEARRAY OF VT_I4 SAFEARRAY OF VT_R4 SAFEARRAY OF VT_R8 SAFEARRAY OF VT_UI1 SAFEARRAY OF VT_UI2 SAFEARRAY OF VT_UI4
<b>Data types for logical test (unavailable):</b>	VT_DATE VT_CY SAFEARRAY OF VT_BOOL SAFEARRAY OF VT_DATE SAFEARRAY OF VT_BSTR SAFEARRAY OF VT_I1 SAFEARRAY OF VT_CY
<b>Number of supported groups:</b>	>= 10
<b>Not available for test:</b>	
<b>Number of test items:</b>	2 Readable items 2 Writeable items 54 Read/Writeable items

## 8.2 DCOM

This Help File was designed to give the user of component's communication through DCOM (specially OPC users) an idea on possible settings in an industrial environment. This Help File just shows possible settings of DCOM security that will make the system running. If the manufacturers or vendors of OPC products provide their own manuals, these manuals should be used instead of this Help File.

### Important Notes

Before changing any settings of DCOM security, a system recovery procedure should be prepared. All settings described in this documentation should be tested in a laboratory environment, before changing machines in production.

- Save system partition including Windows Registry (Image Software)
- Only local administrators are allowed to change DCOM settings

- Test all settings in a laboratory environment before changing the production machine

This documentation deals with COM components that should communicate with each other. There are three different types of COM -Servers known, depending on their operational environment.

- **INPROG- Server**

The COM-Server runs in the memory area of the Client, the Server is a DLL and runs local on the same machine and in the same process.

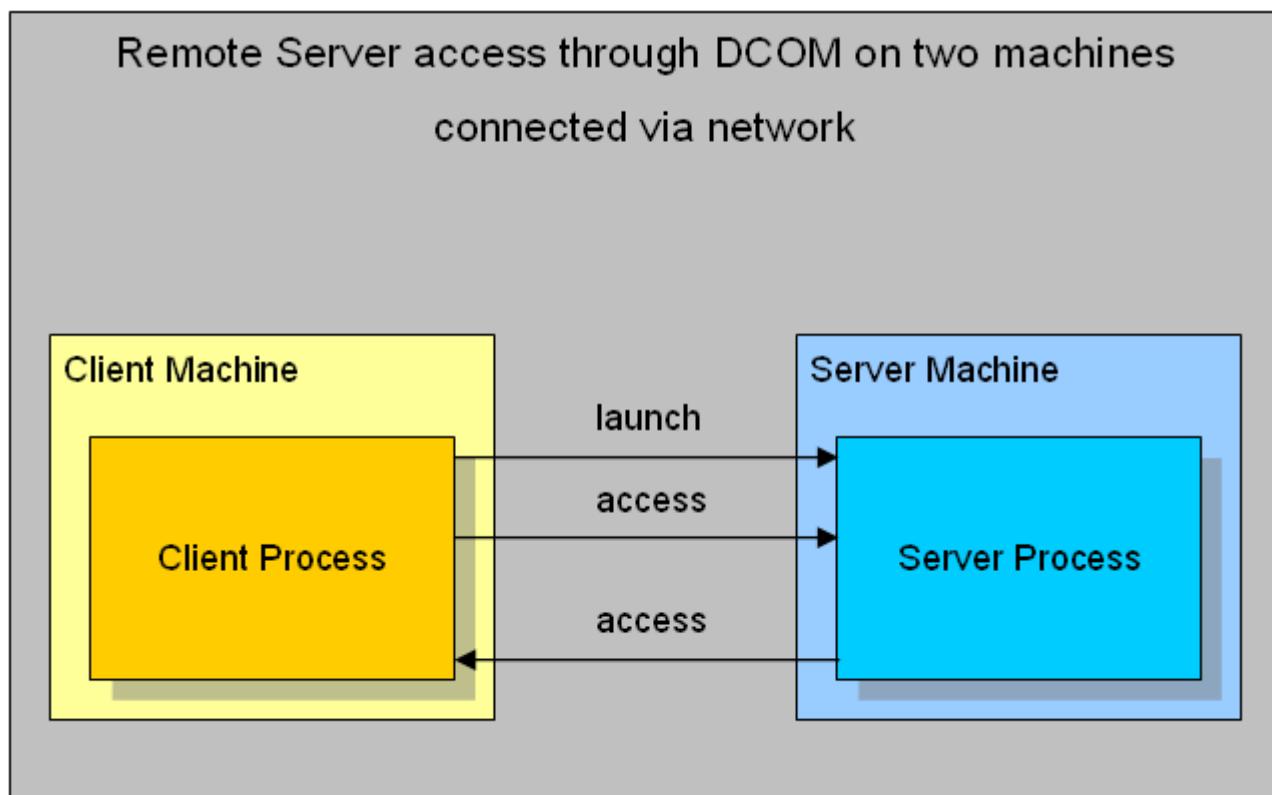
- **LOCAL-Server**

The COM Server runs in its own memory area, the Server is a EXE and runs local on the same machine, but in its own process

- **REMOTE-Server**

The COM Server runs in its own memory area, the Server is a EXE and runs remotely on a different machine and (of course) in its own process as the Client

The communication between Client and Server process follows the rules of COM. Whenever the Client is located on one machine and the Server on another, DCOM (Distributed COM) comes into place. The function calls between Client and Server process are checked for correct security by the operating system. The security settings for DCOM have NOTHING to do with sharing folders between two machines or network shares across a network. When it comes to DCOM-Security we talk about "inter process communication", in other words: the right to start or access a certain component.



To keep configuration simple it is assumed that two machines have the same operating system installed, are both in the same Workgroup and have the same user accounts (same person/PWD actually logged on). On the Server Machine an OPC Server is located and on the Client Machine an OPC Client was installed.

## 8.2.1 Prerequisites

### 8.2.1.1 Operating System

Depending on the operating system of the OPC-Server and OPC-Client computer, the user accounts which are used for communication will be identified differently.

## Windows 2000

This section describes the required settings for Windows 2000 operating system. On Windows 2000 SP 1, there are some known bugs related to proper callback operation. DCOM servers installed on a Win2k SP1 machine sporadically stop sending callbacks (e.g. OnDataChange) after days or weeks. The error code 0x80010108 (RPC\_E\_DISCONNECTED) is returned by the DCOM Server when trying to fire a callback. All other calls from the Client to the Server still work fine.

There are three ways to overcome this problem:

- release the callback object and subscribe again (recreate the callback object)
- install COM + Rollup Package 18.1 (Post Service Pack 2)
- install Service Pack 3 (or higher) for Windows 2000

## Windows XP

This section describes the required settings for Windows XP operating system only. The default installation for XP forces remote users to authenticate as Guest. This means that DCOM clients cannot connect to a server running on an XP machine unless the Guest account is enabled and has enough rights to launch the server. On the other hand when a DCOM server fires a callback to a remote Client installed on an XP machine, the authentication will be "changed back" to the Guest account (which is mostly disabled by default). Thus, the callback (e.g. OnDataChange) will never get though to the Client. To force an XP machine to "behave" like a Windows 2000 computer the *Network Access* should be changed to **Classic**

Open the Security Options dialog with: *START >> Control Panel >> Administrative Tools >> Local Security Policy >> Local Policies >> Security Options*. Find the following entry: *Network access: Sharing and security model for local accounts* and change this setting to: **Classic - users authenticate as themselves**

-  In a mixed configuration (e.g. Client installed on XP and Server installed on Windows 2k) the XP machine automatically "changes" to the classic Win2k behavior when launching and accessing the DCOM Server, but there will be no callbacks coming through.

## Windows 7

This section describes the required settings for Windows 7 operating system only. The default installation for Windows 7 forces remote users to authenticate as "Guest". This means that DCOM clients cannot connect to a server running on an Windows 7 computer unless the Guest account is enabled and has enough rights to launch the server. On the other hand, when a DCOM server fires a callback to a remote Client installed on an Windows 7 machine, the authentication will be "changed back" to the Guest account (which is mostly disabled by default). Thus, the callback (e.g. OnDataChange) will never get though to the Client. To force a Windows 7 computer to "behave" like a Windows 2000 computer, the *Network Access* should be changed to **Classic**

Please perform the following steps on the Windows 7 computer:

- Open the Security Options dialog with: *START >> Control Panel >> Administrative Tools >> Local Security Policy >> Local Policies >> Security Options*.
- Find the following entry: *Network access: Sharing and security model for local accounts* and change this setting to: **Classic - users authenticate as themselves**

-  In a mixed configuration (e.g. a client installed on Windows 7 and the server installed on Windows 2000), the Windows 7 machine automatically "changes" to the classic Win2k behavior when launching and accessing the DCOM Server, but there will be no callbacks coming through.

### 8.2.1.2 Network Configuration

#### 8.2.1.2.1 Network Configuration

Depending on the configuration of the network different identification of Users will be performed by the operating system. Considerations regarding this issues are divided into the following topics.

- [Workgroup \[► 68\]](#)

- [Domain \[► 68\]](#)

### 8.2.1.2.2 Workgroup

For proper DCOM security settings it is essential to identify the configuration of the Client and the Server machine. If both machines are members of the same workgroup there will be no central point for user authentication. This means that every single machine must have all information on all users that should be able to remote access this node (local identification). The administrative effort will increase immensely when adding new users.

- **Workgroup**

The Client and Server Machine should be member of the same Workgroup.

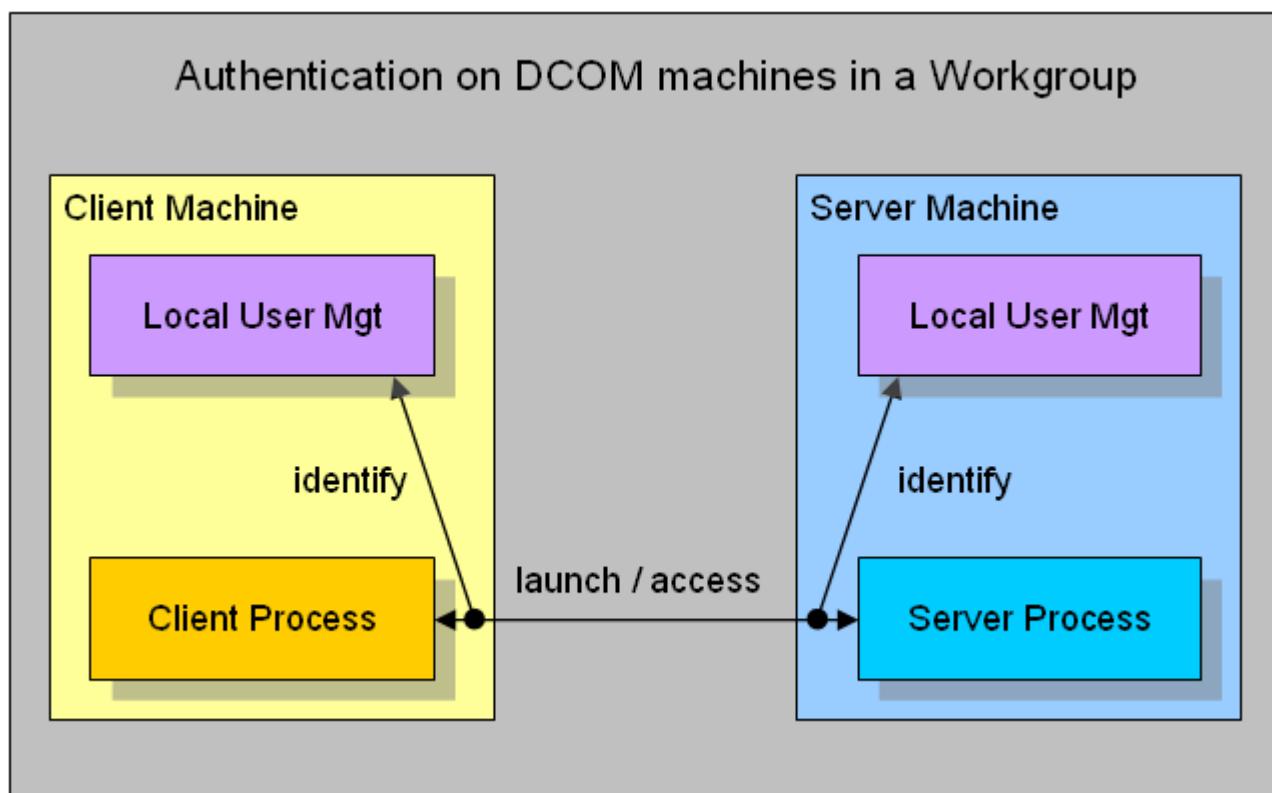
- **Users Accounts**

As authentication is performed locally on every single machine, the User Accounts (Name and PWD) should be identical on the Client and the Server Machine.

- **Operation System**

The Operation System on the Client and the Server Machine should be from the same family (all NT, all 2K or all XP). When doing "mixed configuration" certain (OS specific) settings must be taken into account.

To keep configuration simple it is assumed that two machines have the same operating system installed, are both in the same Workgroup and have the same user accounts (same person/PWD actually logged on). On the Server Machine an OPC Server is located and on the Client Machine an OPC Client was installed.



### 8.2.1.2.3 Domain Controller

For proper DCOM security settings it is essential to identify the configuration of the Client and the Server machine. If both machines are members of the same domain there will be one central point for user authentication. If both machines are in different domains, these domains must trust each other. The administrative effort will decrease because new users will only be added to the domain.

- **Domain**

The Client and Server Machine should be member of the same Domain. Different domains must trust each other.

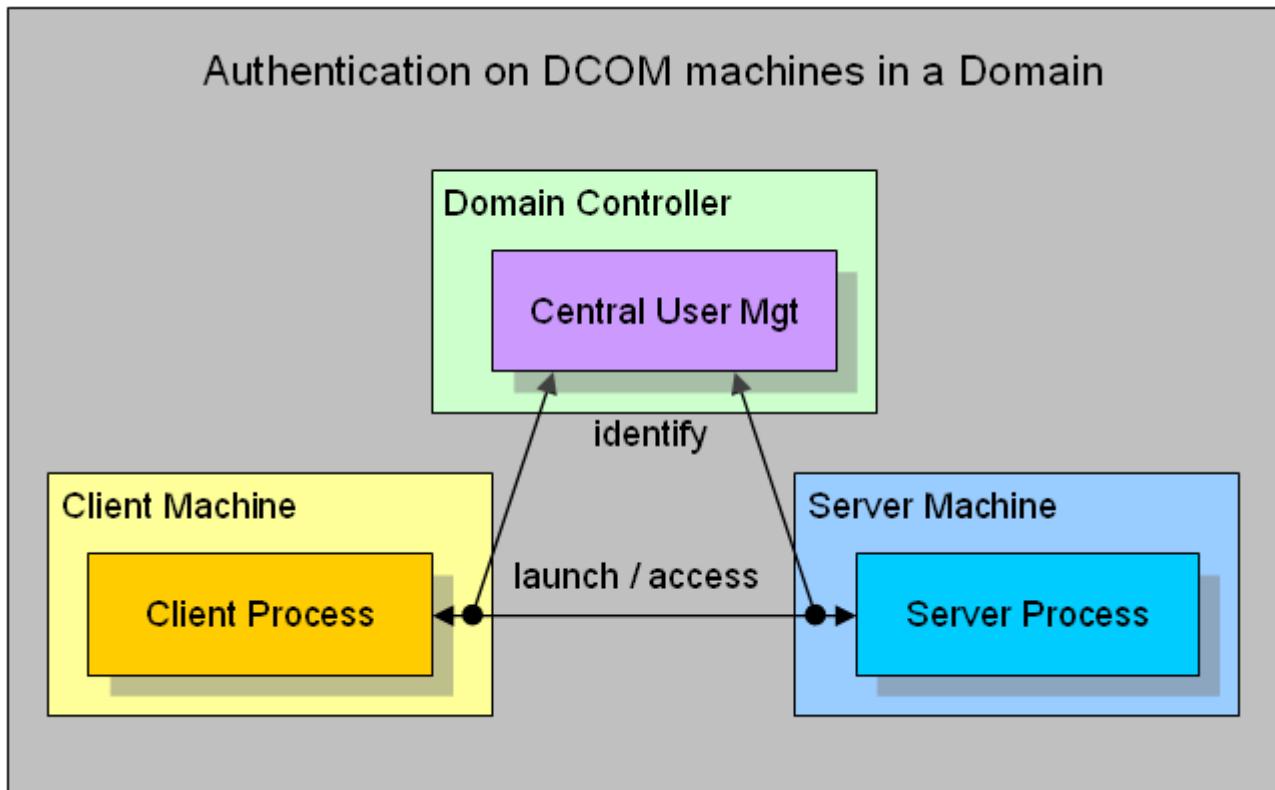
- Users Accounts**

Authentication is performed on the domain machine, the User Accounts (Name and PWD) or groups are used in the DCOM settings of the Client and the Server Machine.

- Operation System**

The Operation System on the Client and the Server Machine should be from the same family (all NT, all 2K or all XP). When doing "mixed configuration" certain (OS specific) settings have to be taken into account.

To keep configuration simple it is assumed that two machines have the same operating system installed, are both in the same Domain and have different user accounts logged on. The different users are members of one User Group. This for this User Group access is granted in the DCOM settings of the Client and the Server Machine.



In a mixed configuration (e.g. Server Machine in a Domain and Client Machine in a Workgroup) the so called double identification should be done. This means to add User Accounts to the Workgroup Machine which are identically (Name and PWD) the same as the User Accounts on the Domain. By this function calls to the Workgroup Machine can (locally) be identified and granted without "asking" the Domain Controller.

### 8.2.2 Client

This section describes the required settings on the client's side. The client is the computer on which the OPC-Client application, e.g. a visualization, is running. Usually the OPC-Server is also located on this computer. However, in some environments it may be necessary that OPC-Client and OPC-Server need to be installed on different computers. Both, the OPC-Server's and OPC-Client's DCOM settings need to be configured, so that a remote communication between client and server is possible.



The following settings have been tested on Windows 2000, Windows XP and Windows 7 computers.

## Step 1: General network configuration

Depending on the client's operating system, some additional network settings need to be taken. Basically, the same settings must be performed. However, some "operating system specific" settings must be done to get the DCOM security running.

Please refer to our [Article about Operating Systems \[▶ 66\]](#) for more information.

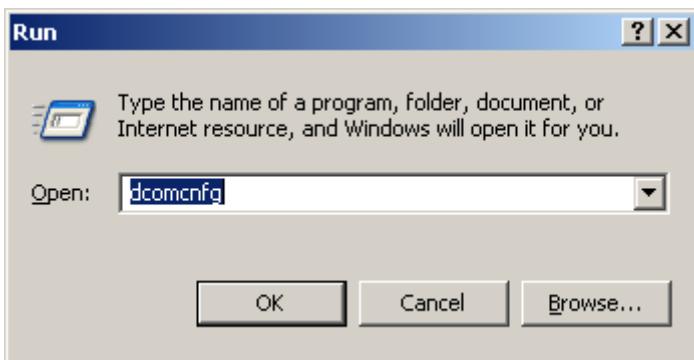
## Step 2: DCOM configuration

To configure DCOM for a remote OPC communication, please perform the following steps on the computer running the OPC-Client.



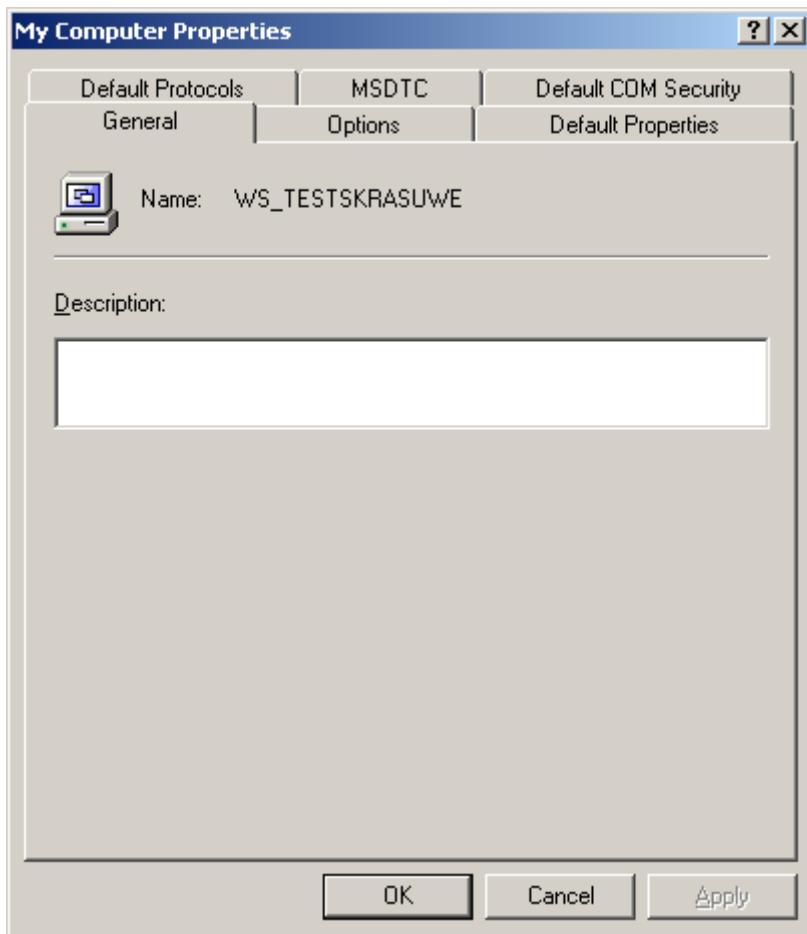
Only local administrators are allowed to open and change the DCOM security.

- Open *Start --> Run --> dcomcnfg.exe* to start the DCOM configuration dialog.

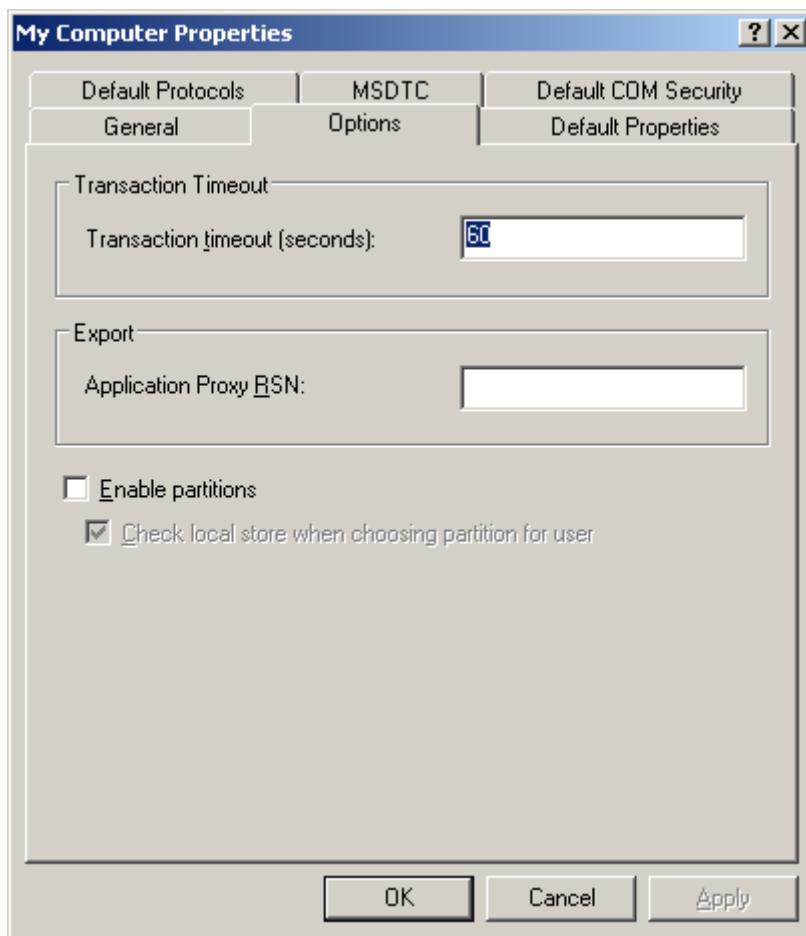


- Navigate to *Console Root --> Component Services --> Computers --> My Computer*
- Select "My Computer", right click it and select **Properties**

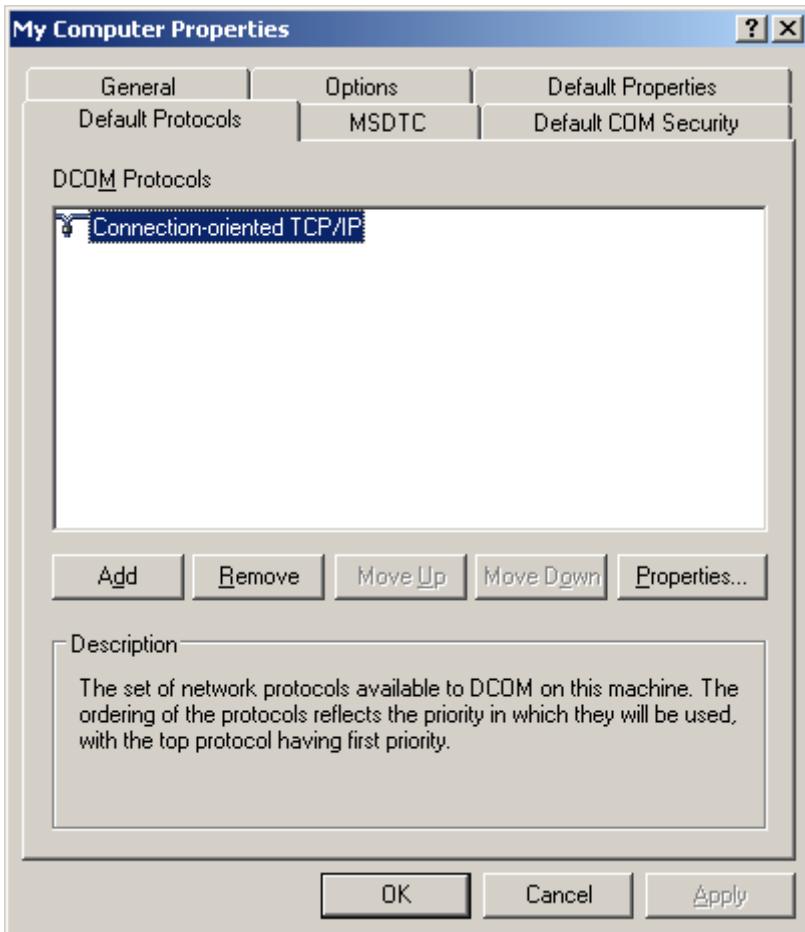
- On the "General" tab no changes must be made. The default settings will be correct for OPC Client side security settings



- On the "Options" tab no changes have to be made. The default settings will be correct for OPC Client side security settings.

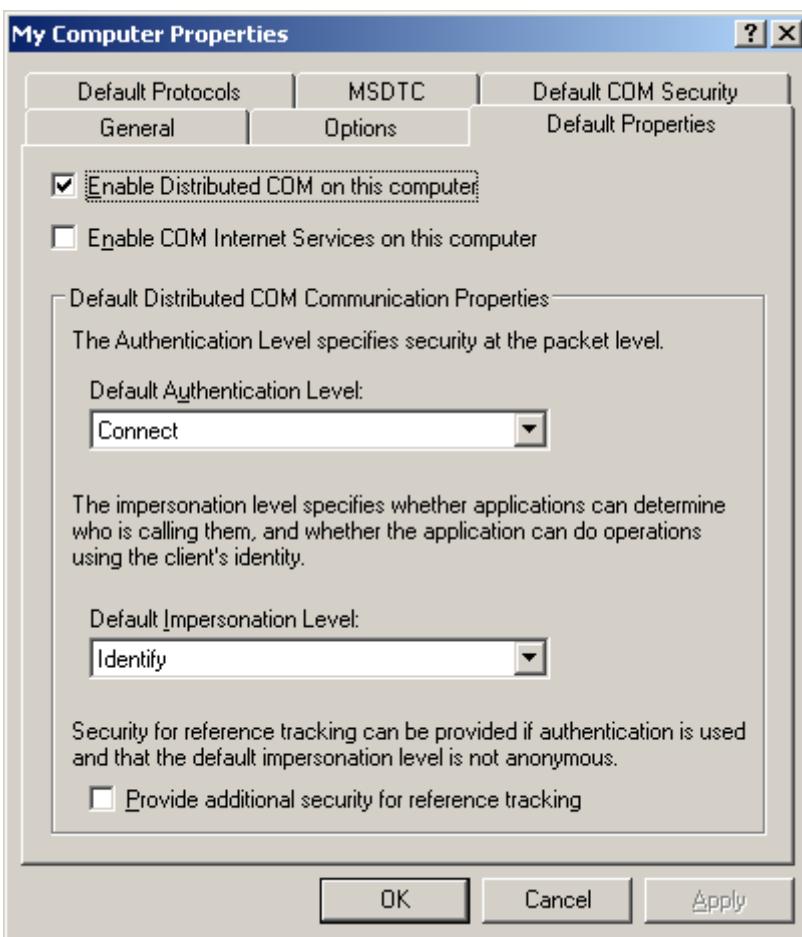


- On the "Default Protocols" tab the *Connection-oriented TCP/IP* protocol should be moved to the top position. This setting forces the use of TCP/IP for DCOM connections. All other protocols can be removed if they are not used with DCOM. The timeout will be reduced if DCOM tries to connect only on TCP/IP connections.



- On the Default Properties Tab the *Enable Distributed COM on this computer* must be checked. The Authentication Level and the Impersonation Level are set to *Connect* and *Identify* by default. If the client machine runs in a **Workgroup** the level should be changed
  - Authentication Level = None
  - Impersonation Level = Anonymous
- If the client machine runs in a **Domain** the level remains to default settings
  - Authentication Level = Connect
  - Impersonation Level = Identify
- If the client machine runs in a **mixed configuration** (e.g. the Client machine in a Workgroup and the Server machine in a Domain) the level should be changed. The machine being part of the Domain must be able to identify the security context without "asking" the Domain. Therefore the machine must "know" the users (they must have a local Login).
  - Authentication Level = None

- Impersonation Level = Anonymous



Not all possible combinations of setting these two levels make sense.

**Known Bugs:** on Windows 2000 operating systems the Network Configuration Icon disappears when setting DCOM security levels to *None* and *Anonymous*. The network still works but the IP-Address of the NIC can not be changed anymore. Change temporarily to default settings to change IP Address or use *None* and *Delegate*.

- On the Default COM Security Tab the Access- and Launch permission for all COM-Objects can be changed. As the OPC Client is nothing else than a COM Client, the security settings should be changed to grant access to the Client application. Specially when the OPC Server sends callbacks (e.g. *OnDataChange*) to the OPC Client the server's process must have access permission on the Client.
  - The **Default Access Permission** should be granted for

Administrators

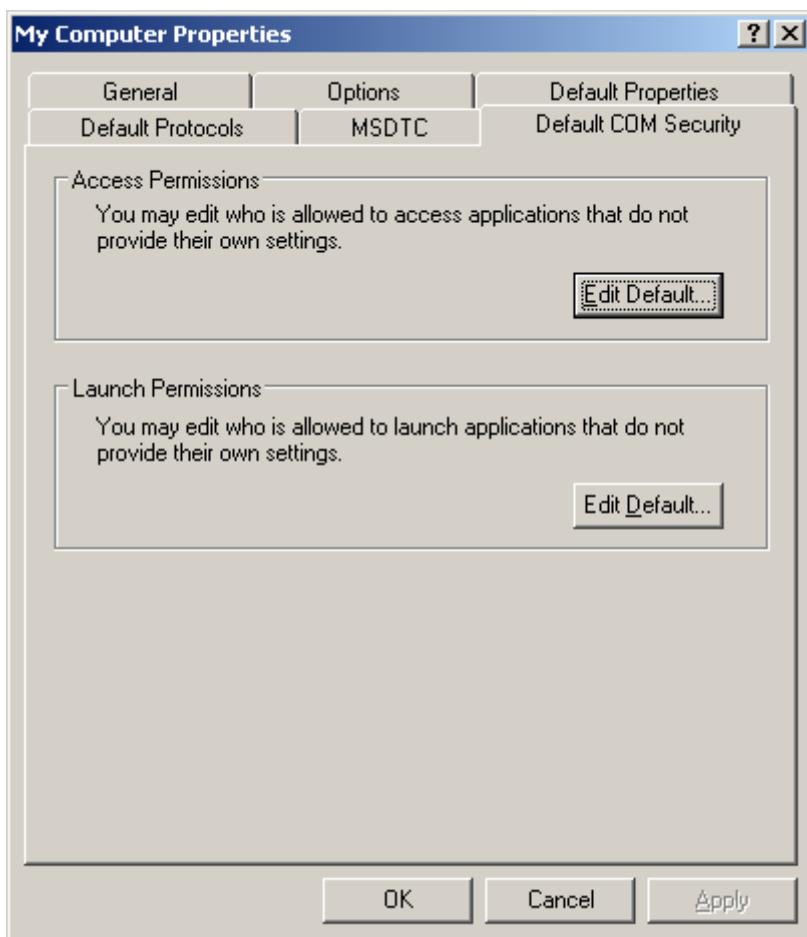
Interactive User

System

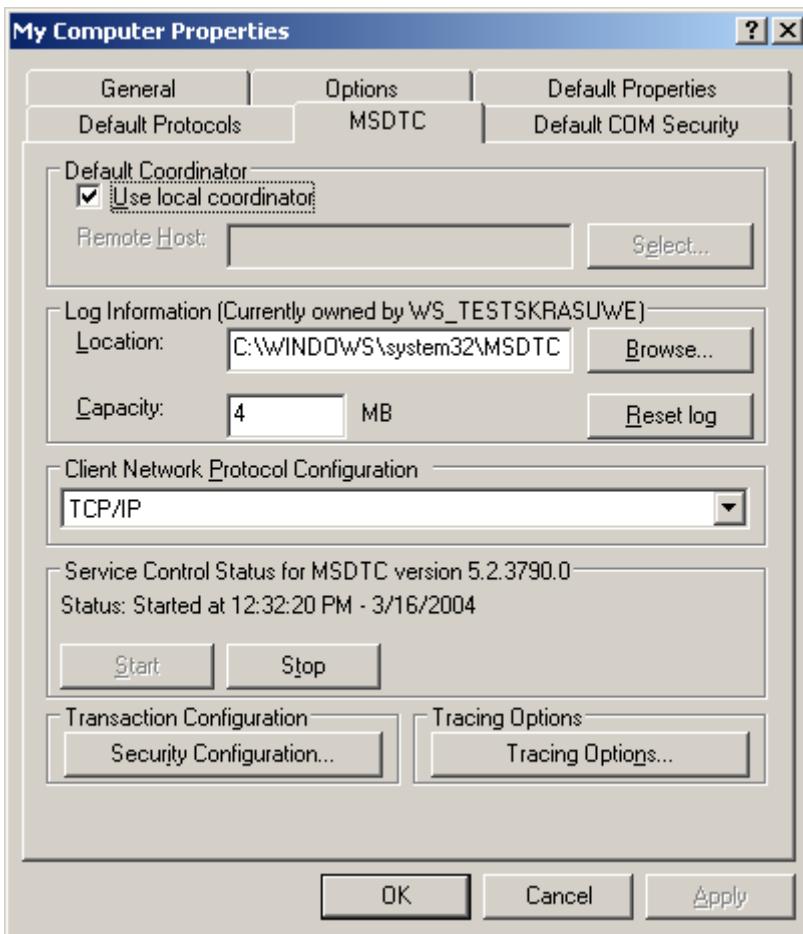
Network

"OPC Server's Security Context"

- The **Default Launch Permission** should not be changed on the Client machine.



- On the MSDTC Tab no changes must be done. The default settings will be correct for OPC Client side security settings.



## 8.2.3 Server

This section describes the required settings on the server's side. The server is the computer on which the TwinCAT OPC-Server application is running. Usually the OPC-Client is also located on this computer. However, in some environments it may be necessary that OPC-Client and OPC-Server need to be installed on different computers. Both, the OPC-Server's and OPC-Client's DCOM settings need to be configured, so that a remote communication between client and server is possible.



The following settings have been tested on Windows 2000, Windows XP and Windows 7 computers.

### Step 1: General network configuration

Depending on the server's operating system, some additional network settings need to be taken. Basically, the same settings must be performed. However, some "operating system specific" settings must be done to get the DCOM security running.

Please refer to our [Article about Operating Systems \[▶ 66\]](#) for more information.

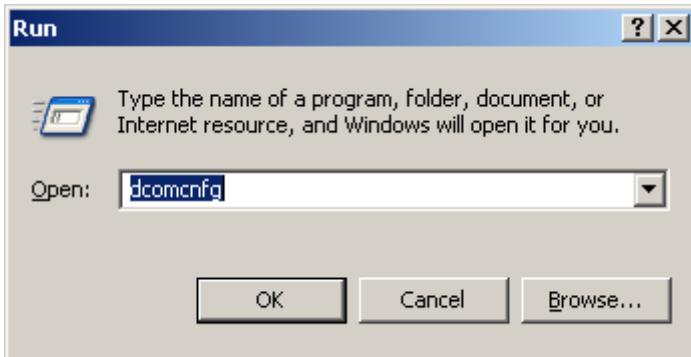
### Step 2: DCOM configuration

To configure DCOM for a remote OPC communication, please perform the following steps on the computer running TwinCAT OPC-Server.

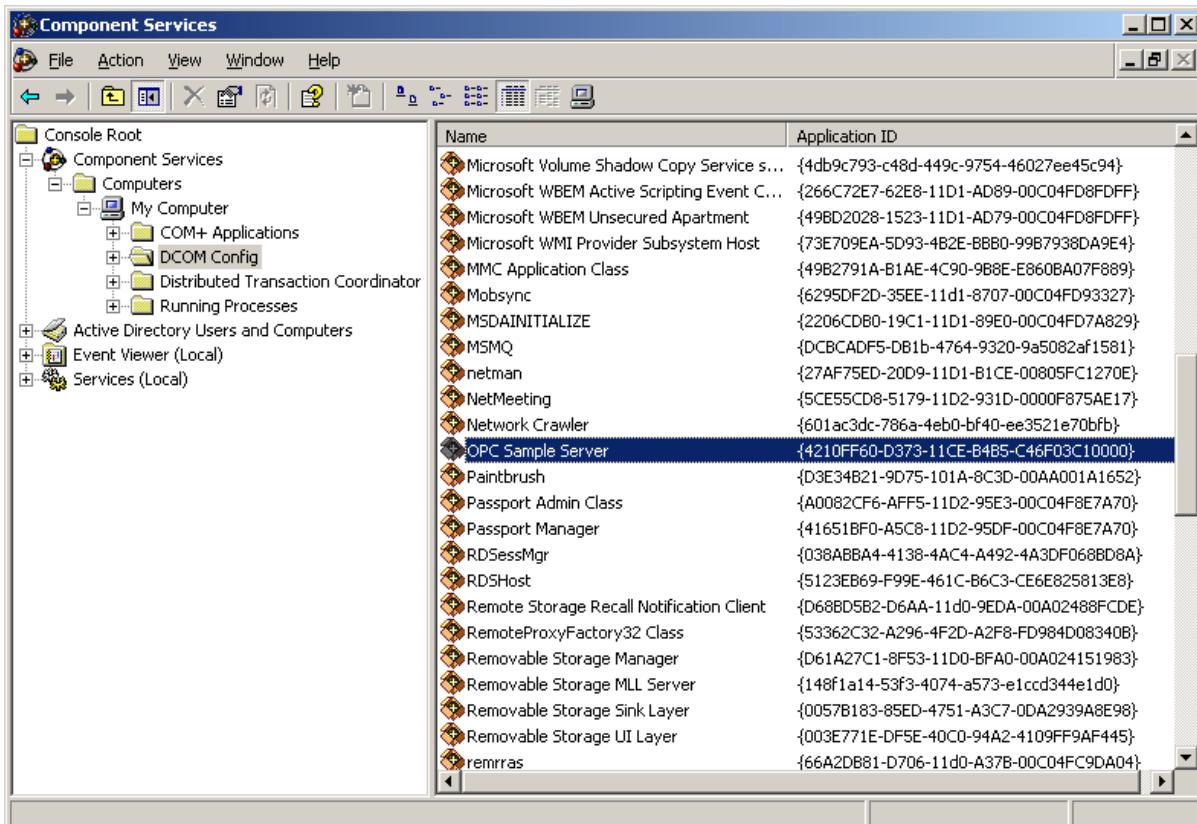


Only local administrators are allowed to open and change the DCOM security.

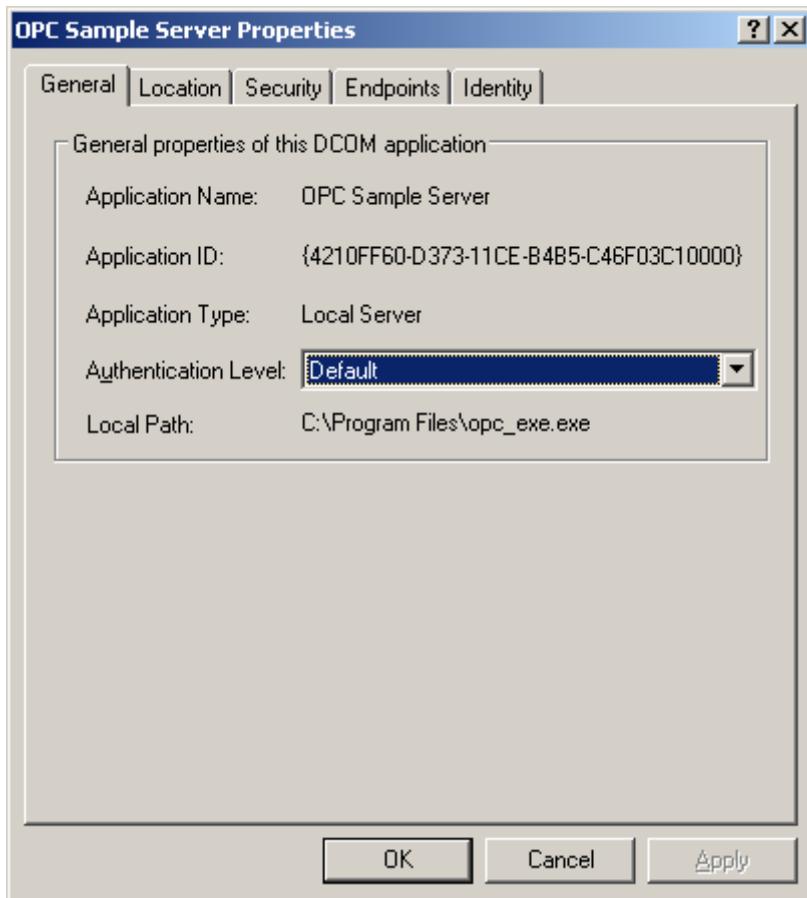
- Open *Start --> Run --> dcomcnfg.exe* to start the DCOM configuration dialog.



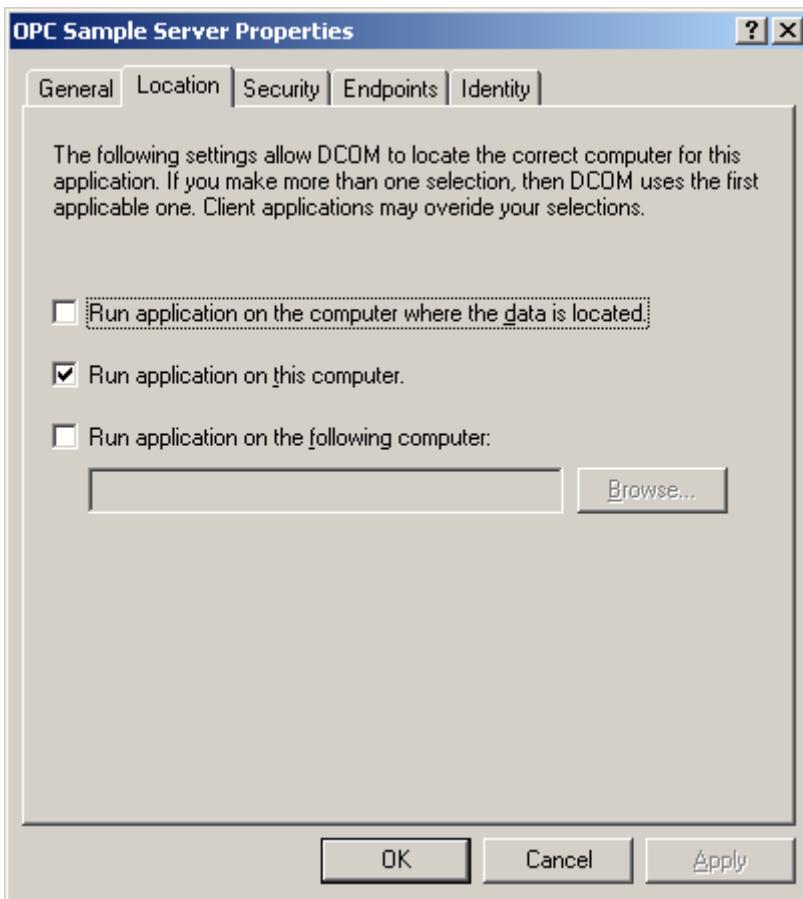
- Navigate to *Console Root --> Component Services --> Computers --> My Computer --> DCOM Config* to display all DCOM server applications.
- Select the TwinCAT OPC-Server (or one of its Clones), right click it and select **Properties** to change the DCOM security for this specific DCOM Server only.



- On the "General" tab no changes must be made. The default settings will be correct for OPC Server's security settings. The Authentication Level set to *Default* will overtake the settings from the Default Properties Tab [► 69] valid for all COM Objects on this machine (*Connect by default*).

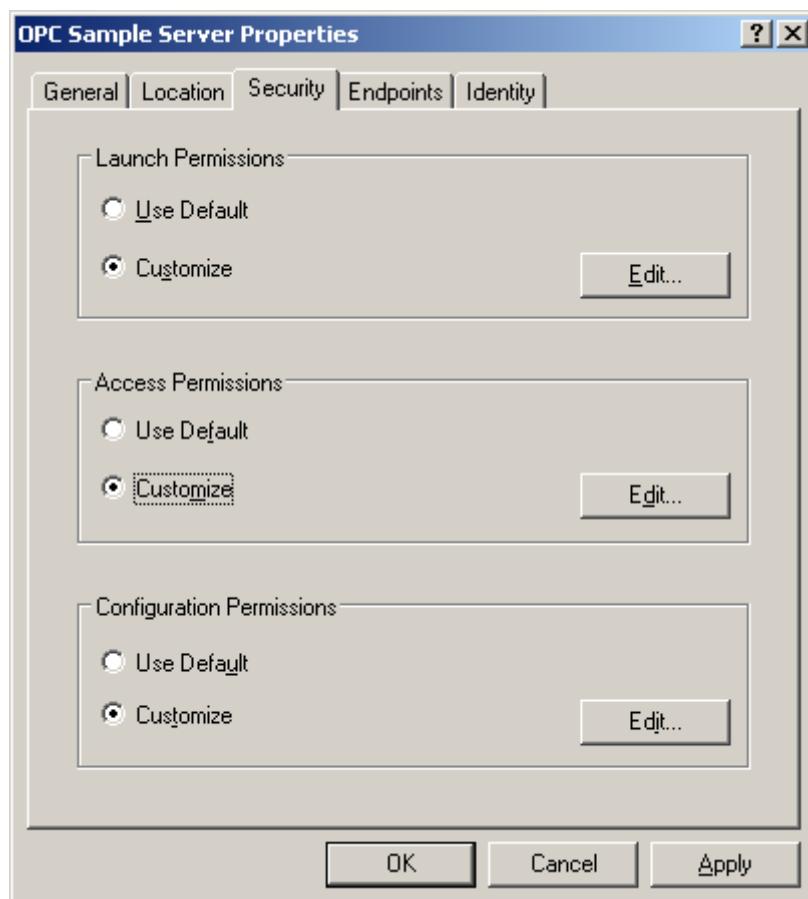


- On the "Location" tab no changes must be made. The default settings will be correct for the OPC Server's security settings.

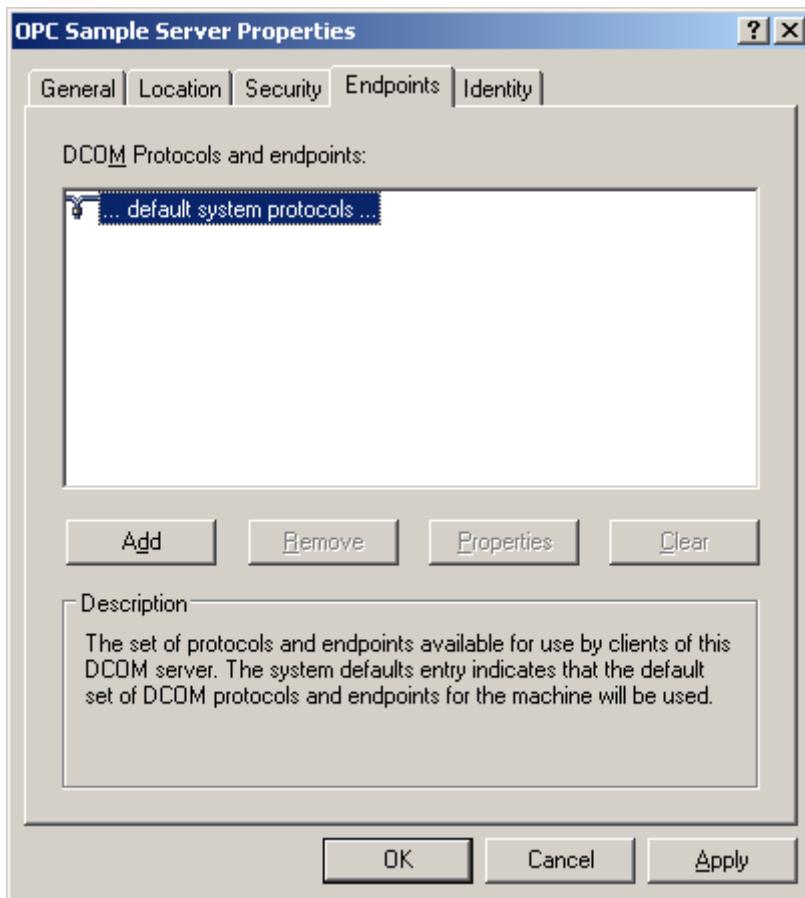


- On the "Security" tab the Access- and Launch permission for this specific COM-Server can be changed. As the OPC Server is nothing else than a COM Server, the security settings should be changed to grant access and launch permission to the Server application.
- The **Server Specific Access Permission** should be granted for
  - Administrators
  - Interactive User
  - System
  - Network
  - "OPC Client's Security Context"
- The **Server Specific Launch Permission** should be granted for
  - Administrators
  - Interactive User
  - System
  - Network
  - "OPC Client's Security Context"

- The **Server Specific Configuration Permission** should not be changed on the Server machine.



- On the "Endpoints" tab the default settings should remain. In the [Default Protocols Tab \[▶ 69\]](#) for all COM-Objects *Connection-oriented TCP/IP* protocol should be moved to the top position.



- On the "Identity" tab no changes must be made. By default, the *Interactive User* will be selected. This means that the server will be launched with the security context of the interactive user (the user that is logged on). As this security context changes if somebody else is logged on, in most cases a specific person should be selected i.e even if nobody is logged on to the machine (after reboot) the server can be launched always having the same security context.

The preferred setting should be the third selection.

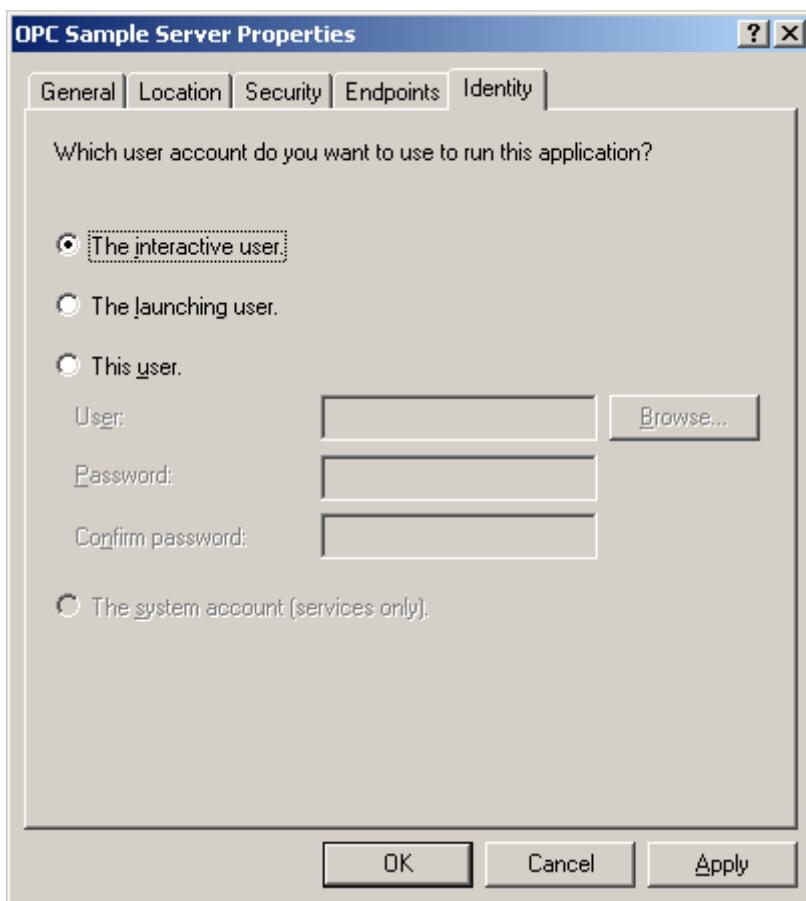
- The interactive user = default

The interactive user depends on the person that is logged on, thus it can be different each time and only exists if somebody is logged on.

- The launching user = should NEVER be used

The launching user will have the security context of the Client application (the OPC Client launches/connects the OPC Server). When having different Clients in the network, several instances of the Server will be launched having different security context each.

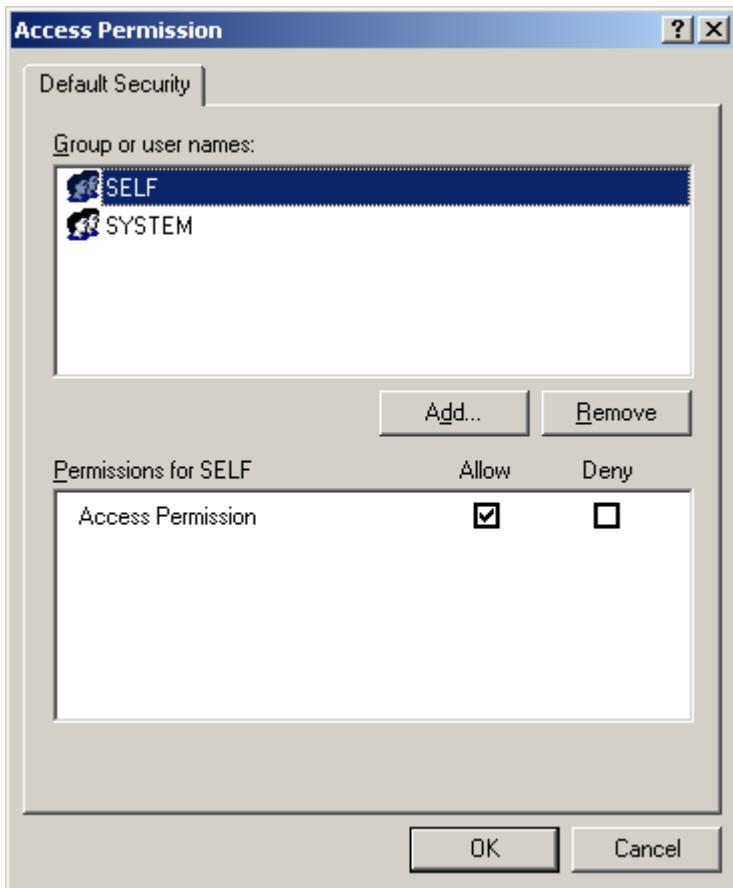
- **This user** = Server will overtake the security context of this person  
By selecting *This user* it will be guaranteed that always the same person's security context is used when the server is started. On the Client side only for this person the *Access Permission* must be granted.



## 8.2.4 DCOM Permission

### 8.2.4.1 Access Permission

For granting access permission (accessing the application) to a certain user the *Edit...* button must be clicked opening the following dialog.

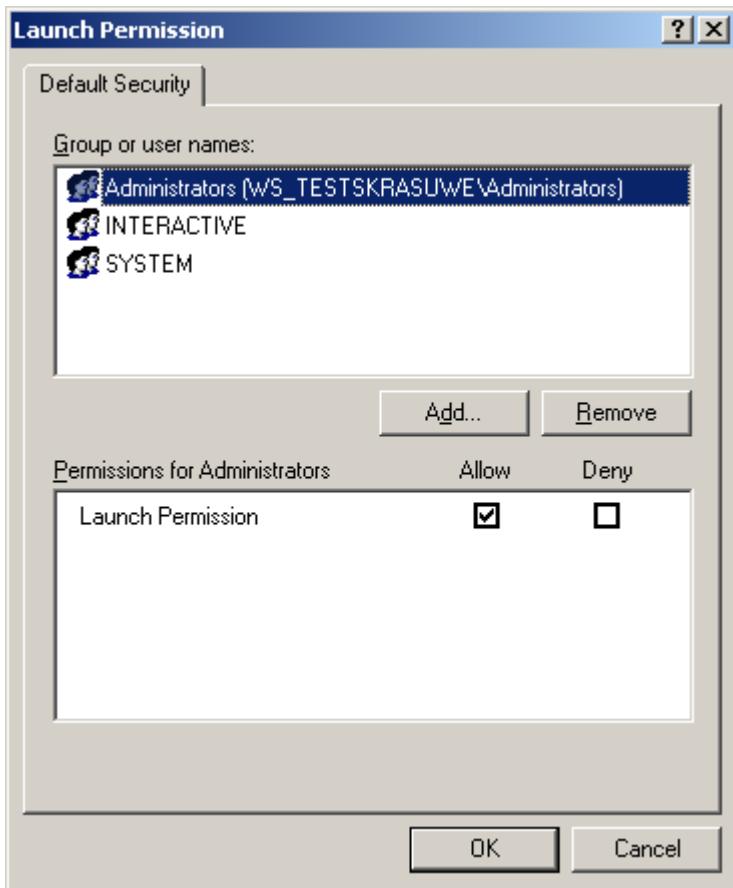


For selecting users click the *Add..* button:

Selecting Users [▶ 84]

#### 8.2.4.2            Launch Permission

For granting *launch* permission (starting the application) to a certain user the *Edit...* button must be clicked opening the following dialog.

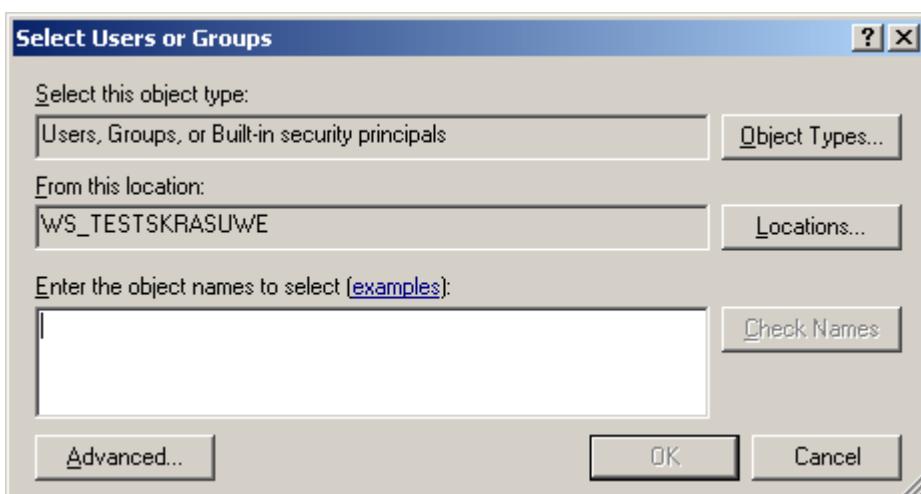


For selecting users click the *Add..* button:

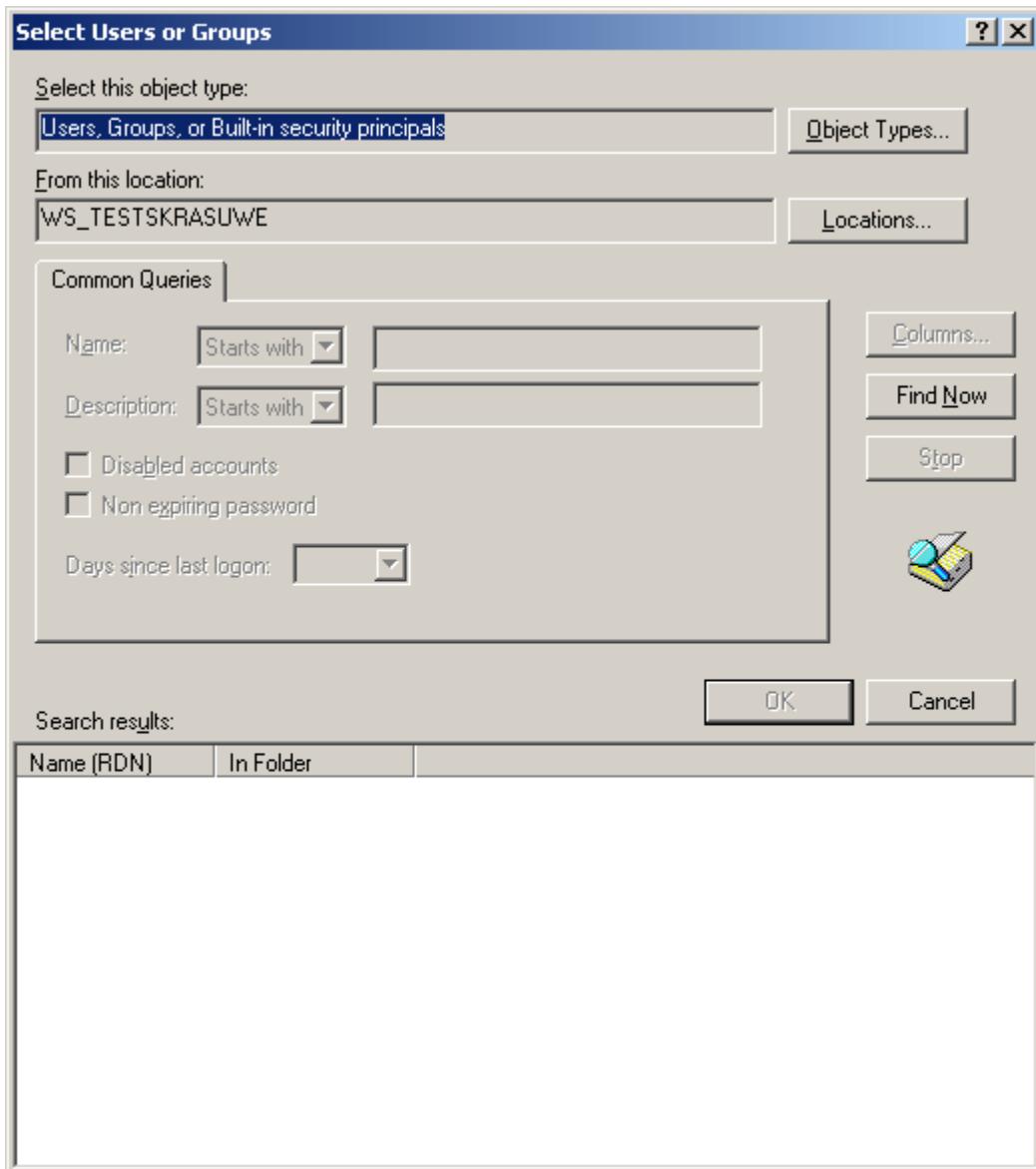
[Selecting Users \[► 84\]](#)

### 8.2.4.3 Select User

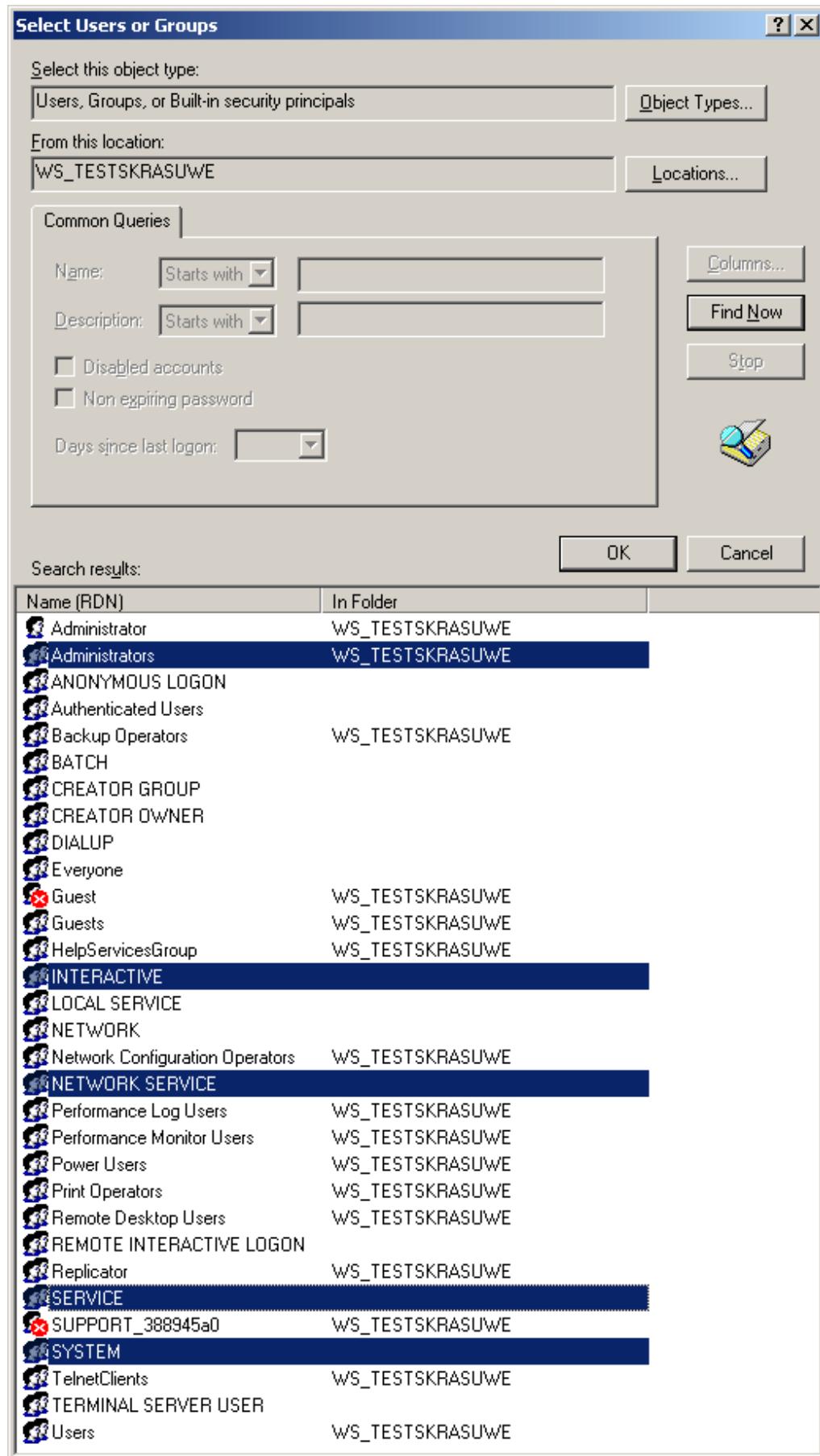
A filter on object types can be set to reduce the listing of all user names. The location states the machine name where the user account is stored, this can be the local workstation or the central domain controller machine. The edit box below the names can be typed in and verified afterwards by pressing the *Check Names* button.



All users known by the machine selected in *Location* can be displayed by pressing the *Advanced...* button.



The list of all known users will be displayed after clicking the *Find Now* button.





More Information:  
**[www.beckhoff.com](http://www.beckhoff.com)**

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20

33415 Verl

Germany

Phone: +49 5246 9630

[info@beckhoff.com](mailto:info@beckhoff.com)

[www.beckhoff.com](http://www.beckhoff.com)

