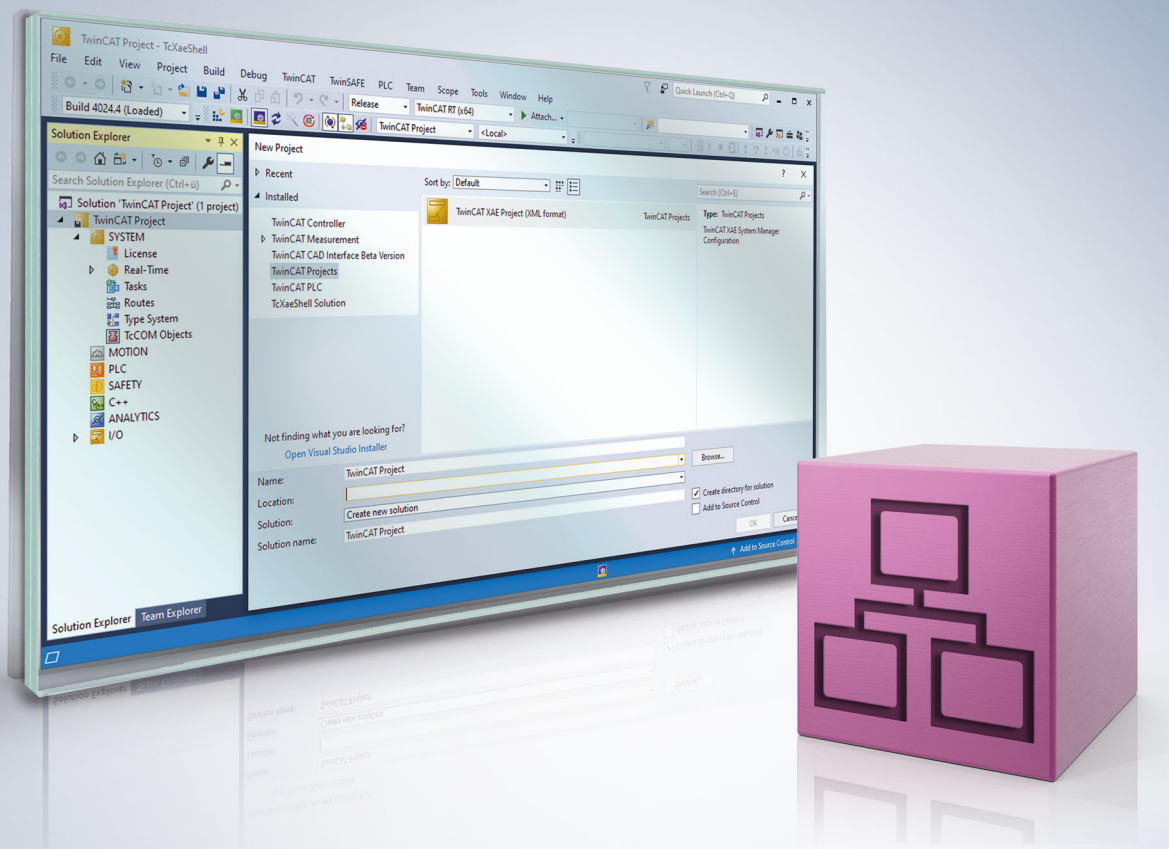
BECKHOFF New Automation Technology

Manual | EN

TF6120

TwinCAT 3 | OPC DA



2024-04-05 | Version: 1.2.1

Table of contents

1 Foreword 5
1.1 Notes on the documentation 5
1.2 For your safety 5
1.3 Notes on information security 7
2 Product description 8
3 Installation 9
3.1 System requirements 9
3.2 Installation 9
3.3 Licensing 12
3.4 Setup OPC XML-DA on Windows XP 14
3.5 Setup OPC XML-DA on Windows 7 18
4 Configuration 24
4.1 OPC DA Server 24
4.1.1 OPC DA Server 24
4.1.2 General 24
4.1.3 Data Access 30
4.1.4 Conversion 38
4.1.5 Simulation 41
4.1.6 Item Properties 43
4.1.7 Data exchange via network 44
4.2 OPC XML-DA Server 47
4.2.1 OPC XML DA 47
4.2.2 Status information 48
5 Appendix 49
5.1 OPC Compliance Certificate 49
5.2 DCOM 50
5.2.1 Overview 50
5.2.2 Prerequisites 51
5.2.3 Client 54
5.2.4 Server 61
5.2.5 DCOM Permissions 66

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all 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 notice.
No claims to modify 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 and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT®

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

Copyright

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 Product description

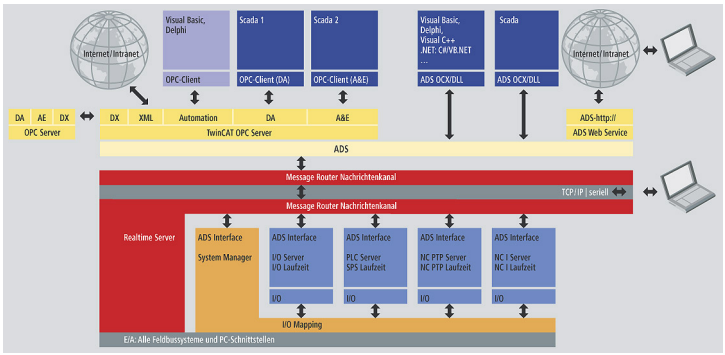
The described product is replaced by the product TF6100 TC3 OPC UA.

OPC stands for "Openess Productivity and Collaboration" and describes an initiative for standardisation of data exchange in automation. Beckhoff is an active member of the OPC Foundation. (More information about OPC may be obtained under www.opcfoundation.org) The OPC standard specifies the communication of industrial process data between sensors, instruments, controllers and software systems. An OPC-Server is a software component, which offers an OPC interface to other applications. The TwinCAT 3 Function TF6120 OPC-DA contains such an OPC-Server and provides the following features:

- 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
- OPC XML DA Server

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.



What is included

The installation version includes

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

3 Installation

3.1 System requirements

The following chapter lists the system requirements.

The TS6120 supplement or the TF6120 function is no longer being further developed and is therefore not released for newer operating systems. The so-called UA Gateway in the product TS6100/TF6100 OPC UA is available to you as a free OPC DA interface:

TwinCAT OPC UA Gateway

OPC DA Server

- Operating systems:
- Windows XP Pro SP3
- Windows Server 2008 R2
- Windows Server 2012
- TwinCAT:
- TwinCAT 3 XAE Build 3100 (or higher)
- TwinCAT 3 XAR Build 3100 (or higher)

OPC XML DA Server

- Operating systems:
- Windows XP Pro SP3
- TwinCAT:
- TwinCAT 3 XAE Build 3100 (or higher)
- TwinCAT 3 XAR Build 3100 (or higher)
- Other:
- Internet Information Services 5.1, 6.0, 7.0 and 7.5

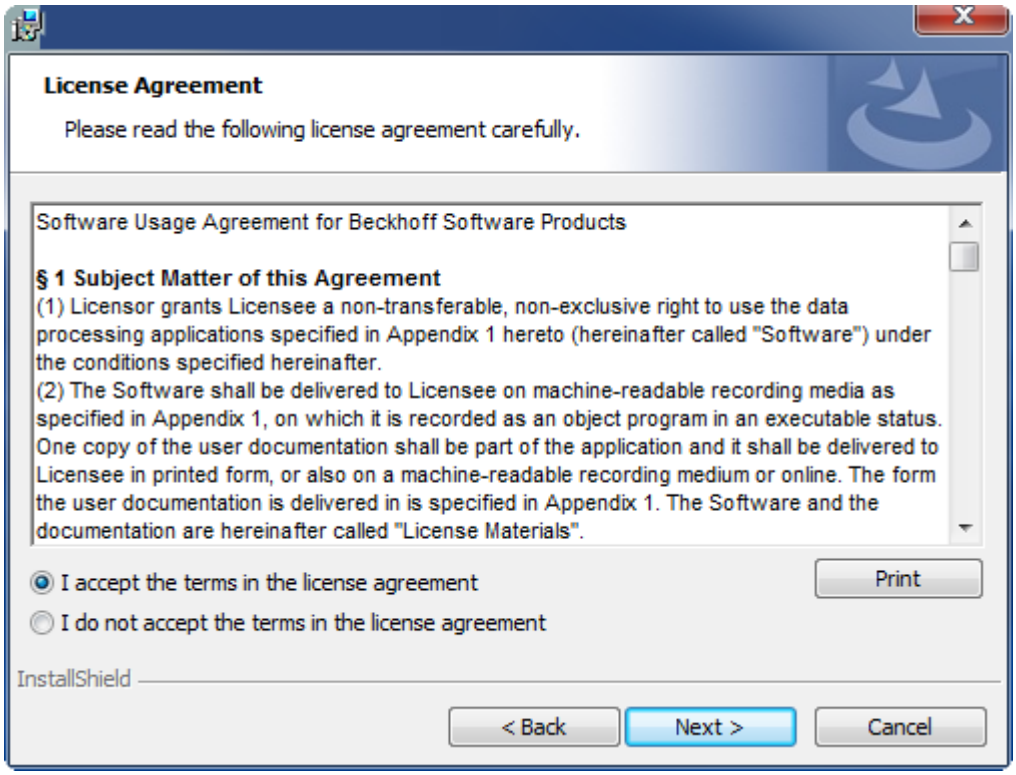
3.2 Installation

The following section describes how to install the TwinCAT 3 Function for Windows-based operating systems.

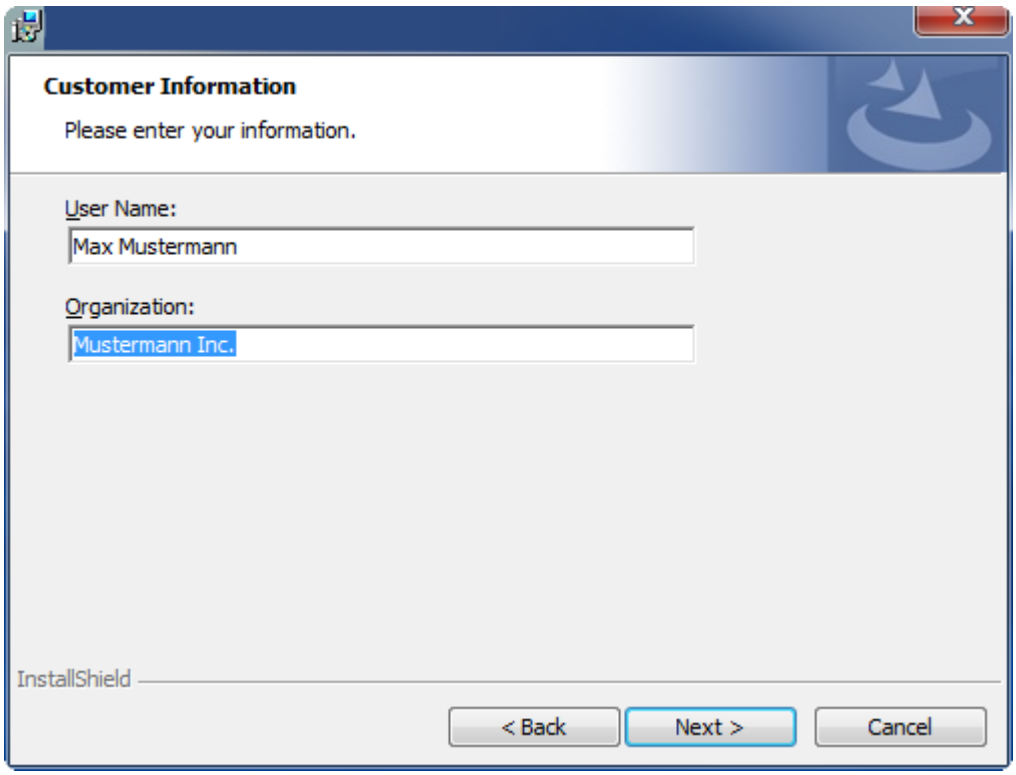
✓ The TwinCAT 3 Function setup file was downloaded from the Beckhoff website.

1. Run the setup file as administrator. To do this, select the command **Run as administrator** in the context menu of the file.
   ⇨ The installation dialog opens.

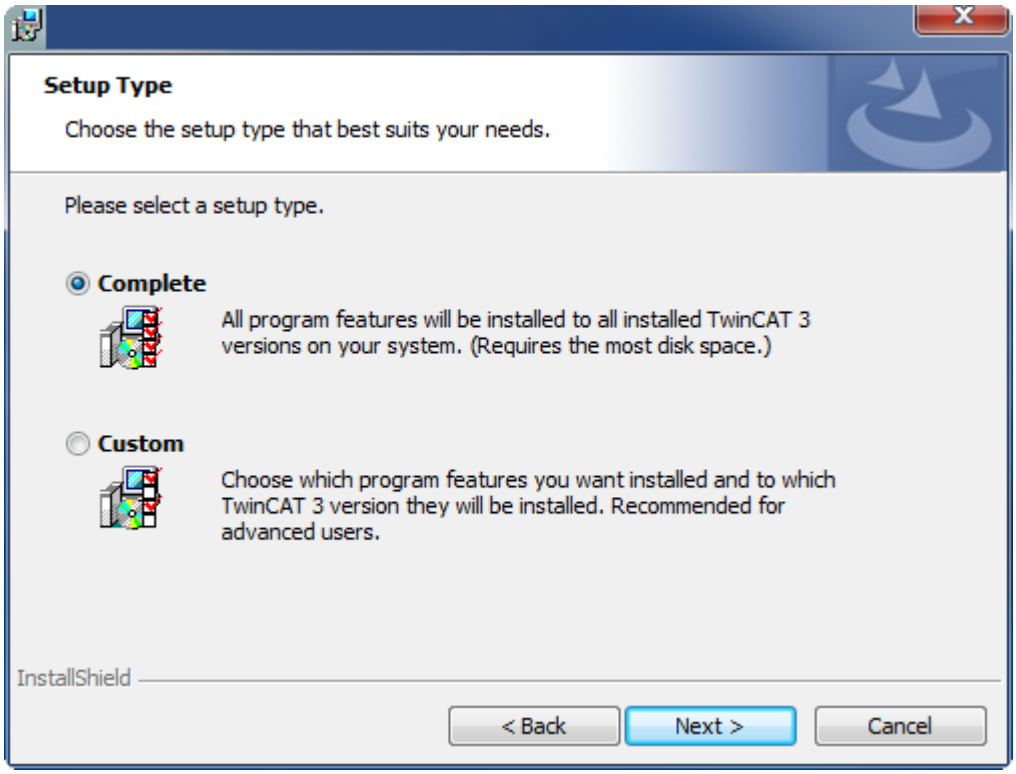
2. Accept the end user licensing agreement and click **Next**.



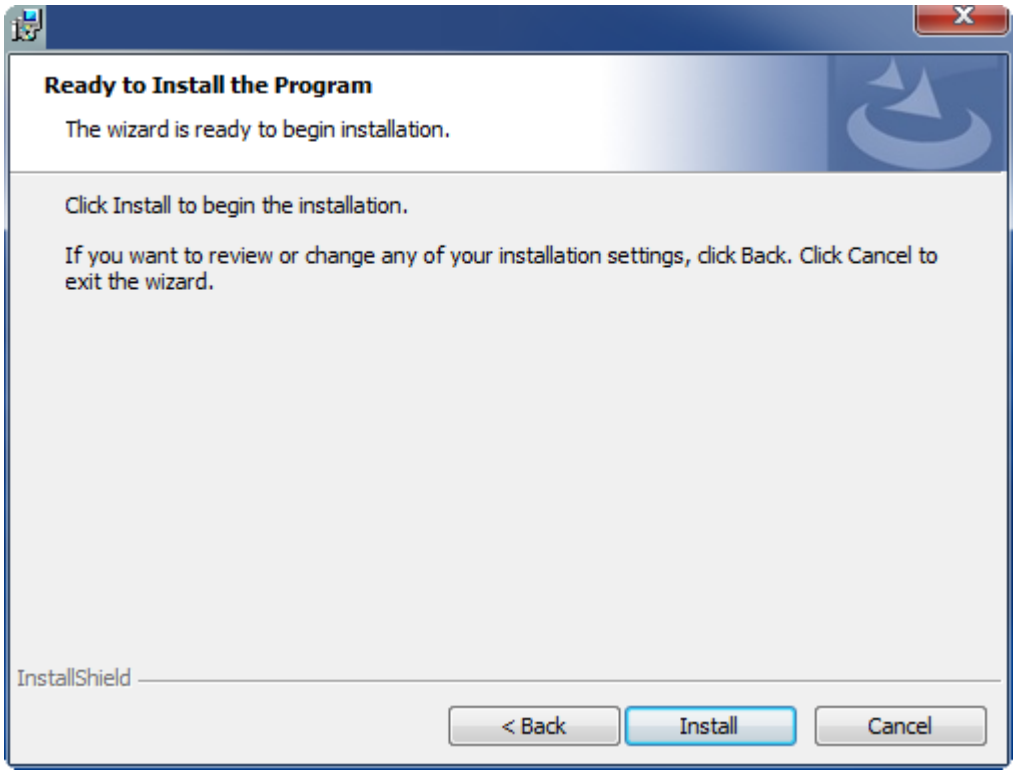
3. Enter your user data.



4. If you want to install the full version of the TwinCAT 3 Function, select **Complete** as installation type. If you want to install the TwinCAT 3 Function components separately, select **Custom**.

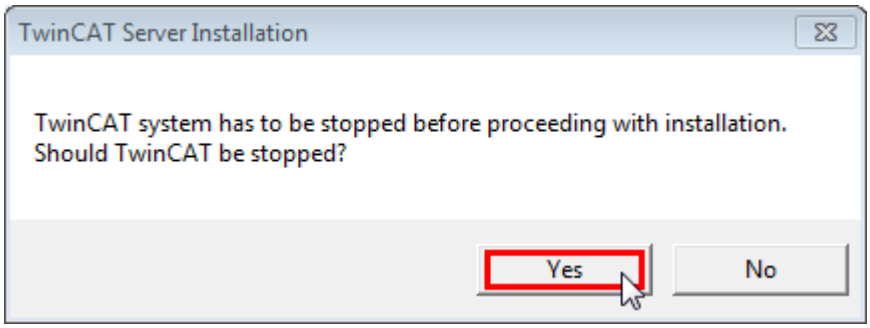


5. Select **Next**, then **Install** to start the installation.

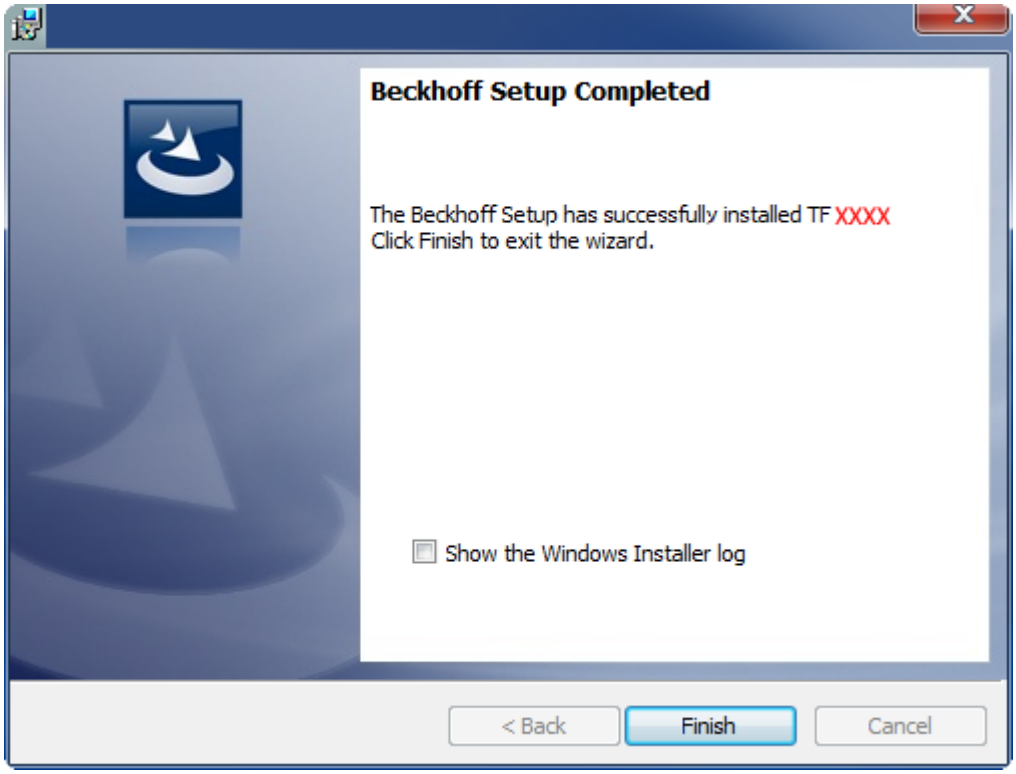


⇨ A dialog box informs you that the TwinCAT system must be stopped to proceed with the installation.

6. Confirm the dialog with **Yes**.



7. Select **Finish** to exit the setup.



⇨ The TwinCAT 3 Function has been successfully installed and can be licensed (see Licensing [▶ 12]).

3.3 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

Licensing the full version of a TwinCAT 3 Function

A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "TwinCAT 3 Licensing".

Licensing the 7-day test version of a TwinCAT 3 Function

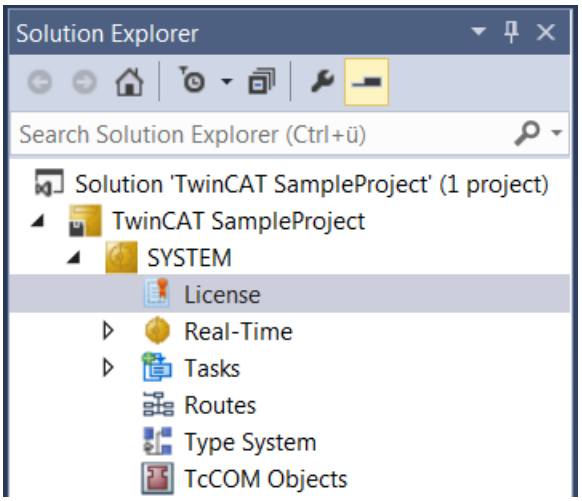
A 7-day test version cannot be enabled for a TwinCAT 3 license dongle.

1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.

3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.

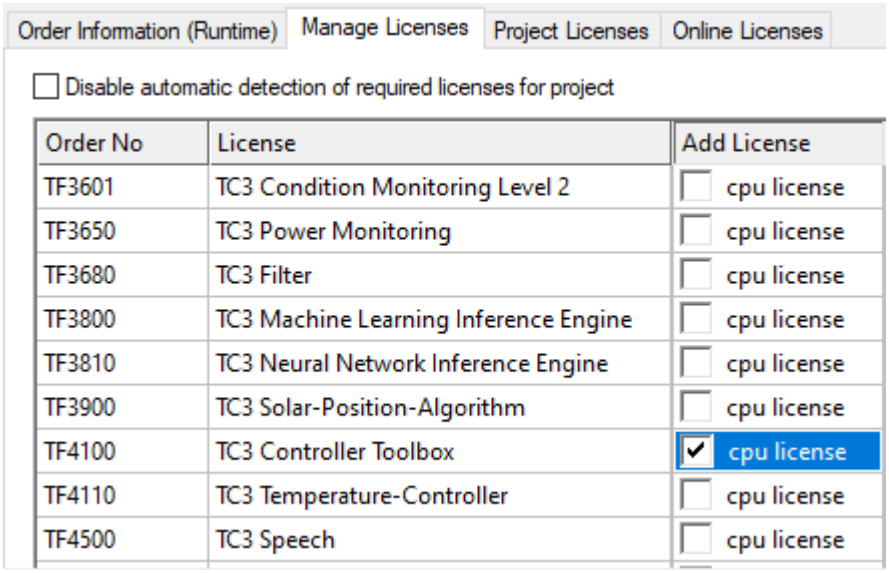
   ⇨ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.

4. In the **Solution Explorer**, double-click **License** in the **SYSTEM** subtree.



   ⇨ The TwinCAT 3 license manager opens.

5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF4100 TC3 Controller Toolbox").

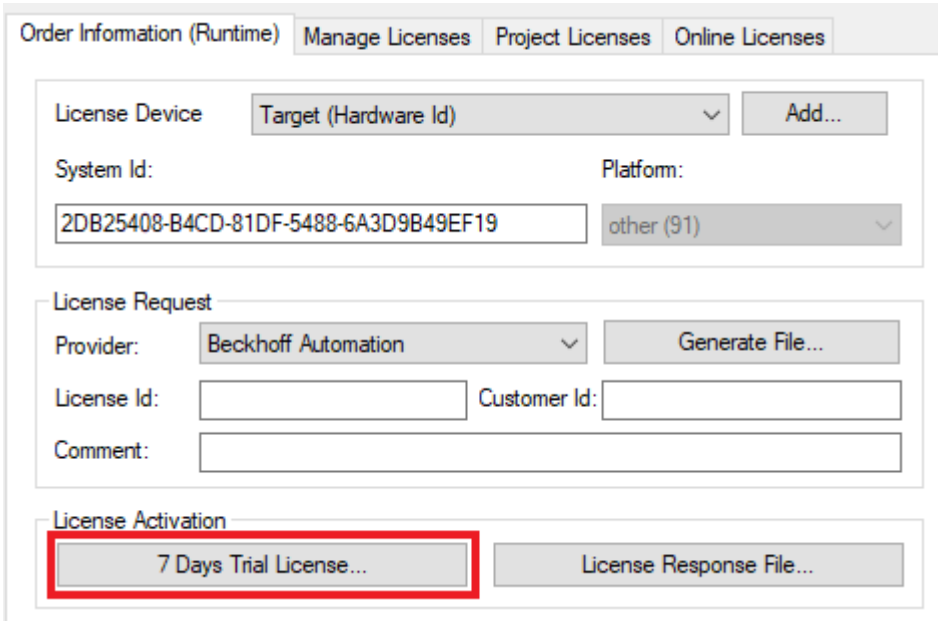


Order No	License	Add License
TF3601	TC3 Condition Monitoring Level 2	cpu license
TF3650	TC3 Power Monitoring	cpu license
TF3680	TC3 Filter	cpu license
TF3800	TC3 Machine Learning Inference Engine	cpu license
TF3810	TC3 Neural Network Inference Engine	cpu license
TF3900	TC3 Solar-Position-Algorithm	cpu license
TF4100	TC3 Controller Toolbox	cpu license
TF4110	TC3 Temperature-Controller	cpu license
TF4500	TC3 Speech	cpu license

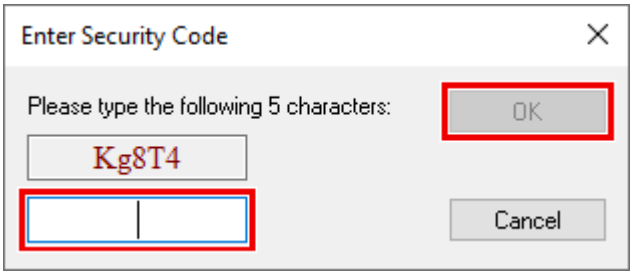
6. Open the **Order Information (Runtime)** tab.

   ⇨ In the tabular overview of licenses, the previously selected license is displayed with the status “missing”**.**

7. Click **7-Day Trial License...** to activate the 7-day trial license.



⇨ A dialog box opens, prompting you to enter the security code displayed in the dialog.



8. Enter the code exactly as it is displayed and confirm the entry.
9. Confirm the subsequent dialog, which indicates the successful activation.
   ⇨ In the tabular overview of licenses, the license status now indicates the expiry date of the license.
10. Restart the TwinCAT system.

⇨ The 7-day trial version is enabled.

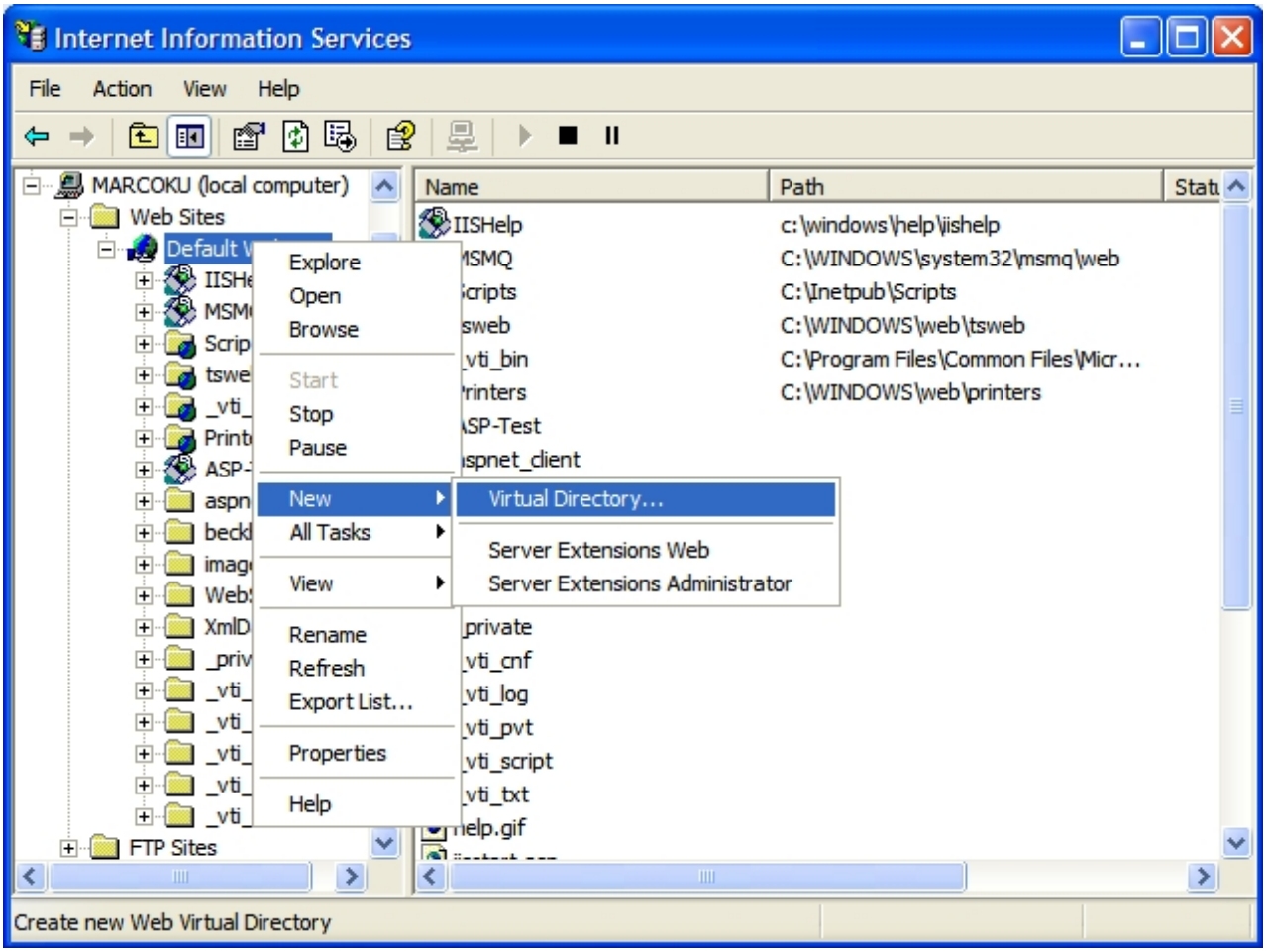
3.4 Setup OPC XML-DA on Windows XP

All necessary files for OPC XML-DA will be automatically installed by the setup routine. This chapter describes the required IIS (Internet Information Services) configuration for OPC XML DA on Windows XP

Please note: The configuration may be different in other Windows Operating Systems, for example Windows 7 [▶ 18].

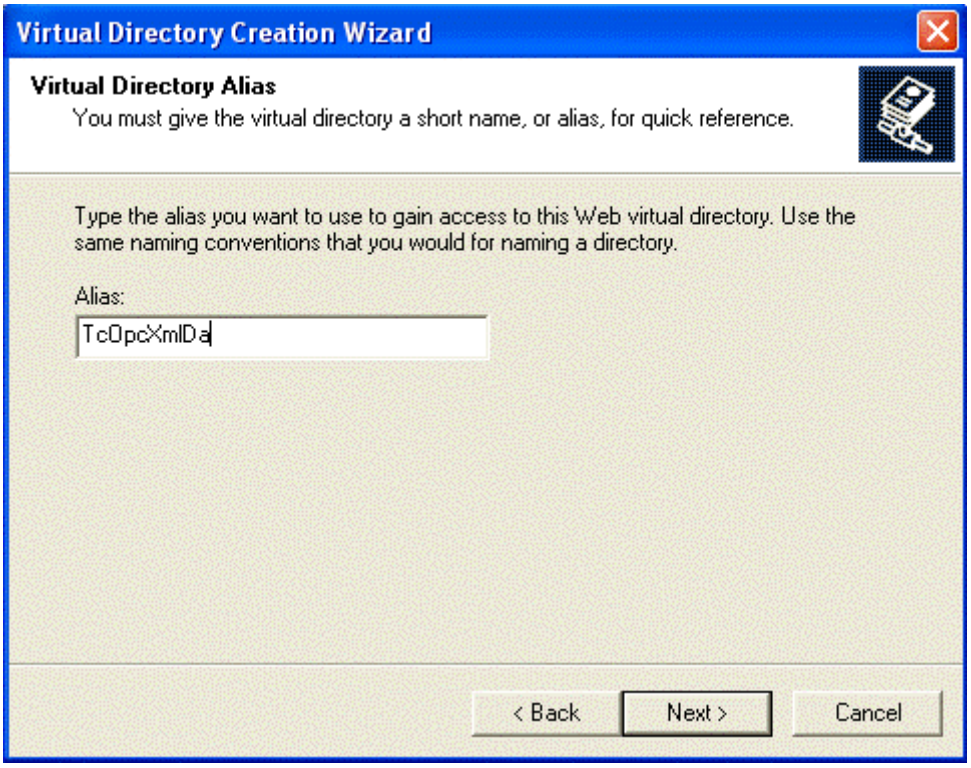
Step 1: Create "Virtual Directory" in IIS (Internet Information Service)

- Open "Internet Information Services" (under "Control Panel/Administrative Tools/").
- Right click on "Default Web Site"
- Select "New" and "Virtual Directory..."

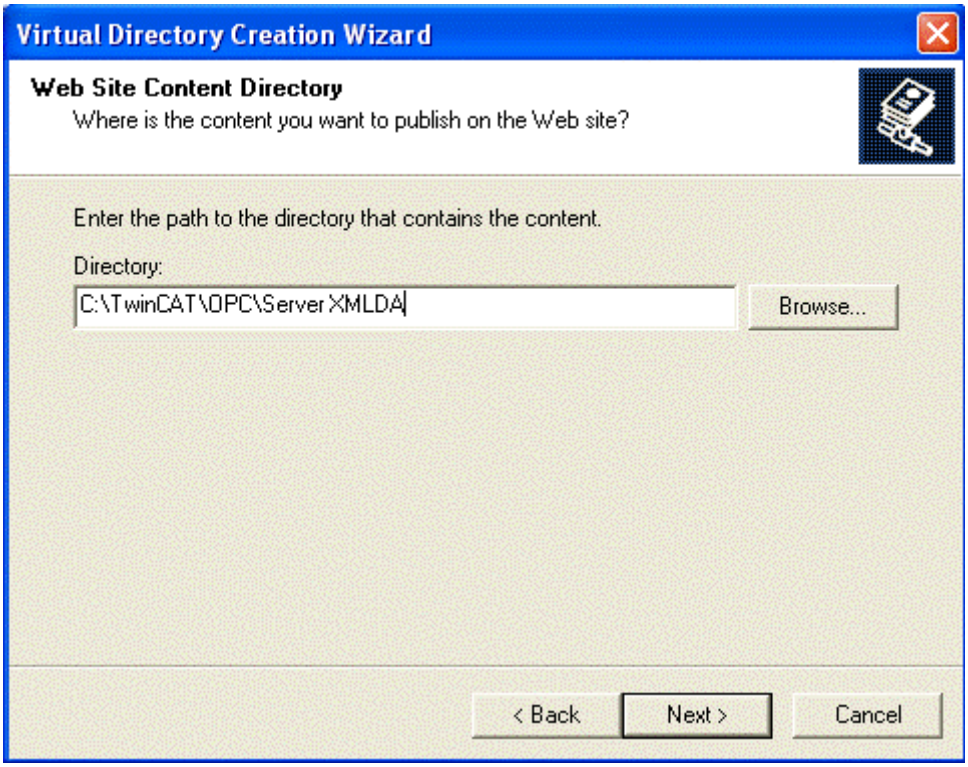


- Everybody is welcome, so just click next.

- 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.

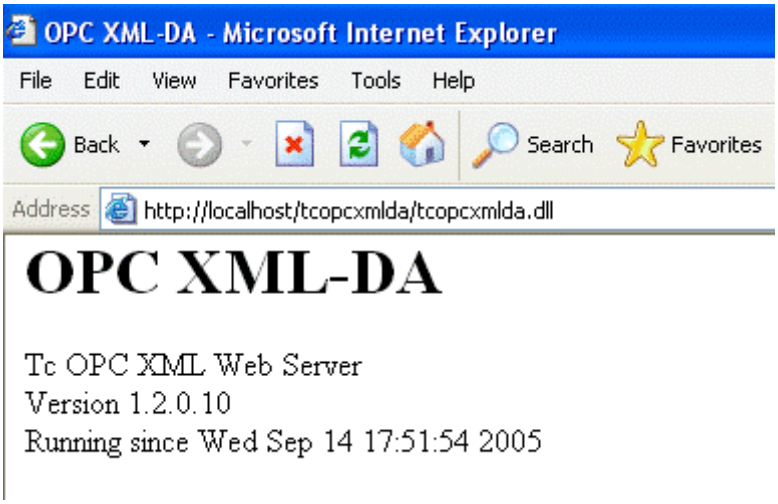
Step 2 : Testing the configuration

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>/tcopcxmlda/tcopcxmlda.dll

Sample :
http://192.16.17.5/tcopcxmlda/tcopcxmlda.dll

or
http://localhost/tcopcxmlda/tcopcxmlda.dll

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



Please note:
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.

3.5 Setup OPC XML-DA on Windows 7

All necessary files for OPC XML-DA will be automatically installed by the setup routine. This chapter describes the required IIS (Internet Information Services) configuration for OPC XML DA on Windows 7.

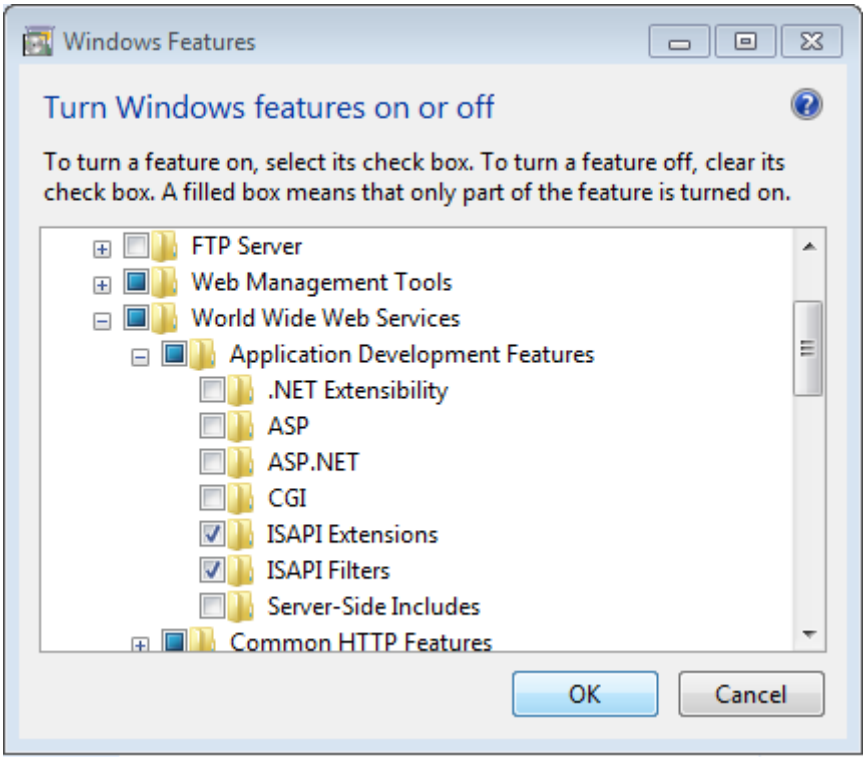
Please note: The configuration may be different in other Windows Operating Systems, for example Windows XP [▶ 14].

Step 1: Installing IIS on Windows 7

By default, IIS 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.

Please note: When installing IIS, the following extensions need to be activated:

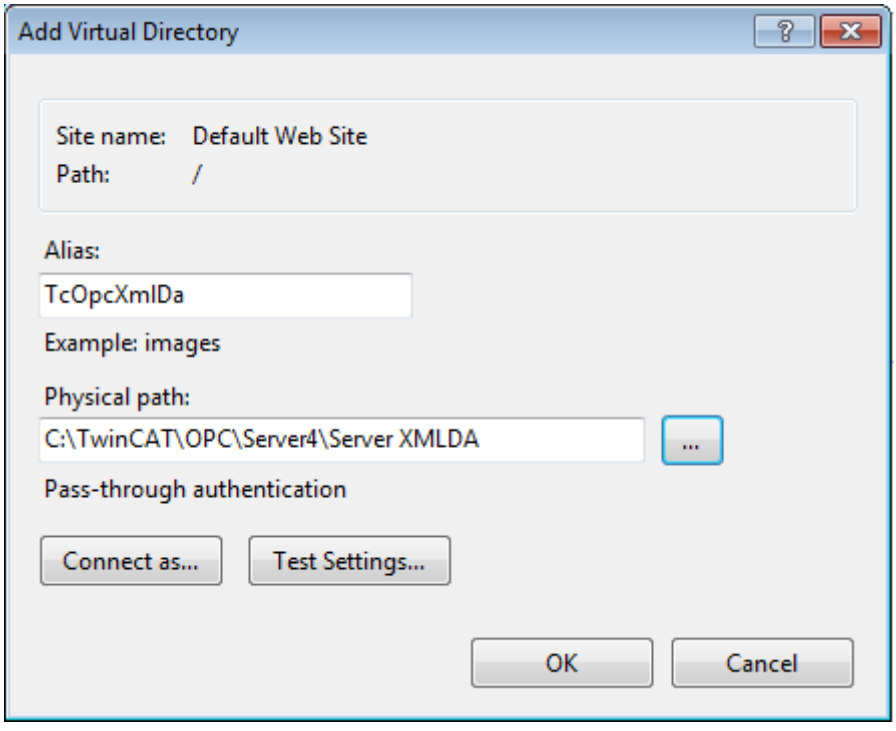
- ISAPI Extensions
- ISAPI Filters



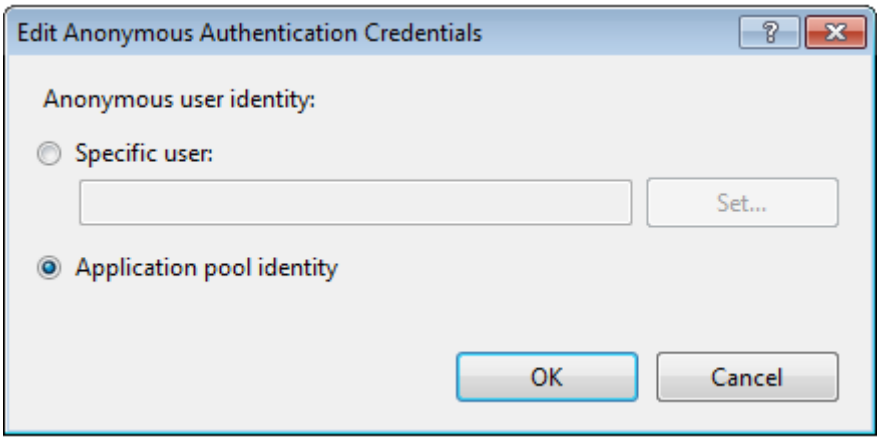
Step 2: Create "Virtual Directory" in IIS (Internet Information Service)

Usually this step will be performed automatically by the Setup.

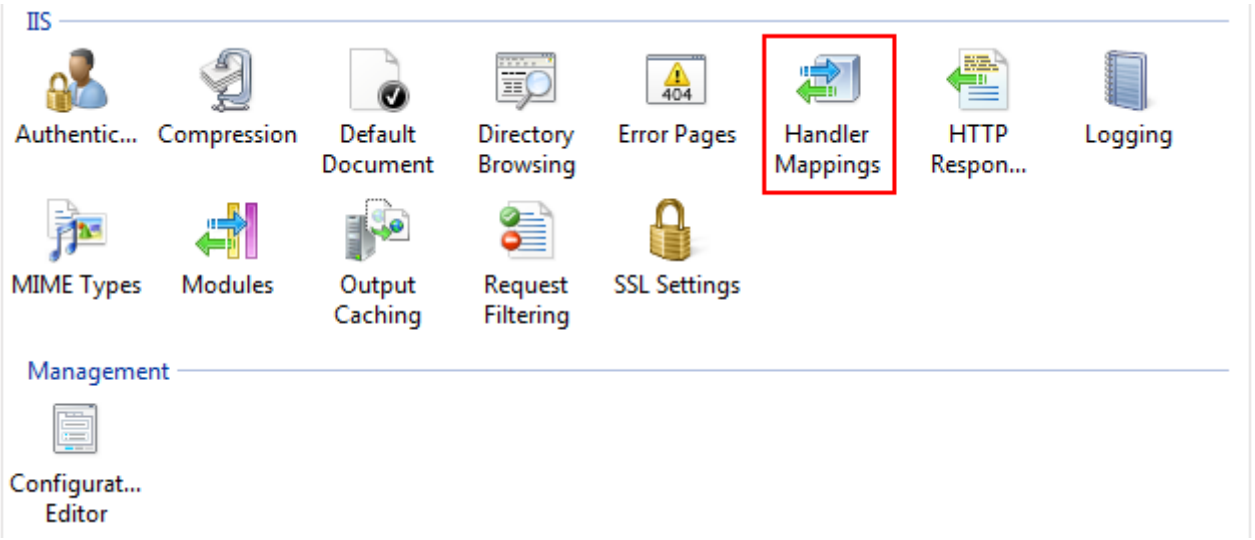
- Open "Internet Information Service (IIS) Manager" which can be found under "Control Panel\Administrative Tools\"
- Right click on "Default Web Site"
- Select "Add Virtual Directory..."
- Please enter the alias "TcOpcXmlDa" and the physical path to your TwinCAT OPC XML DA Server installation. By default,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



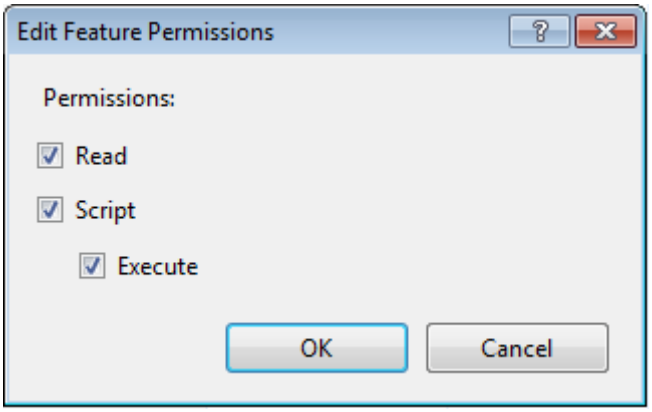
- Double-click the "Authentication" icon, select "Anonymous Authentication" and click on "Edit". Instead of specifying a user account, select the "Application pool identity" and click on "Ok".



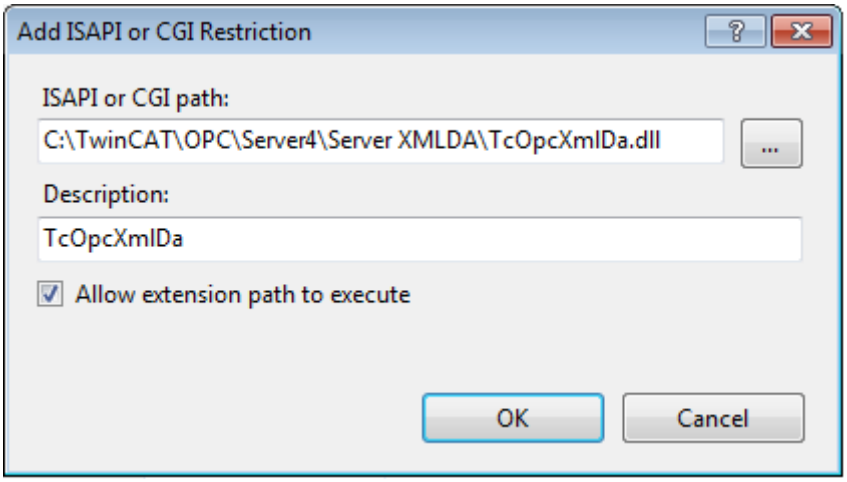
- Next, you need to set execute permissions on that virtual directory. Select the added directory and double-click on "Handler Mappings"



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



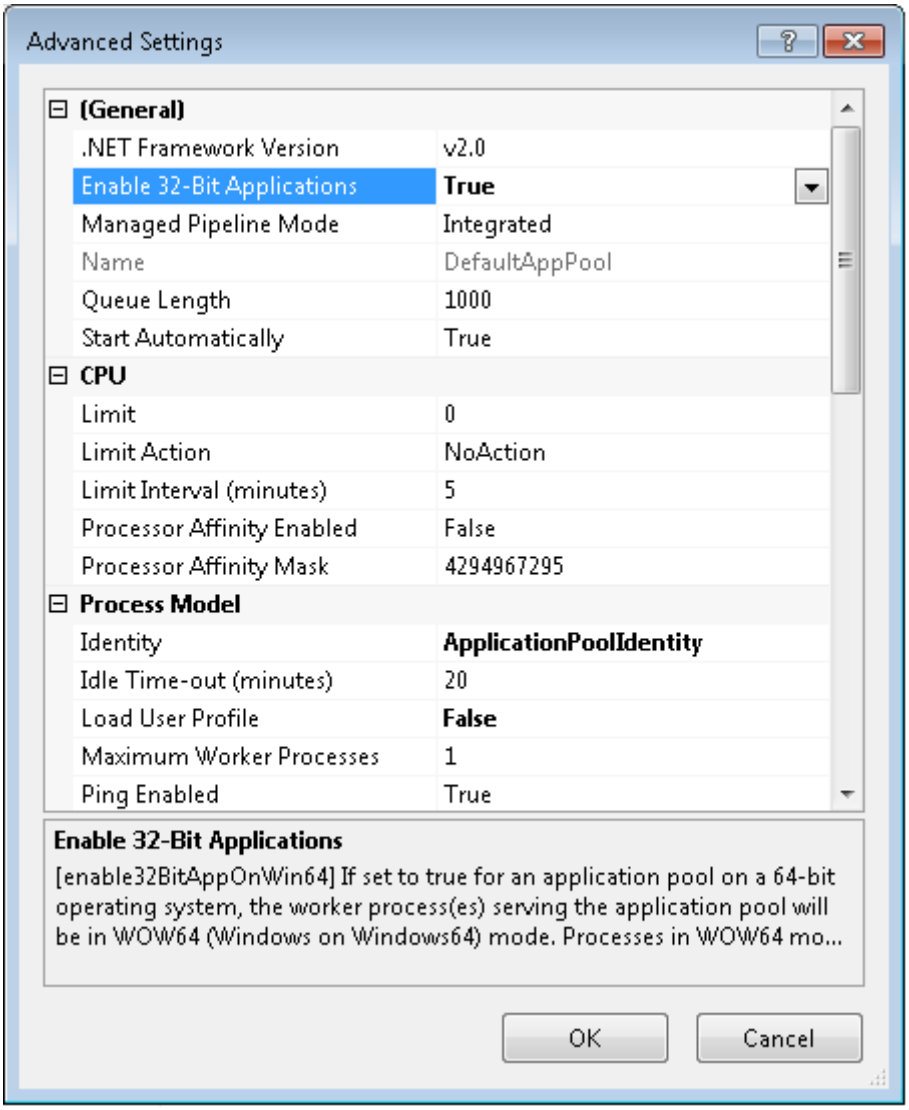
- As a next step you need to create an ISAPI allowance. Please select the root node in IIS Manager (named after your computername) and then double-click "ISAPI and CGI Restrictions".
- Click on "Add" to create a new enabled extensions. In the "ISAPI or CGI path" textbox, please specify the path to TcOpcXmlDa.dll, which normally is "C:\TwinCAT\OPC'Server4\Server XMLDA\TcOpcXmlDa.dll". Also select the checkbox "Allow extension path to execute".



- Restart your system.

Please note: If you use Windows 7 64-bit, you need to explicitely 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



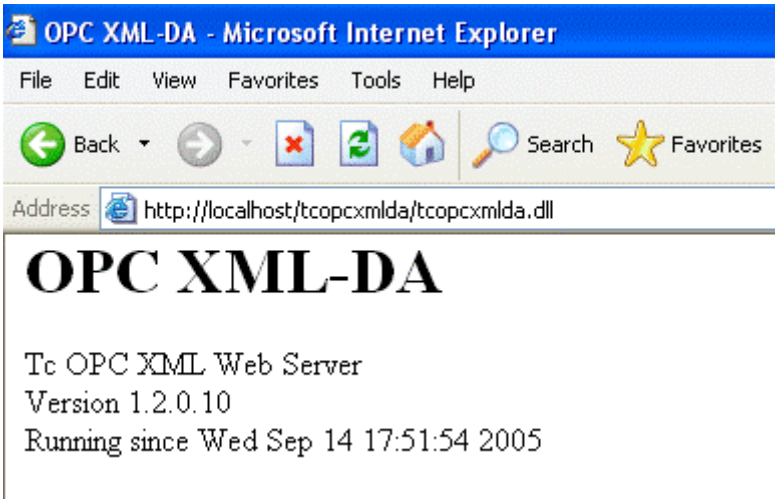
Step 3: Testing OPC-XML DA configuration

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.



Please note:
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.

4 Configuration

4.1 OPC DA Server

4.1.1 OPC DA Server

The TwinCAT 3 Function TF6120 provides an OPC-Server, which provides different features. The following table gives an overview about all features and links to the corresponding chapter in this documentation.

Feature	Description
General [▶ 24]	Provides documentation about general configuration settings.
Data Access (DA) [▶ 30]	Data Access provides the functionality to read/write process values from TwinCAT PLC or TwinCAT I/O.
Conversion [▶ 38]	Converting process values "on-the-fly"
Simulation [▶ 41]	Simulating process values without actually connecting to the underlying device.
Item properties [▶ 43]	Definition of Item Properties.
Data exchange via network [▶ 44]	Communicating with OPC via the network.

4.1.2 General

4.1.2.1 General configuration

The following tables gives an overview about all articles in this documentation, which describe general configuration parameters of the OPC-Server.

Feature	Description
Optimizations [▶ 24]	Describes some basic concepts about how to optimize the OPC-Server, e.g. by reducing its namespace.
OPC-Server as EXE oder Inproc (DLL) [▶ 25]	Describes different execution modes of the OPC-Server.
Automatic Cache Update [▶ 26]	Describes the feature of an automatic cache update after each write operation.
Description of the XML Configuration [▶ 28]	Contains a description of all parameters in the OPC-Server configuration file.

4.1.2.2 Optimizations

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

1. Minimize 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. POUs and global variables). After compiling the PLC-project the new symbolfile "<yourProjectName>.sym" will be in the same folder as your plc-project.

- 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 [▶ 30]

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

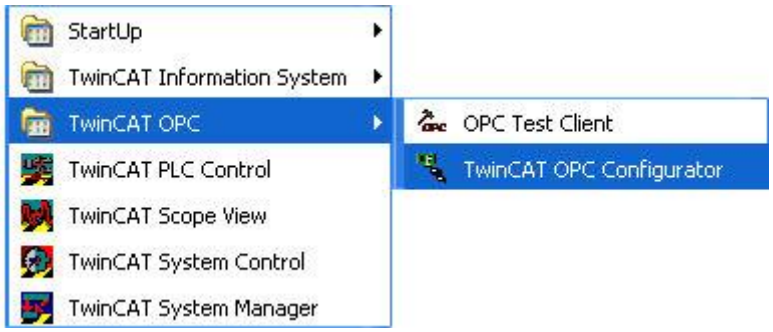
- Arguments for working with EXE :
  - One single OPC-client will access data from TwinCAT-OPC-Server
  - The OPC-Client is installed on a different PC , the network has to be crossed via DCOM.

- 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.

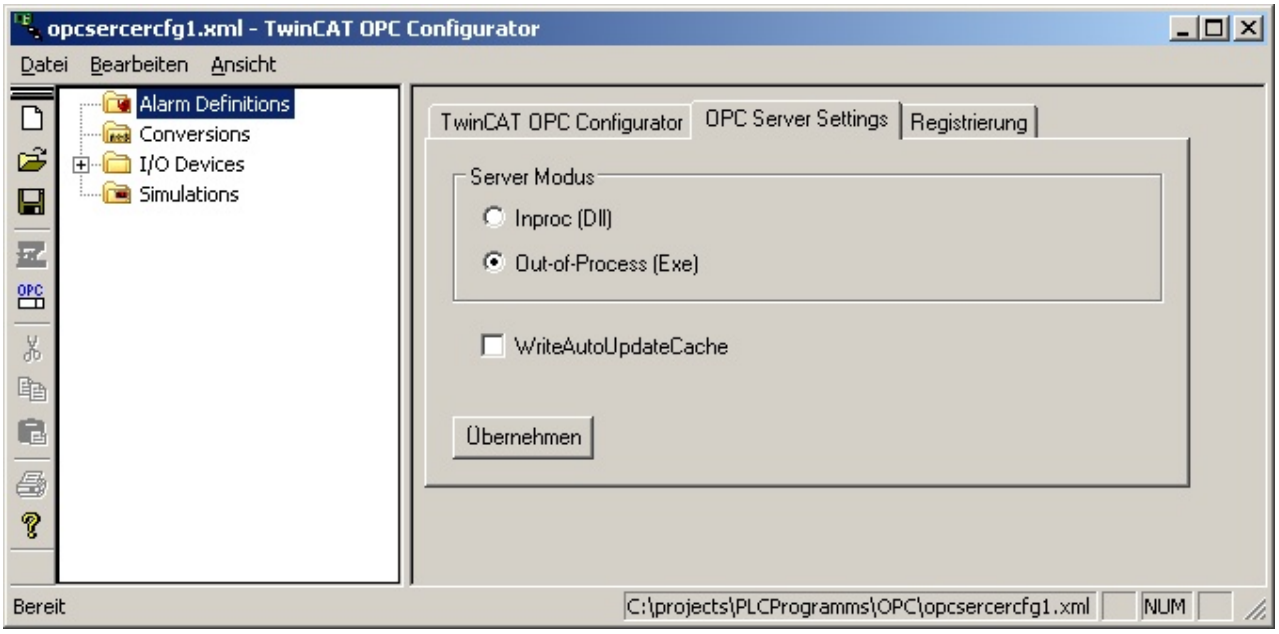
4.1.2.3 OPC Server as EXE / DLL

TwinCAT OPC-Server is available as an "out-of-process" server (with EXE as the file extension) and as an "in-process" server (with DLL as the file extension). When connecting to the server, the OPC-Client uses the ProgID to identify the OPC-Server. In case of the TwinCAT OPC-Server, the ProgID is **"BECKHOFF.TwinCATOpcServerDA"**, which is the same for Out-of-Process and In-Process. Both execution types can be configured via the OPC-Configurator.

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"** .



Activation of the TwinCAT OPC Server as Inproc 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 execution type is 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.

Aktivation of the TwinCAT OPC Servers as 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.

Notes about using DCOM

We do not advise the use of DCOM connections. If, however, a DCOM connection is essential, it is implemented with the type "Out-of-process" OPC server.

4.1.2.4 Automatic Cache Update

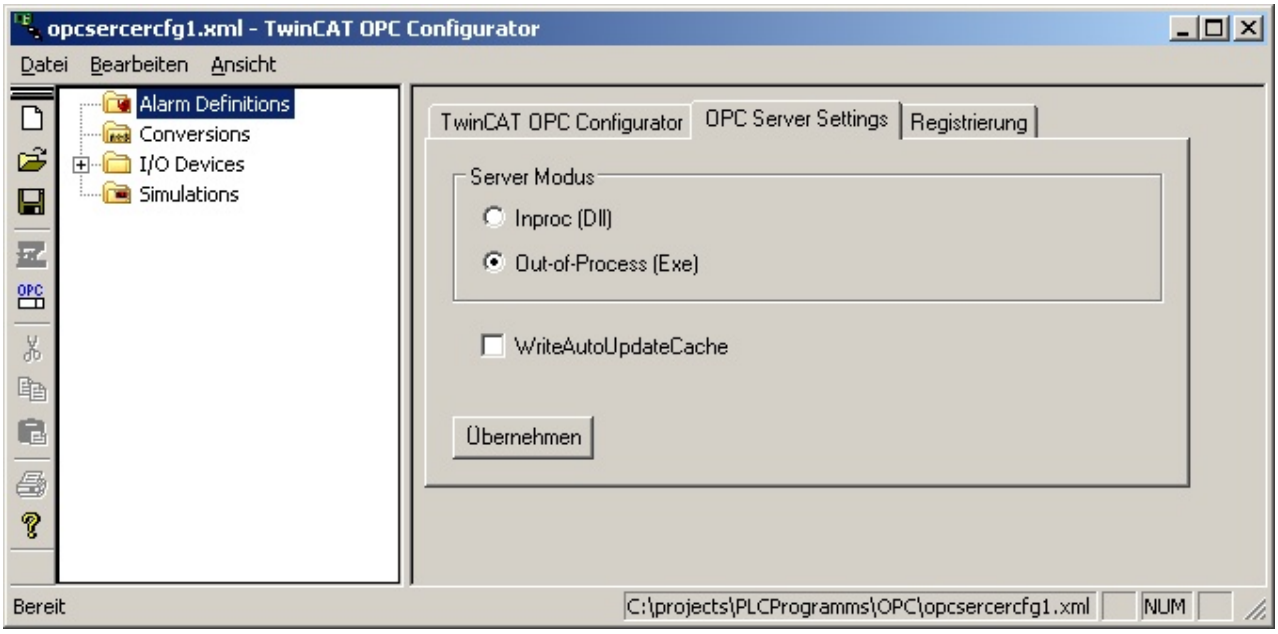
Regard platform

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"



4.1.2.5 Schemata

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" Note: If not specified or like "0.0.0.0.0.0", the OPC-Server will always communicate to AdsNetId of local ADS-router
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"
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

Type	Req./Opt.	Description
SignalID	Req	unique ID-number which identifies this simulation-signal
SignalDesc	Optional	
SignalType	Req	**0 : Read Count 1 : Write Count** **2 : Random** **3 : Ramp** **4 : Sine** **5 : Square** **6 : Triangle** **7 : Step** **8 : Reserver** **9 : Step Read Count** : incremented by one every time when the item is read**Write Count** : incremented by one every time when the item is written**Random:** generates random value within the Amplitude range starting with Position**Ramp**, **Sine**, **Square**, **Triangle**, **Step: (**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 axisSquare 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.
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

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 specifiedIf 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)

4.1.3 Data Access

4.1.3.1 Overview

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.

Configuration TwinCAT OPC Server

- Receiving data from the TwinCAT PLC [▶ 30]: Configuration by variable import from TwinCAT PLC control
- Receiving Data from the TwinCAT I/O task [▶ 35]: Configuration by variable upload from the TwinCAT I/O Task

4.1.3.2 Receiving data from TwinCAT PLC

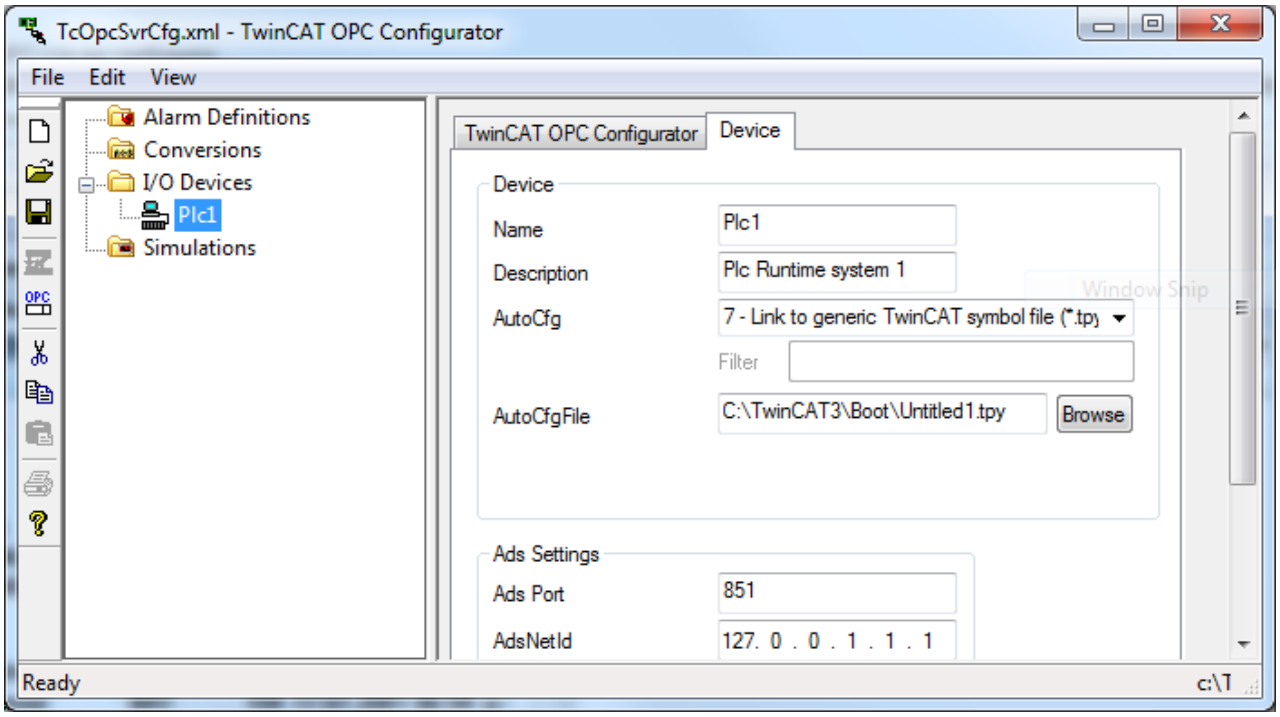
As mentioned bfore, the OPC-Server gives the possibility to read/write process values from ADS devices and therefore also from the TwinCAT PLC.

The following chapter gives you an overview about two different modes for accessing PLC symbols.

General

To configure the OPC-Server for PLC Data Access, you have two options. Both options can be configured via OPC-Configurator, which will be explained later in Step 2.

- **AutoCfg "7"** : All PLC variables are available via OPCAlle SPS Variablen sind über OPC verfügbar. Please note: This could result in a very big OPC namespace, depending on the amount of PLC variables, structures, etc.. As this setting may be sufficient for smaller PLC project, we highly recommend to use AutoCfg "8" for bigger projects.
- **AutoCfg "8"** : A subset of PLC variables is available via OPC.
  The subset will be defined by special PLC comments directly in the PLC project. We recommend to use this setting.



When using AutoCfg "8", the following steps need to be taken to make PLC variables accessible via OPC:

- Step 1: Configuration of variables in the PLC
- Step 2: Configuration of the OPC-Server (this is a one-time step)

When using AutoCfg "7", step 1 will become obsolete and you can directly move on to step 2.

Step 1: Configuration of variables in the PLC

To make a PLC variable accessible via OPC, this variable needs to be explicitely configured. This happens via special PLC comments directly behind a variable, a structure or an instance. Two better understand the behavior of these comments, here are two examples:

Sample 1:

In this sample, the PLC variables bMemFlag1, bMemFlag2 and iReadOnly are configured for OPC. The PLC variable bMemAlarm1 should not be accessible via OPC. The PLC program may look as follows:

```
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:available for OPC Clients but ReadOnly) *)
```

The comment OPC_PROP[0005]:1 causes that the variable is read only.

Sample 2:

This sample makes the instances fbTest1 and fbTest2 of the function block FB_BLOCK1 accessible via OPC. If an instance is made available via OPC, all contained variables are also published into the OPC namespace. The PLC program may look as follows:

```
PROGRAM MAIN
VAR
    fbTest1     :   FB_BLOCK1;  (*~ (OPC : 1 : available for OPC-clients)   *)
    fbTest2     :   FB_BLOCK1;
END_VAR

FUNCTION_BLOCK FB_BLOCK1
VAR_INPUT
    ni1 :   INT;    (*~ (OPC : 1 : available for OPC-clients)   *)
    ni2 :   INT;
END_VAR
VAR_OUTPUT
    no1 :   INT;    (*~ (OPC : 1 : available for OPC-clients)   *)
    no2 :   INT;
END_VAR
VAR
    nx1 :   INT;    (*~ (OPC : 1 : available for OPC-clients)   *)
    nx2 :   INT;
END_VAR
```

The instance fbTest1 has been made available via OPC, therefore all symbols within are also available via OPC, e.g. fbTest.ni1, fbTest.ni2, ... The instance fbTest2 has not been configured for OPC, however, the function block defines the three variables ni1, no1 and nx1 for OPC. These are therefore OPC-enabled in all instances.

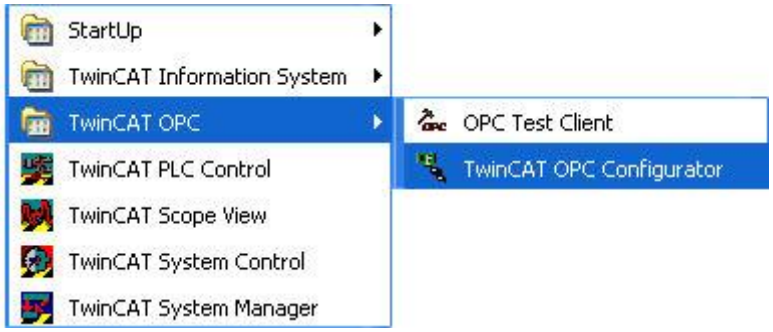
After the first PLC project has been compiled, the project directory contains a TPY-File, which needs to be configured in the OPC-Server, see Step 2. This TPY-File includes information about all PLC variables and whether a variable has been configured for OPC or not.

Please note: When making structural changes to the PLC program, this TPY-File may also change and you may need to re-start the OPC-Server.

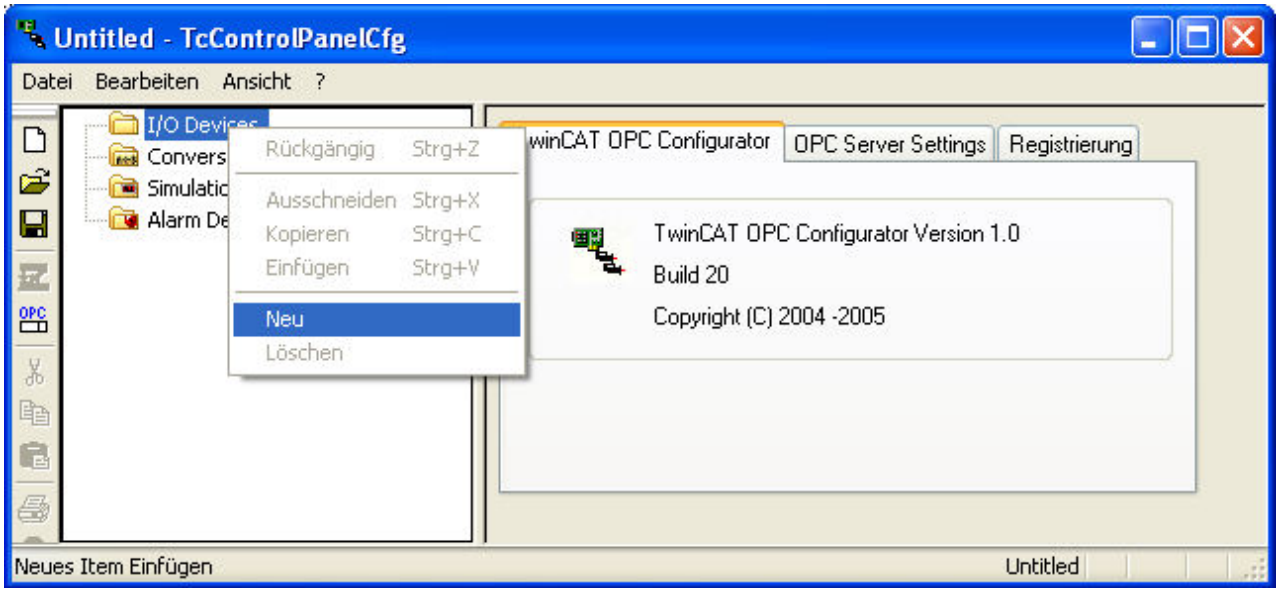
Step 2: Configuration of the OPC-Server (one-time step)

In the second step you need to configure the OPC-Server. This is a one-time configuration.

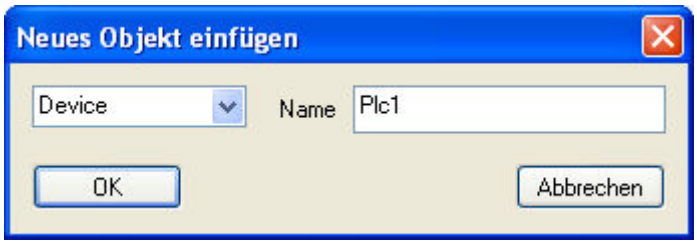
Start the TwinCAT-OPC-Configurator **"Start - All Programs - TwinCAT System - TwinCAT OPC - TwinCAT OPC Configurator"**



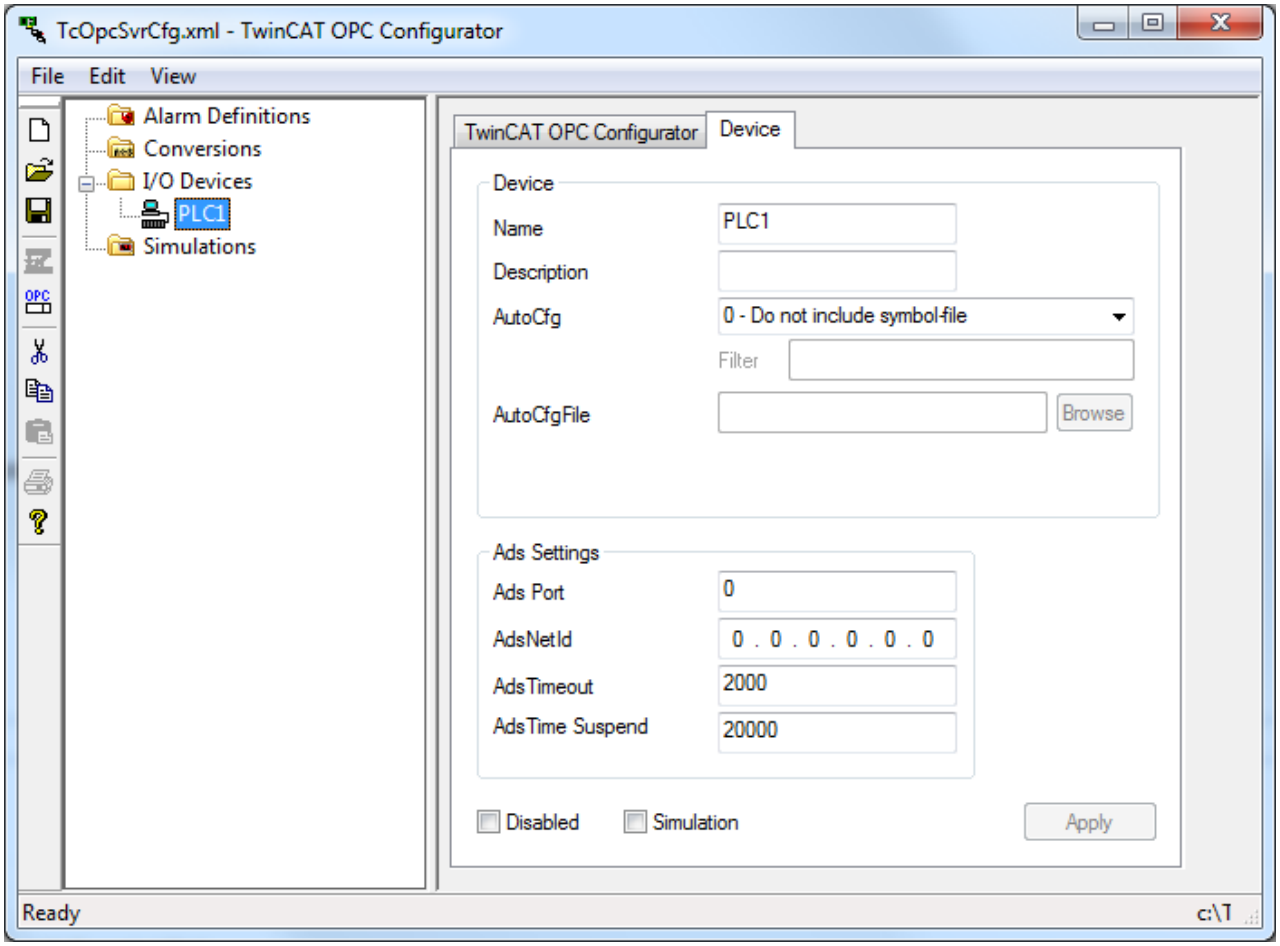
Select **"I/O Devices"** with right click and then menu **"New"**.



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.



The new device has been added and you can now configure it according to your system environment.



The following table gives an overview about all possible configuration settings.

Parameter	Beschreibung
Name	Name of the device, e.g. PLC1. This will be the name under which the device will be accessible for the OPC-Client.
Description [optional]	Optional description for the device.
AutoCfg	7 : All PLC variables are available via OPC 8 : Only a subset of PLC variables is available via OPC
AutoCfgSymFile	Path to TPY-File, which is by default located in the PLC project directory.
AdsPort	ADS port numer of the device, e.g. 851 for the first PLC runtime.
AdsNetId [optional]	Address of the ADS device. By default this is 0.0.0.0.0.0 for local system.
AdsTimeout [optional]	Timeout for ADS connection to the device, measured in [ms]. If the device is not reachable within that timeframe, the OPC-Server returns BAD_QUALITY to the OPC-Client.
AdsTimeSuspend [optional]	Suspend time for the ADS device, measured in [ms]. If the ADS connection breaks, the OPC-Server waits this time before sending the next ADS request to the runtime.
Disable [optional]	Deactivates the device.

Save the configuration via "File" --> "Save As". After the configuration has been saved, you will be asked if this configuration should be set as the startup-configuration.

The configuration will be automatically activated on next restart of the OPC-Server.

4.1.3.3 Receiving data from I/O-Task

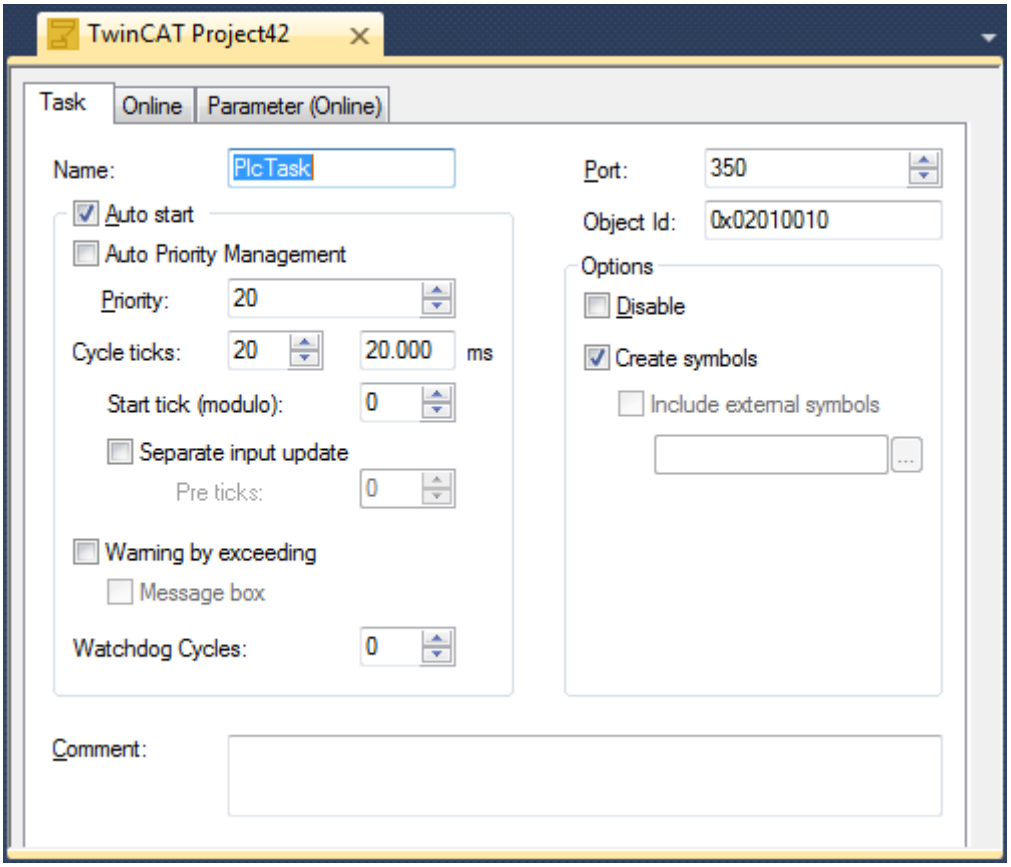
As mentioned bfore, the OPC-Server gives the possibility to read/write process values from ADS devices and therefore also from the TwinCAT I/O.

The following steps need to be taken to make an I/O task available via OPC:

- Step 1: Configuration of I/O Task
- Step 2: Configuration of the OPC-Server (one-time step)

Step 1: Configuration of I/O-Task

To make an I/O Task available via OPC, please open the TwinCAT Solution and navigate to "SYSTEM" - "Tasks" - "TaskName".

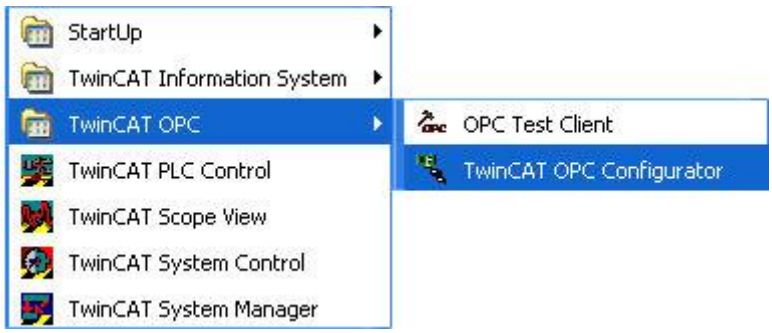


Activate the checkbox **"Create Symbols"** .

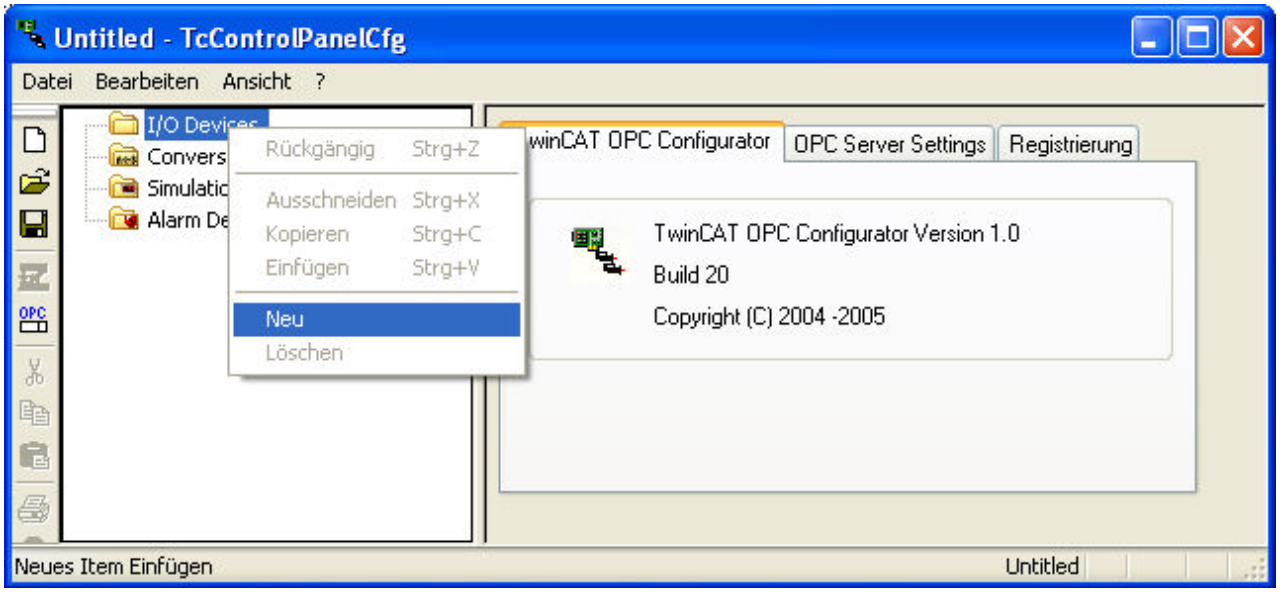
Step 2: Configuration of the OPC-Server (one-time step)

In the second step you need to configure the OPC-Server. This is a one-time configuration.

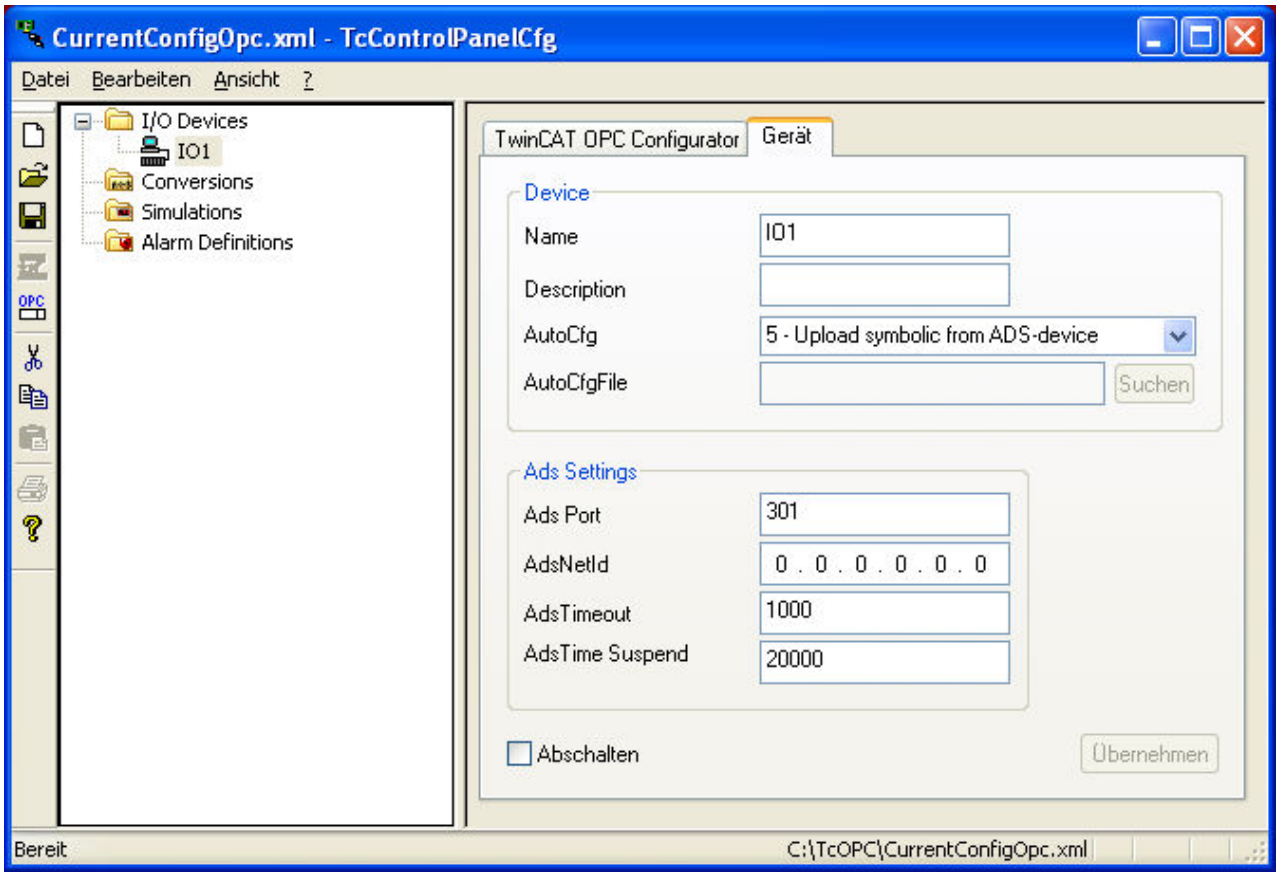
Start TwinCAT-OPC-Configurator **"Start \ All Programs\ Beckhoff \ TwinCat3 Functions \ TF6120 OPC-DA"**



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 :



The following table gives an overview about all possible settings:

Parameter	Beschreibung
Name	Name of Name device, e.g. IO1. This will be the name under which the device will be accessible for the OPC-Client.
Description [optional]	Optional description for the device.
AutoCfg	5 : Get symbolic via ADS
AutoCfgSymFile	Not needed in this case.
AdsPort	ADS port number of device, e.g. 351 for I/O Task
AdsNetId [optional]	Address of the ADS device. By default this is 0.0.0.0.0.0 for local system.
AdsTimeout [optional]	Timeout for ADS connection to the device, measured in [ms]. If the device is not reachable within that timeframe, the OPC-Server returns BAD_QUALITY to the OPC-Client.
AdsTimeSuspend [optional]	Suspend time for the ADS device, measured in [ms]. If the ADS connection breaks, the OPC-Server waits this time before sending the next ADS request to the runtime.
Disable [optional]	Deactivates the device.

Save the configuration via "File" --> "Save As". After the configuration has been saved, you will be asked if this configuration should be set as the startup-configuration.

The configuration will be automatically activated on next restart of the OPC-Server.

Apply settings

If you have activated a configuration and the OPC-Server does not seem to apply the settings, please make sure that the OPC-Server has been stopped and started once.

4.1.4 Conversion

4.1.4.1 Configuration of OPC-Conversions

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 --> convertion within OPC-Server --> Communication to OPC-Client --> OPC-Client
- OPC-Client --> Communication to OPC-Server --> convertion 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.

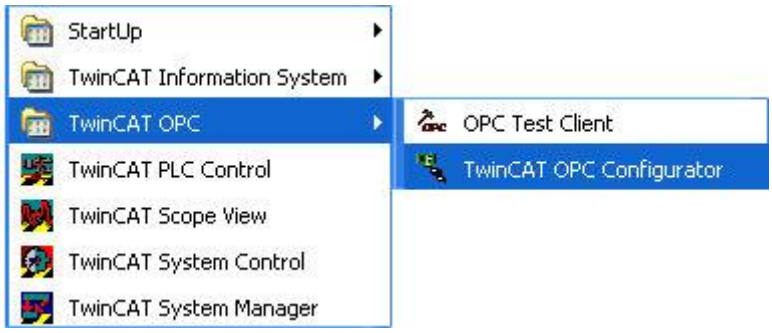
The OPC-Server presents a converted variable as data type "VT_R8 / Double" to OPC-client - independent of the actual numerical datatyp in the PLC.

The usage of OPC-Conversions needs the configuration of a TPY-File (AutoCfgSymFile) for the corresponding device and AutoCfg 8.

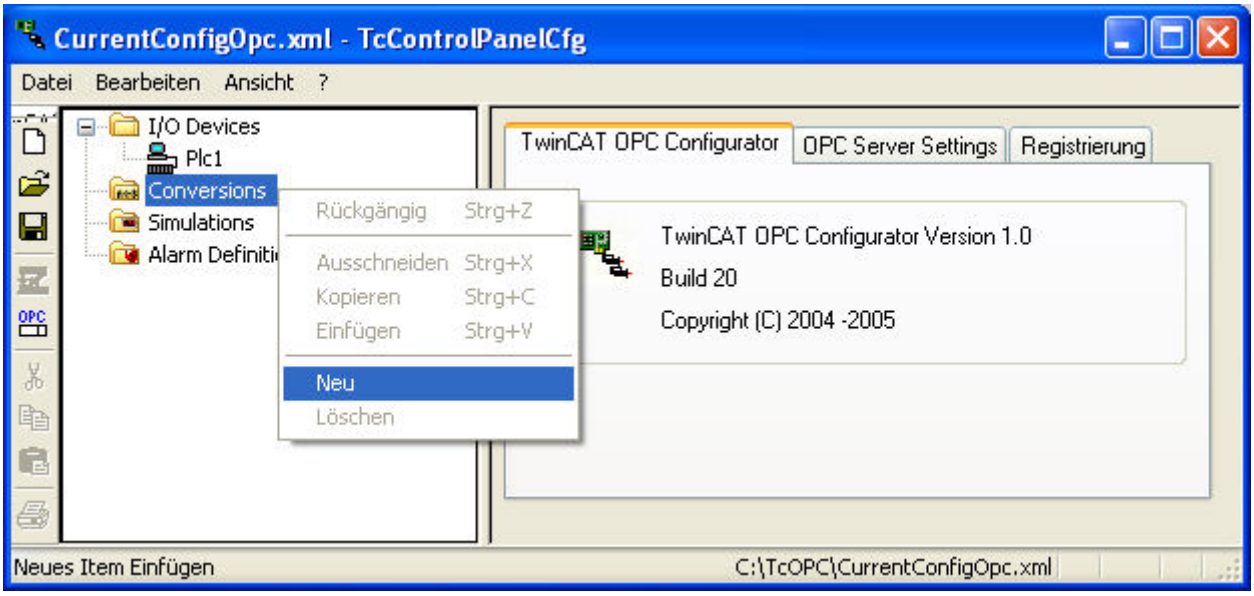
Step 1: Configuration of OPC-Conversions in OPC-Server

In the first step you need to configure TwinCAT OPC-Server for conversions. This is a one-time configuration and does not need to be repeated when changes to the PLC project occur.

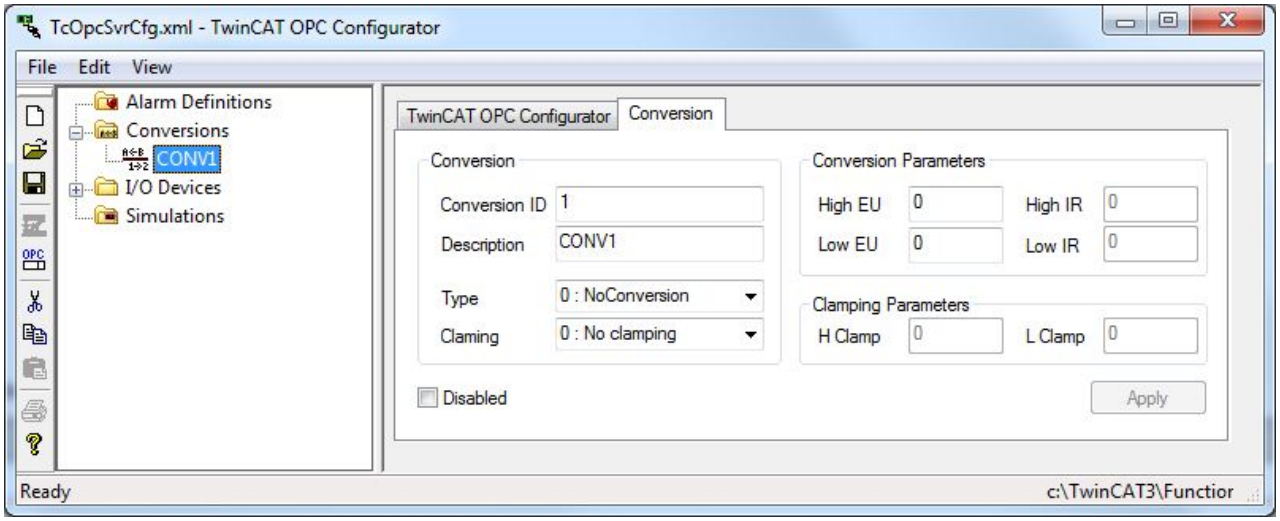
Start TwinCAT-OPC-Configurator **"Start - All Programs - TwinCAT System - TwinCAT OPC - TwinCAT OPC Configurator"**



Navigate to **"Conversions",** right click and select menue **"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"**.



The following table gives an overview about all possible configuration settings.

Parameter	Beschreibung
Conversion ID	ID which is important for step 2.
Type	0: No conversion. 1. Linear conversion.
Clamping [optional]	Defines a maxmium and minimum value. If this value is higher/lower, it will be clamped.
High EU	
Low EU [optional]	
High IR [optional]	
Low IR [optional]	
H Clamping [optional]	
L Clamping [optional]	
Disable [optional]	Deactivates the conversion setting.

Save the configuration via "File" --> "Save As". After the configuration has been saved, you will be asked if this configuration should be set as the startup-configuration.

The configuration will be automatically activated on next restart of the OPC-Server.

Step 2: Configuration of PLC-Variables for OPC-Conversion

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. Similar to the Data Access feature, this configuration also occurs by adding comments to the PLC-project.

Beispiel:

```
PROGRAM MAIN
VAR
    bBool1  :   BOOL;   (*~ (OPC : 1 : available for OPC-Clients)
                            (OPC_PROP[6010] : 1 : OPC_PROP_CONV_ENABLE)
                            (OPC_PROP[6011] : 42 : OPC_PROP_CONV_ID)
                        *)
END_VAR
```

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

4.1.5 Simulation

4.1.5.1 Configuration of OPC-Simulations

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 OPC-Server offers:

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

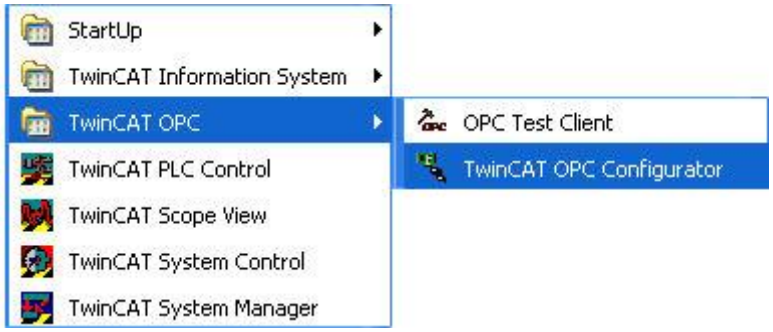
It is not possible to combine both simulated and real-process values.

The usage of OPC-Conversions needs the configuration of a TPY-File (AutoCfgSymFile) for the corresponding device and AutoCfg 8.

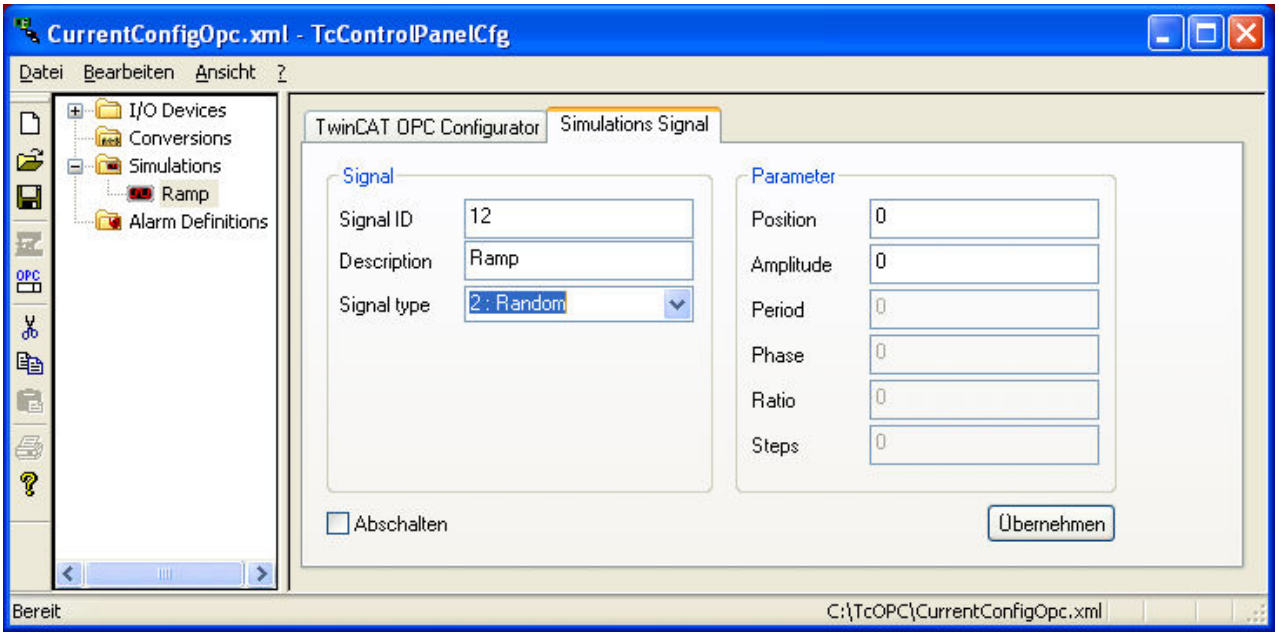
Step 1: Configuration of OPC-Simulation in OPC-Server

In the first step you need to configure TwinCAT OPC-Server for conversions. This is a one-time configuration and does not need to be repeated when changes to the PLC project occur.

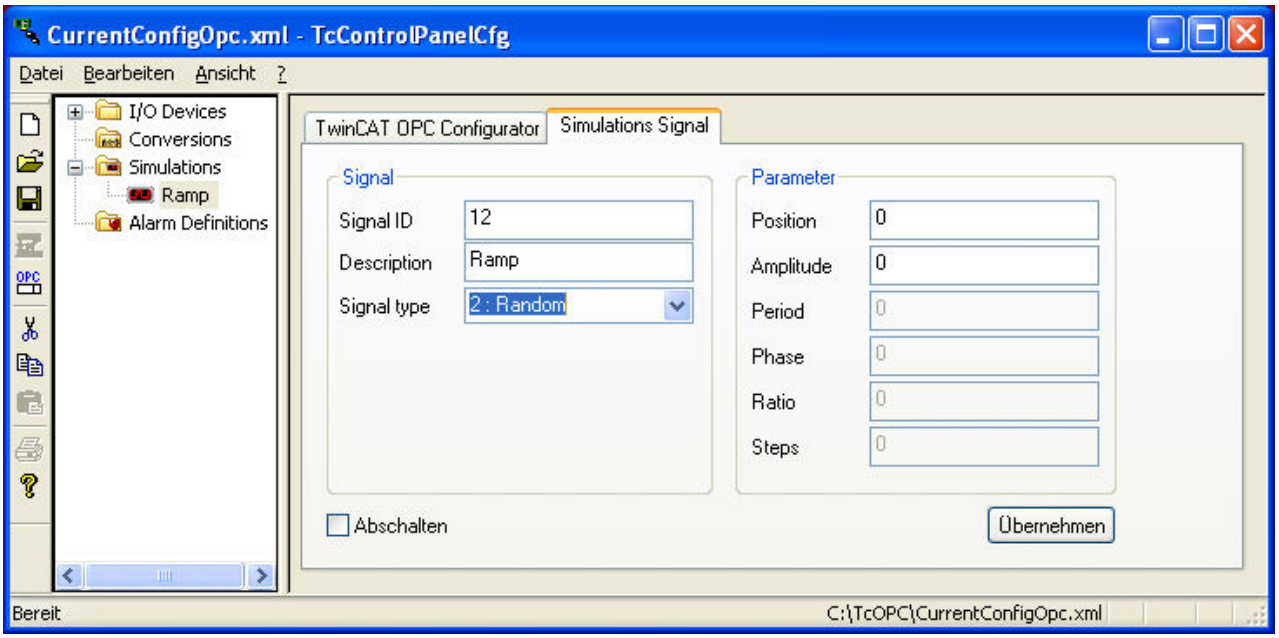
Start TwinCAT-OPC-Configurator **"Start - All Programs - TwinCAT System - TwinCAT - TwinCAT OPC Configurator"**



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



For **"Name"** please enter a name for this simulation template. This name has to be OPC conform, no special characters allowed. With "OK" you get into the dialog to configure in details your new **"Simulations Signal"**.



The following table gives an overview about all possible configuration settings.

Signal Type	Beschreibung
0 : ReadCount	
1 : WriteCount	
2 : Random	Random-Function
3 : Ramp	Ramp-Function
4 : Sine	Sinus-Function
5 : Square	Square-Function
6 : Triangle	Triangle-Function
7 : Stepp	Step-Function
Disable	Deactivates the simulation

Save the configuration via "File" --> "Save As". After the configuration has been saved, you will be asked if this configuration should be set as the startup-configuration.

The configuration will be automatically activated on next restart of the OPC-Server.

Step 2: Configuration of PLC-Variables for OPC-Simulation

With previous steps we defined a "simulation-template" with detailed information about simulation handling. Now we define, which PLC variable should be handled with simulation. Similar to the Data Access feature, this configuration also occurs by adding comments to the PLC-project.

Beispiel:

```
PROGRAM MAIN
VAR
    bBool1  :   BOOL;   (*~ (OPC : 1 : available for OPC-Clients)
                            (OPC_PROP[6006] : 1 : OPC_PROP_SIMU_ENABLE)
                            (OPC_PROP[6007] : 42 : OPC_PROP_SIMU_ID)
                        *)
END_VAR
```

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

4.1.6 Item Properties

4.1.6.1 Configuration of Item Properties

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

The configuration of these properties occurs in the PLC program by adding comments behind a symbol. These comments will be interepreted by the OPC-Server.

Sample:

```
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.

The following table shows a list of all Item Properties:

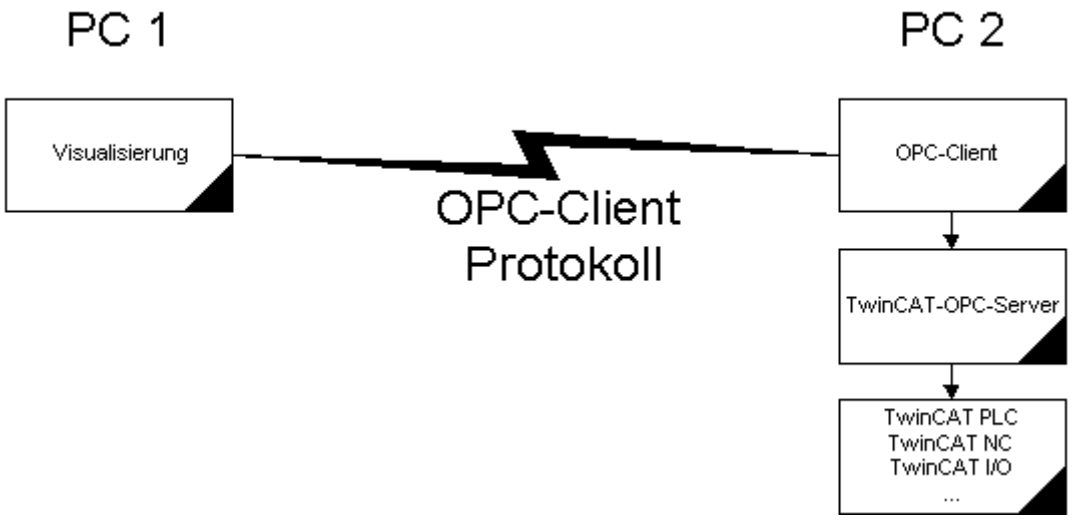
Property ID laut OPC Spezifikation	Description
OPC_PROP[0005]	OPC_PROP_RIGHTS 1 : OPC-Server will publish this variable with access : OPC_READABLE 2 : OPC-Server will publish this variable with access : OPC_WRITEABLE 3 : OPC-Server will publish this variable with access : OPC_READABLE AND OPC_WRITEABLE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

4.1.7 Data exchange via network

4.1.7.1 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:



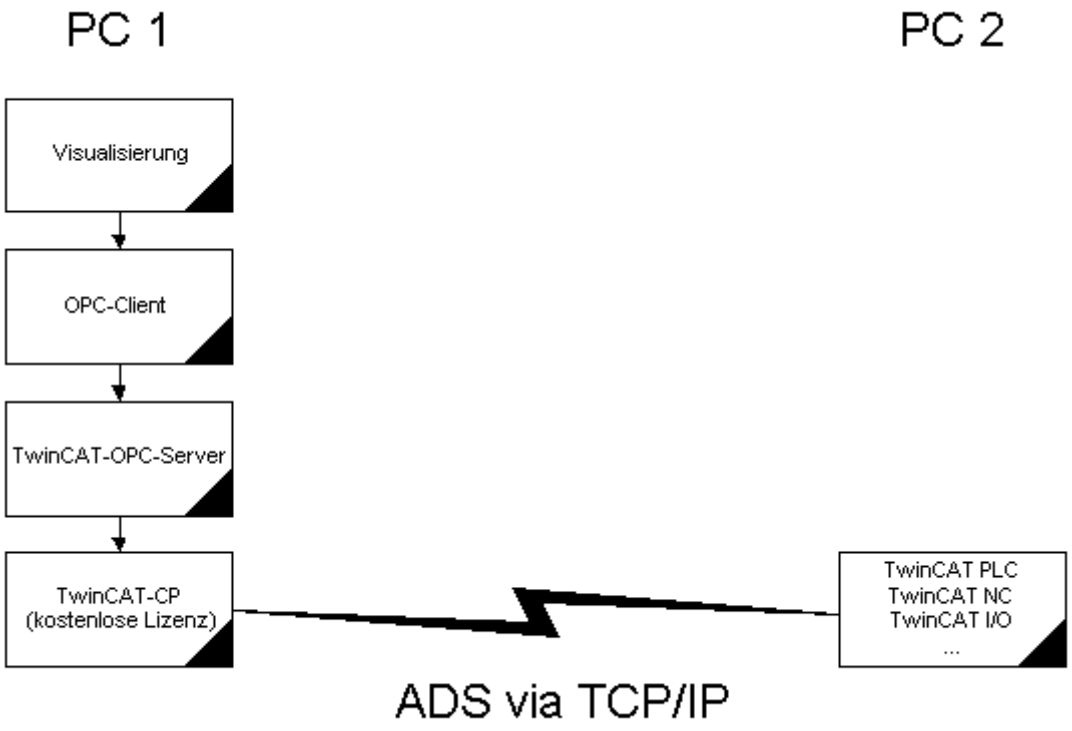
Performance advantage

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.

4.1.7.2 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

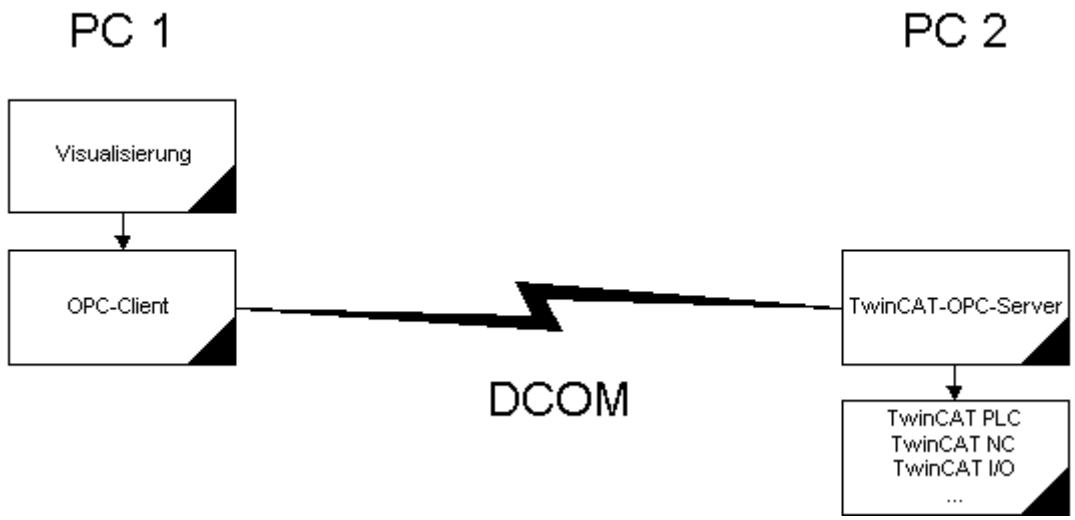
An ADS communication partner is always specified by two parameters: These are what is known as the "AdsAmsNetId" (e.g. 1.2.3.4.5.6) and the "PortId" (e.g. 801 for the first PLC run-time system).

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 [▶ 44]" protocol should be selected for performance reasons.

4.1.7.3 Netzwerk via DCOM

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

Prerequisites

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).

Step 1

Click on "Start"->"Run…" and enter "DCOMcnfg", then select the dialog "Default Settings".

- Activate "Enable Distributed COM on this computer"
- For "Default Authentication Level" select "None"
- For "Default Impersonation Level" select "Anonymous

Step 2

Select the dialog "Applications" and then the entry „TwinCAT OPC Server DA". Double-click this entry.

Step 3

Select the dialog "Location" and then activate "Run application on this computer".

Step 4

Select the dialog "Security":

- Activate the checkbox "Use custom access permissions" and select "Everyone"
- Activate the checkbox „Use custom configuration permissions" and select "Full Control" for "Everyone"
- Activate the checkbox „Use custom launch permissions" and select "Everyone"

4.2 OPC XML-DA Server

4.2.1 OPC XML DA

In addition to the OPC DA Server, the Function TF6120 also contains an OPC XML-DA Server, which will be automatically installed by Setup. Depending on the used Windows Operating System, you may need to perform additional settings to setup XML-DA in the IIS webserver. Please consult the chapter "Installation" to see how to do that.

OPC XML-DA provides a remote OPC communication via Web Service, therfore you do not need to configure any DCOM settings. By default, the OPC XML-DA Server is being pre-configured and you only need to specify the TPY-File of the corresponding PLC project.

The configuration file is located in the directory **"...\wwwroot\TcOpcXmlDa\TcOpcXmlSvrCfg.xml"** and contains a list of all ADS devices which should be made accessible via the OPC XML-DA interface. By default, this file already contains an ADS device for the first PLC runtime of the local system and its corresponding TPY-File C:\TwinCAT\Boot\CurrentPlc_1.tpy.

```
<?xml version="1.0"
encoding="utf-8" standalone="no"?>
<TcOpcXmlSvrConfiguration>

<Namespace>

<Devices>
    <Device>

     <Name>PLC1</Name>

     <AdsNetId>0.0.0.0.0.0</AdsNetId>

     <AdsPort>801</AdsPort>

     <AdsTimeout>2000</AdsTimeout>



<AdsTimeSuspend>20000</AdsTimeSuspend>

     <AutoCfg>7</AutoCfg>


<AutoCfgSymFile>C:\TwinCAT\Boot\CurrentPlc_1.tpy</AutoCfgSymFile>



     <Disabled>0</Disabled>

    </Device>
</Devices>

</Namespace>
</TcOpcXmlSvrConfiguration>
```

The TPY-File will be automatically generated during the PLC compilation process and is located in the same directory as the PLC-project. You need to reference the correct TPY file in the configuration so that the symbol information can be properly read from the ADS device. The OPC XML-DA Server automatically reads this file during startup.

You can always adapt the above configuration file to your needs, for example if you want to make additional ADS devices accessible via OPC XML-DA. You only need to create a new area <Device>...</Device> and fill it with the corresponding settings of your ADS device.

4.2.2 Status information

This chapter provides some useful information about the OPC XML-DA Server. The XML-DA specification describes the interfaces which an OPC XML-DA Server should provide. However, there are some additional functionalities which will help you on your daily work with the OPC XML-DA Server, for example:

- Reading status information
- Reading protocol information
- Reload configuration
- Show active configuration
- Restart OPC XML-DA Server
- Stop OPC XML-DA Server

These functionalities can be called via the URL of the OPC XML-DA Server. Open your favorite web browser and enter the following URL:

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

You can now call the mentioned services Via the parameter <Services>.

The following table gives some examples.

<Services>	Description
	Reading status information of the OPC XML-DA Server
?info=log	Reading protocol information of the OPC XML-DA Server
?info=reload	Reload the configuration
?info=config	Show active configuration
?action=restart	Restart the OPC XML-DA Server
?action=stop	Stop the OPC XML-DA Server

5 Appendix

5.1 OPC Compliance Certificate

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

Server:	Beckhoff.TwinCATOPcServerDA (Local Server)
Compliance Test Result:	Compliant
Date:	11/6/2003 11:59:31 PM
Company:	Beckhoff Industrie Electronic
OS version:	Windows NT 5.1.2600 Service Pack 1
Version info:	*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
Optional interfaces (supported):	IOPCBrowseServerAdressSpace IPersistFile
Optional interfaces (unsupported):	IOPCServerPublicGroups IOPCPublicGroupStateMgt
Data types for logical test (available):	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
Data types for logical test (unavailable):	VT_DATE VT_CY SAFEARRAY OF VT_BOOL SAFEARRAY OF VT_DATE SAFEARRAY OF VT_BSTR SAFEARRAY OF VT_I1 SAFEARRAY OF VT_CY
Number of supported groups:	>= 10
Not available for test:	
Number of test items:	2 Readable items 2 Writeable items 54 Read/Writeable items

5.2 DCOM

5.2.1 Overview

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

Important Notes

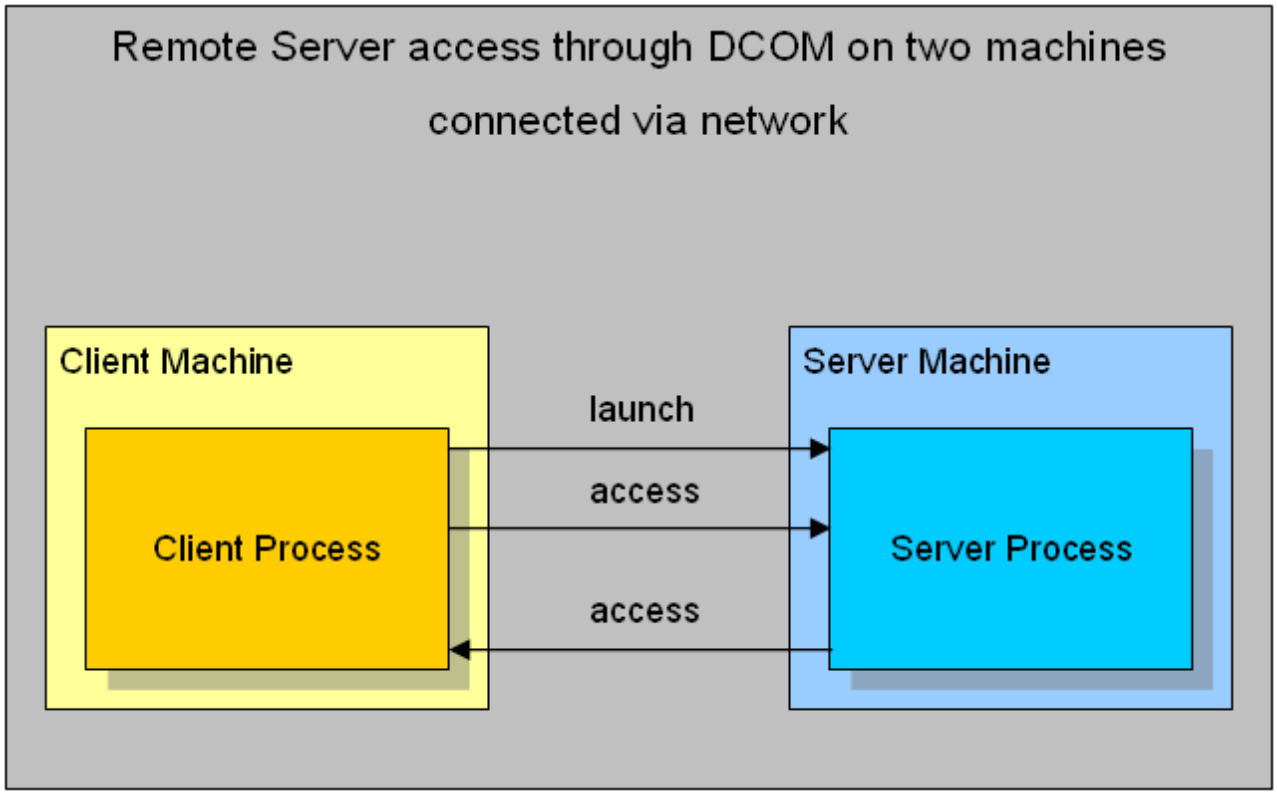
Before changing any settings of DCOM security, a system recovery procedure should be prepared. All settings described in this documentation should be tested in a laboratory environment, before changing machines in production.

- 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 comunicate with each other. There are tree 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 it's own memory area, the Server is a EXE and runs local on the same machine, but in it's own process
- **REMOTE-Server**
  The COM Server runs in it's own memory area, the Server is a EXE and runs remotely on a different machine and (of course) in it's own process as the Client

The communication between Client and Server process followes the rules of COM. Whenever the Client is located on one machine and the Server on an other, 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.

5.2.2 Prerequisites

5.2.2.1 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 form 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 authenicate 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 authentification 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 authenicate 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 bevavior when lauching and accessing the DCOM Server, but there will be no callbacks comming 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 authenicate 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 authentification 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 authenicate 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 bevavior when lauching and accessing the DCOM Server, but there will be no callbacks comming through.

5.2.2.2 Network configuration

5.2.2.2.1 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 [▶ 54]

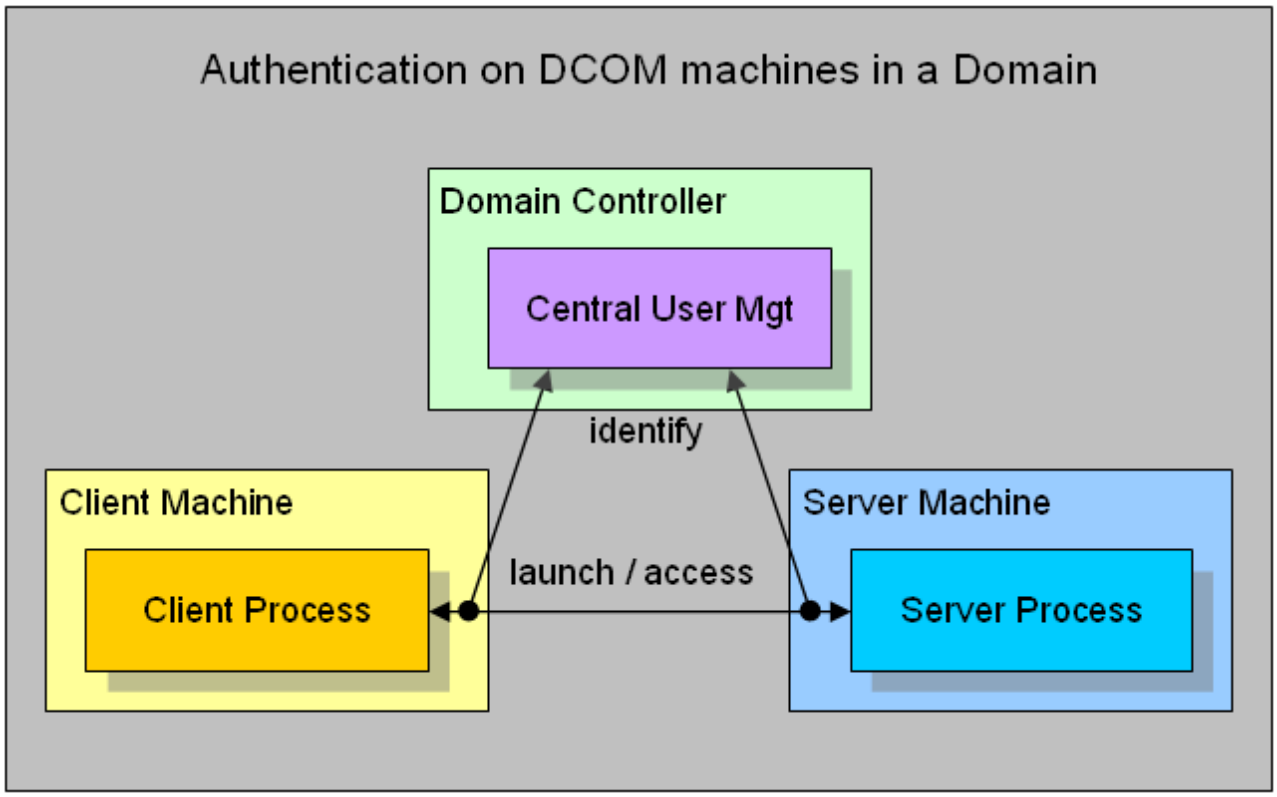
- Domain [▶ 53]

5.2.2.2.2 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 authentification. If both machines are in different domains, these domains must trust each other. The administrative effort will decreases 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 should be trust eachother.
- **Users Accounts**
  Authentification 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.



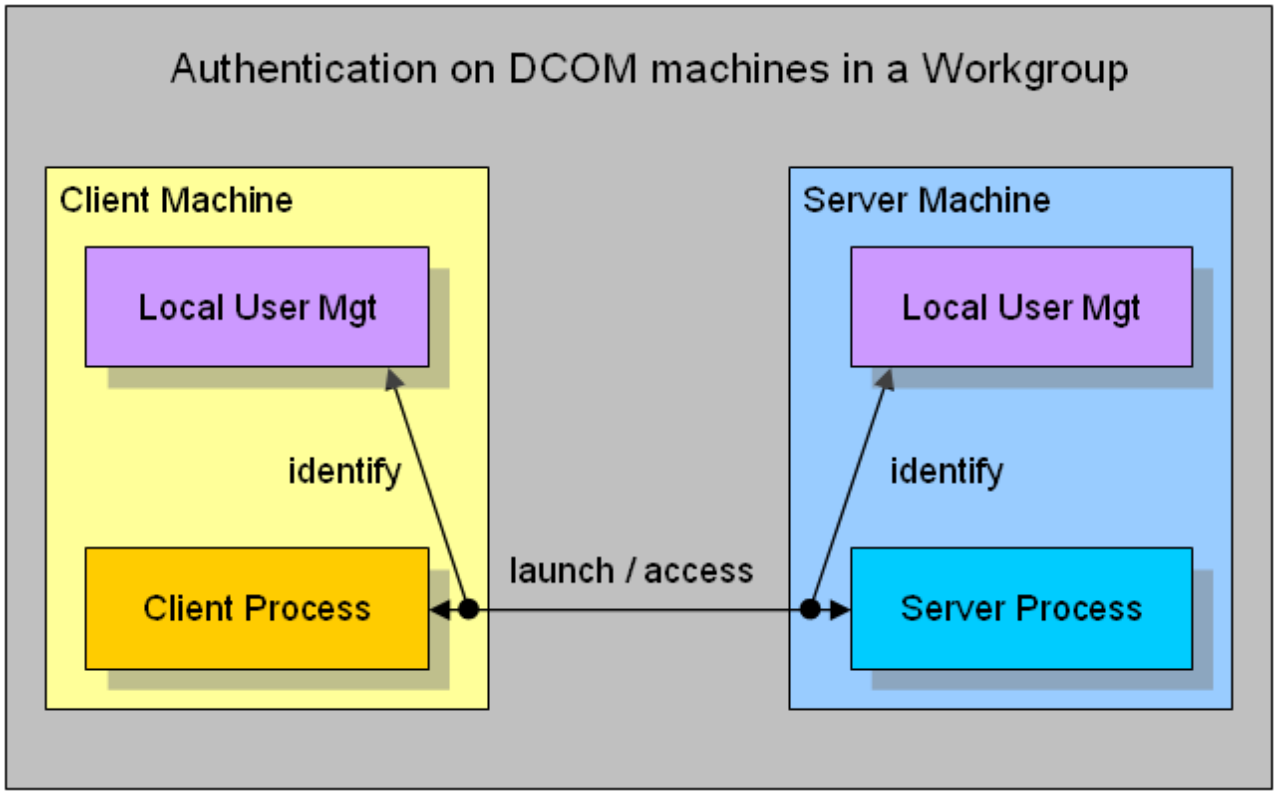
In a mixed configuration (e.g. Server Machine in a Domain and Client Machine in a Workgroup) the so called doubble identification should be done. This means to add User Accounts to the Workgroup Machine which are identically (Name and PWD) the same as the User Accounts on the Domain. By this function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

5.2.2.2.3 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 authentification. 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 imensely when adding new users.

- **Workgroup**
  The Client and Server Machine should be member of the same Workgroup.
- **Users Accounts**
  As authentification 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 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 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.



5.2.3 Client

5.2.3.1 Client Machine

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 runing.

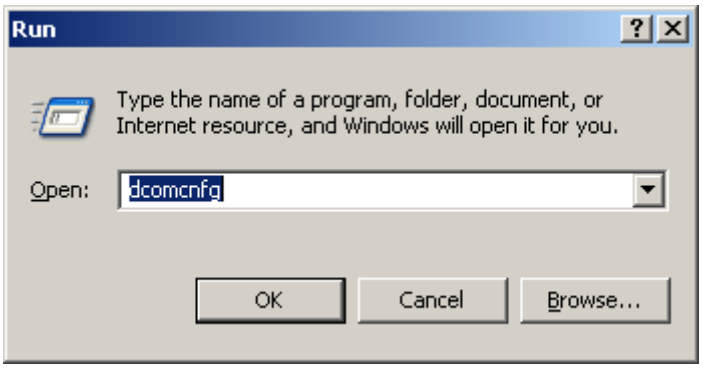
Please refer to our Article about Operating Systems [▶ 51] for more information.

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 have to be made. The default settings will be correct for OPC Client side security settings

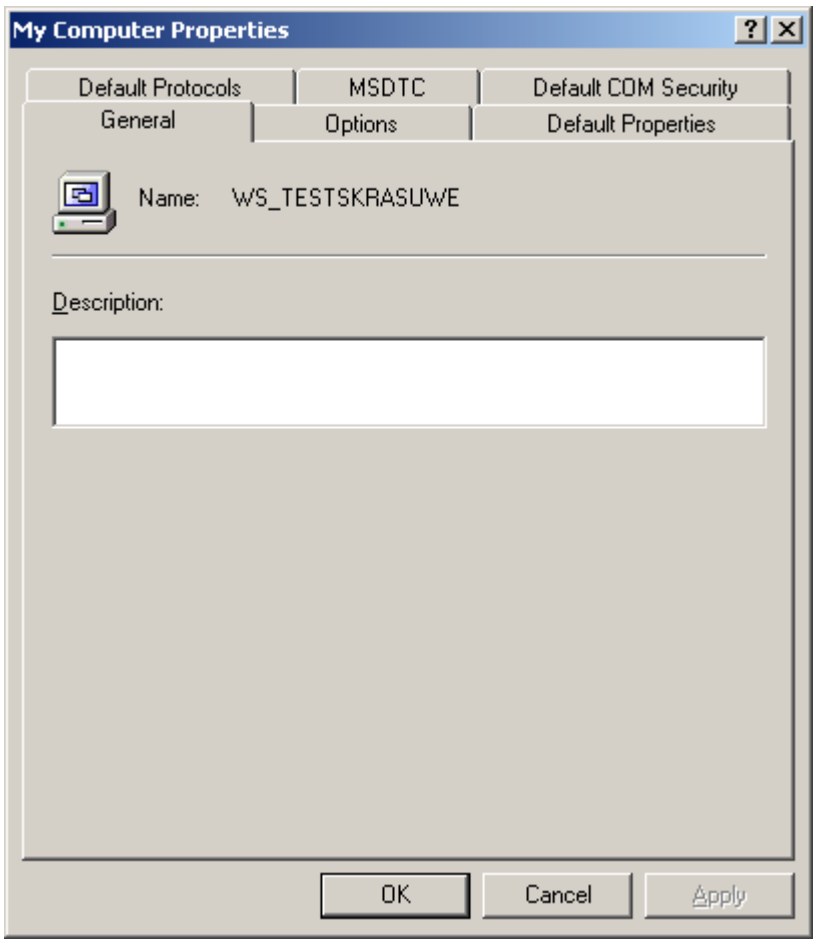
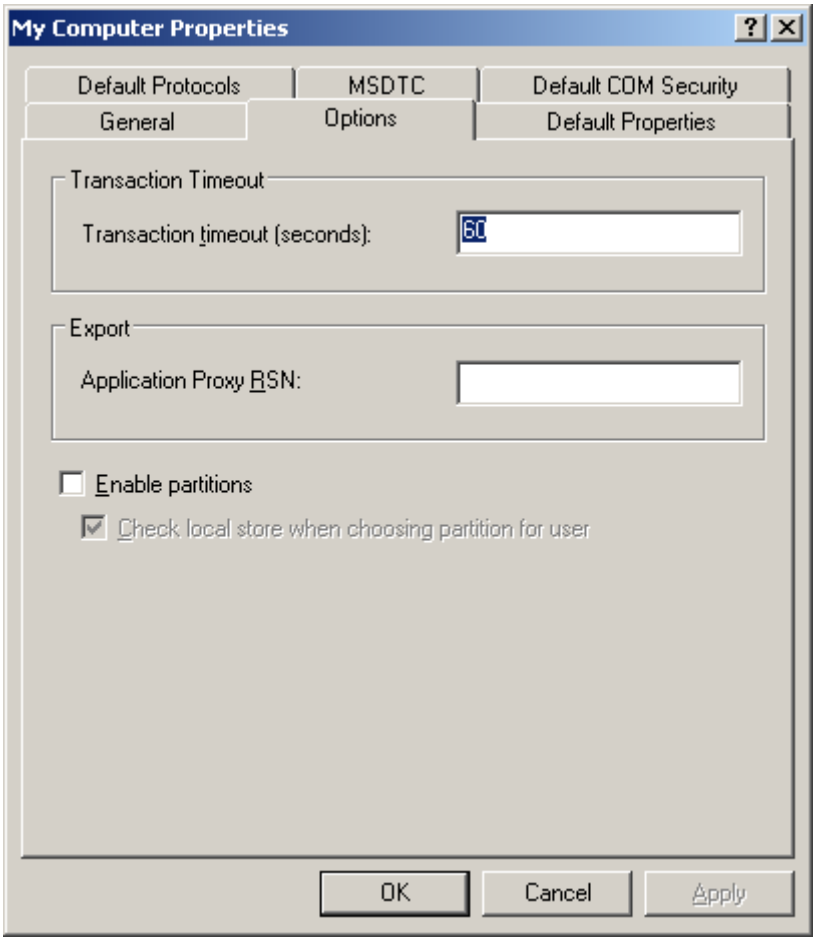
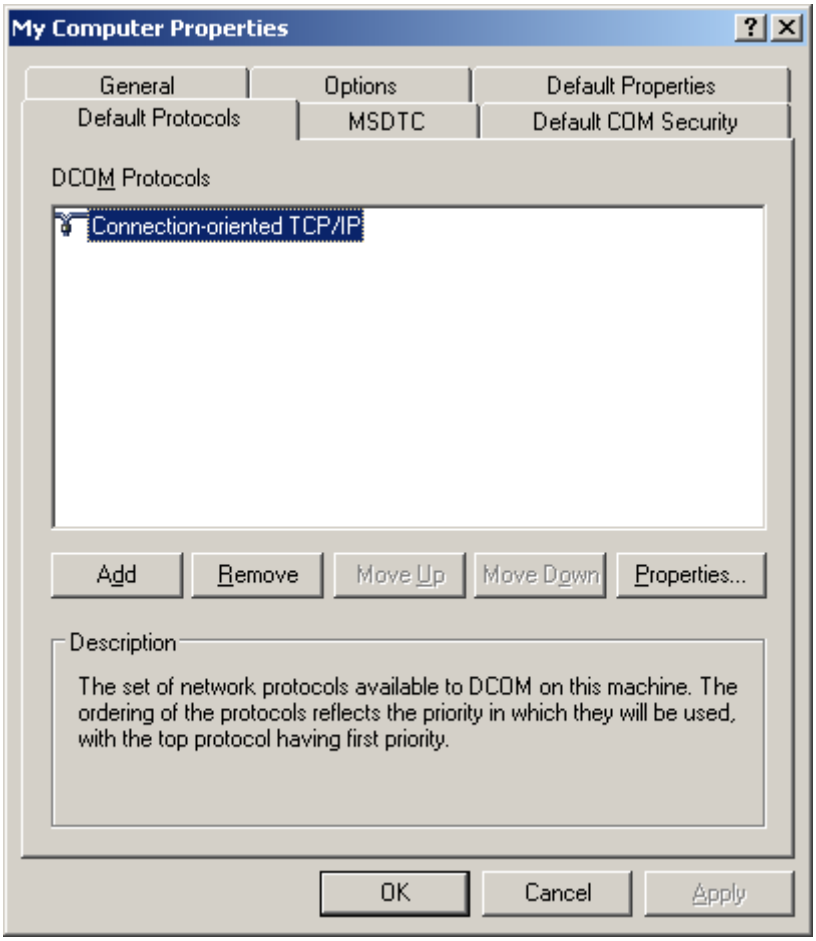


Fig. 1: Dcom_general

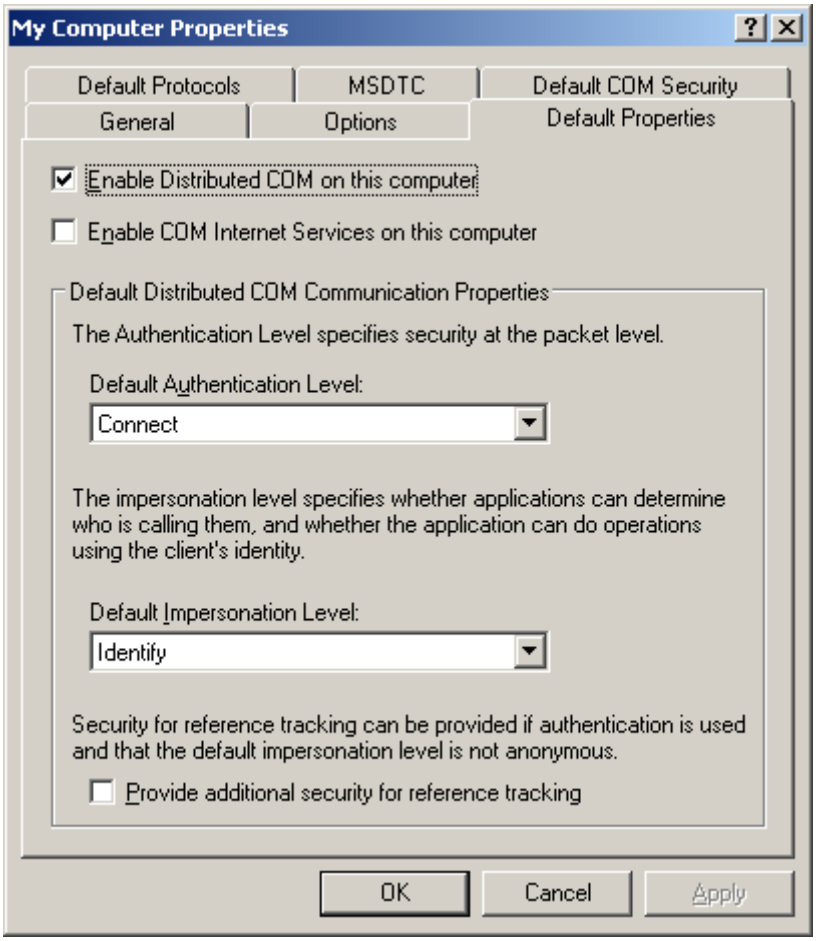
- On the "Options" tab no changes have to be made. The default settings will be correct for OPC Client side security settings.



- 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 the 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 Authentification 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
  - Authentification Level = None
  - Impersonation Level = Anonymous
- If the client machine runs in a **Domain** the level remains to default settings
  - Authentification 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).
  - Authentification 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 disapears 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 temporarely 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 setings 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

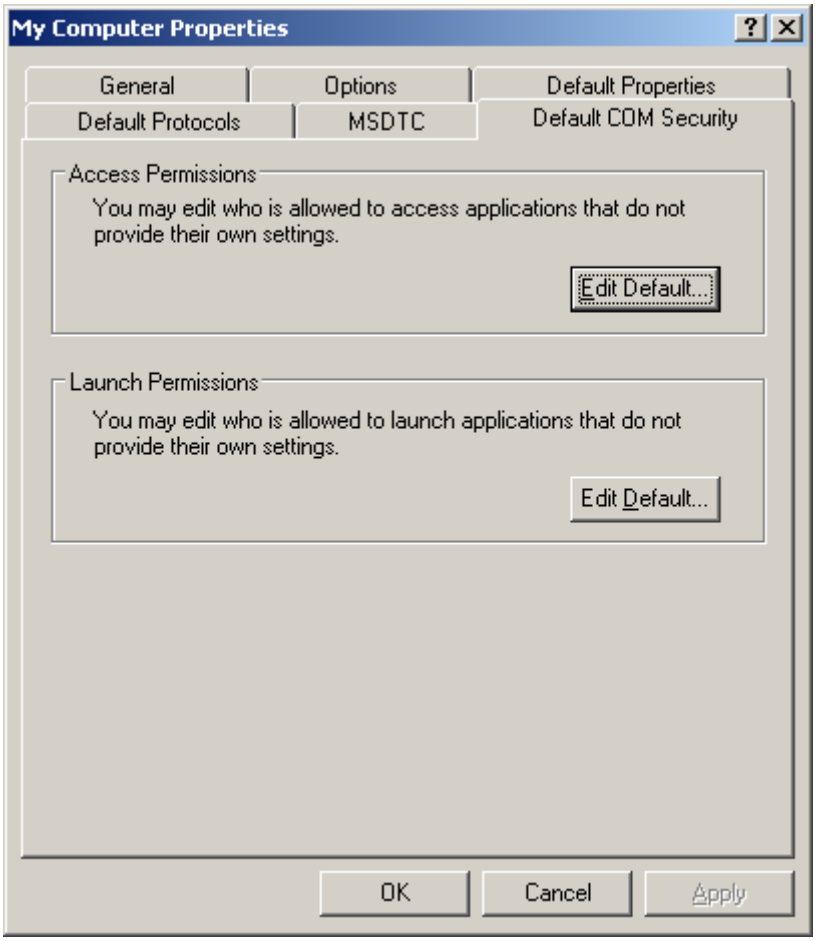
Interactive User

System

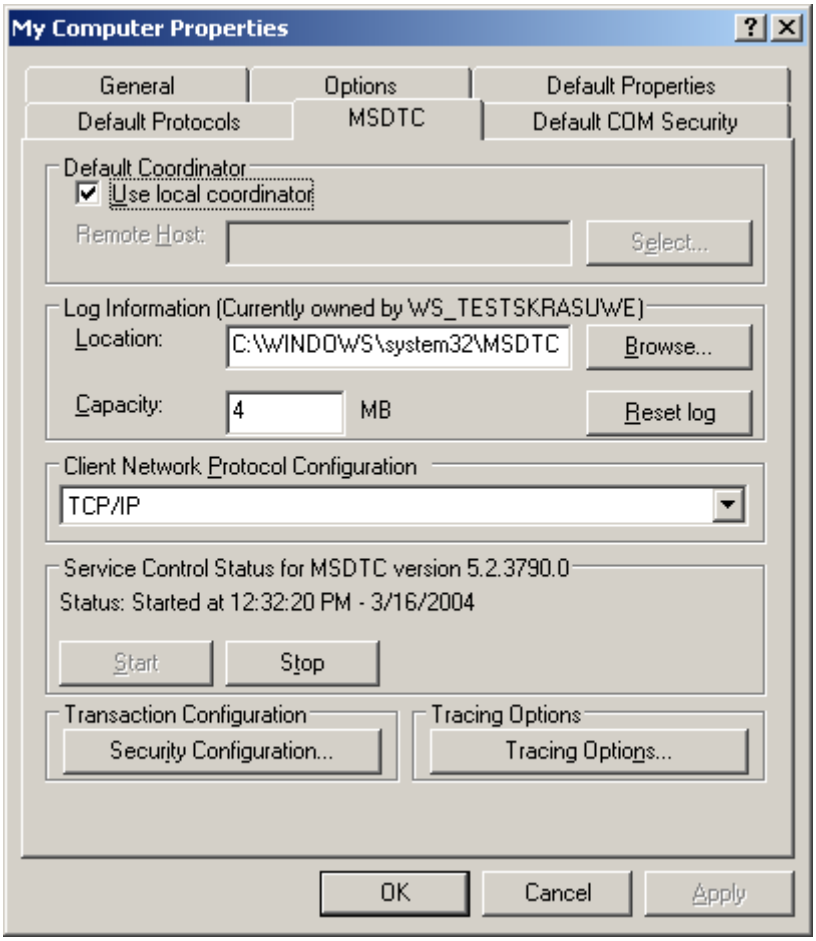
Network

"OPC Server's Security Context"

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



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



5.2.4 Server

5.2.4.1 Server Machine

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.

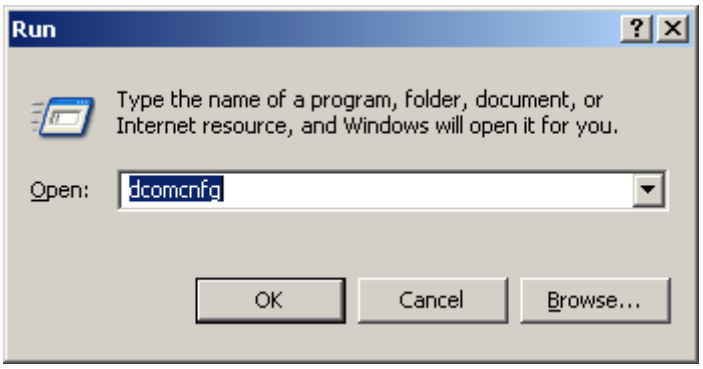
Please refer to our Article about Operating Systems [▶ 51] for more information.

Step 2: DCOM configuration

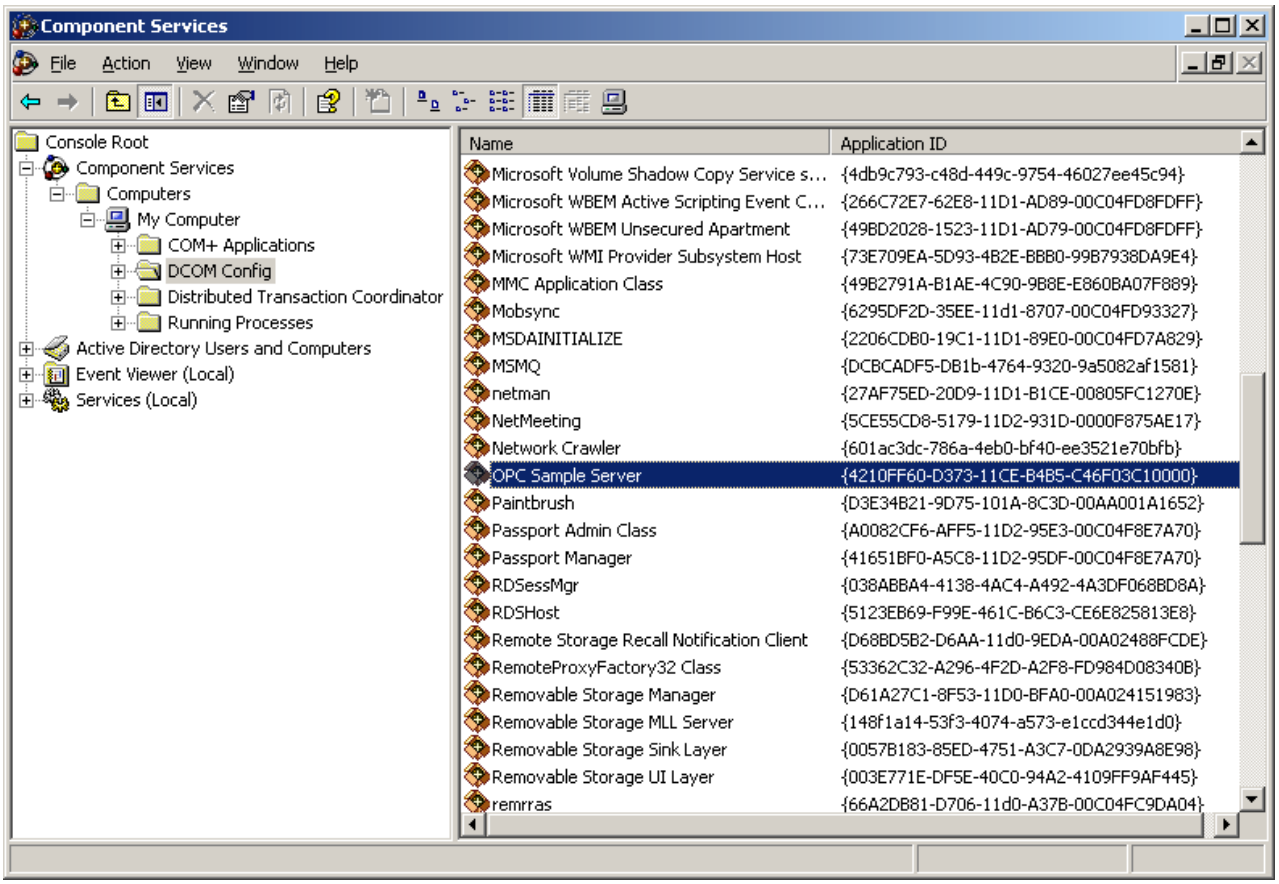
To configure DCOM for a remote OPC communication, please perform the following steps on the computer running TwinCAT OPC-Server.

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

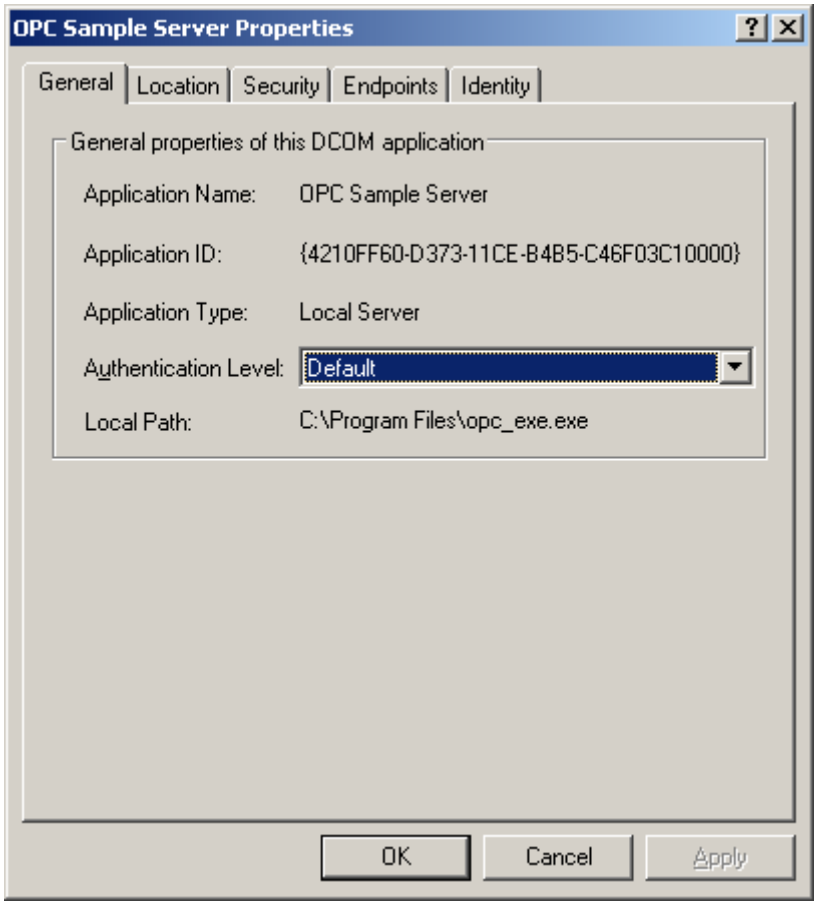
- Open *Start --> Run --> dcomcnfg.exe* to start the DCOM configuration dialog.



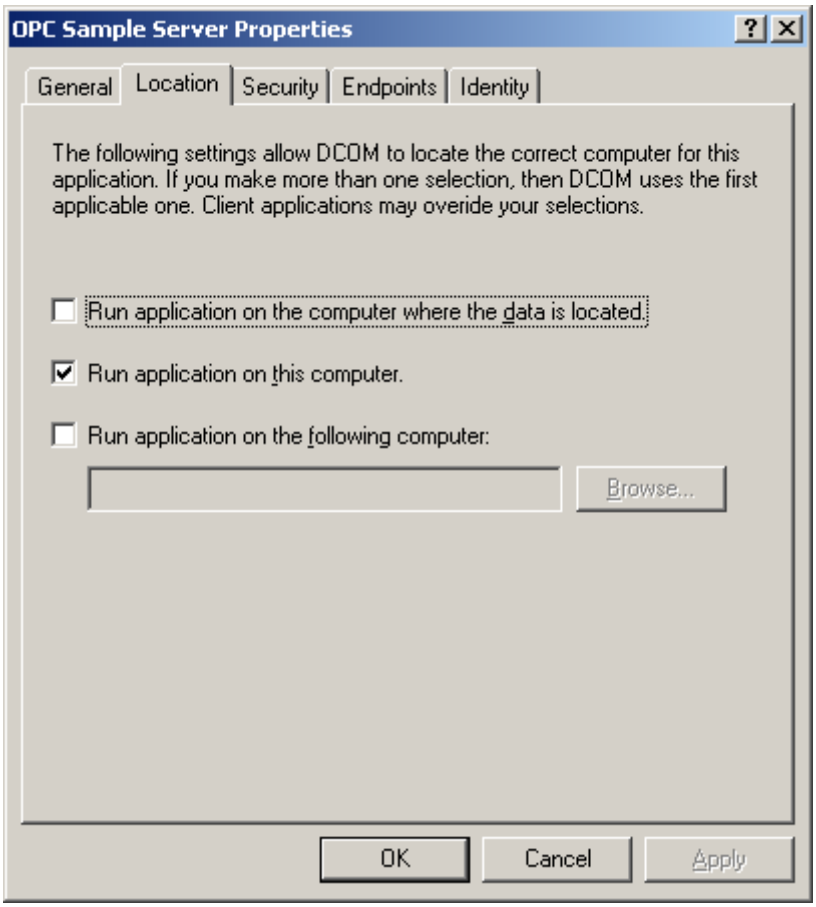
- Navigate to *Console Root --> Component Services --> Computers --> My Computer --> DCOM Config* to display all DCOM server applications.
- Select the TwinCAT OPC-Server (or one of its Clones), right click it and select **Properties** to change the DCOM security for this specific DCOM Server only.



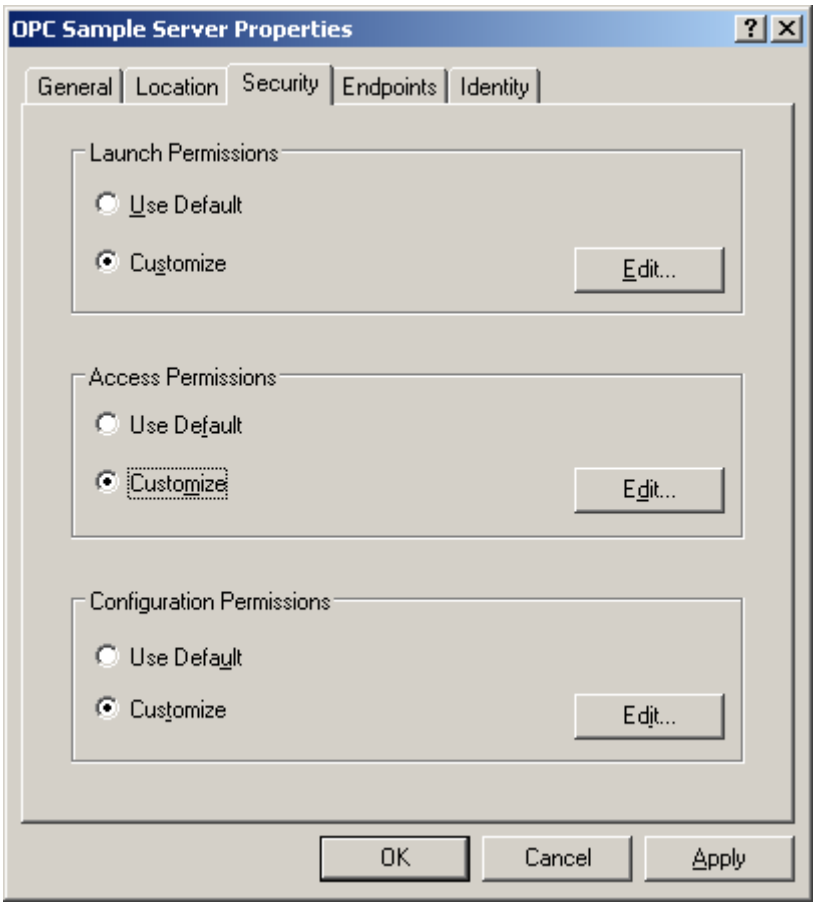
- On the "General" tab no changes have to be made. The default settings will be correct for OPC Server's security settings. The Authentication Level set to *Default* will overtake the settings from the Default Properties Tab valid for all COM Objects on this machine (*Connect* by default).



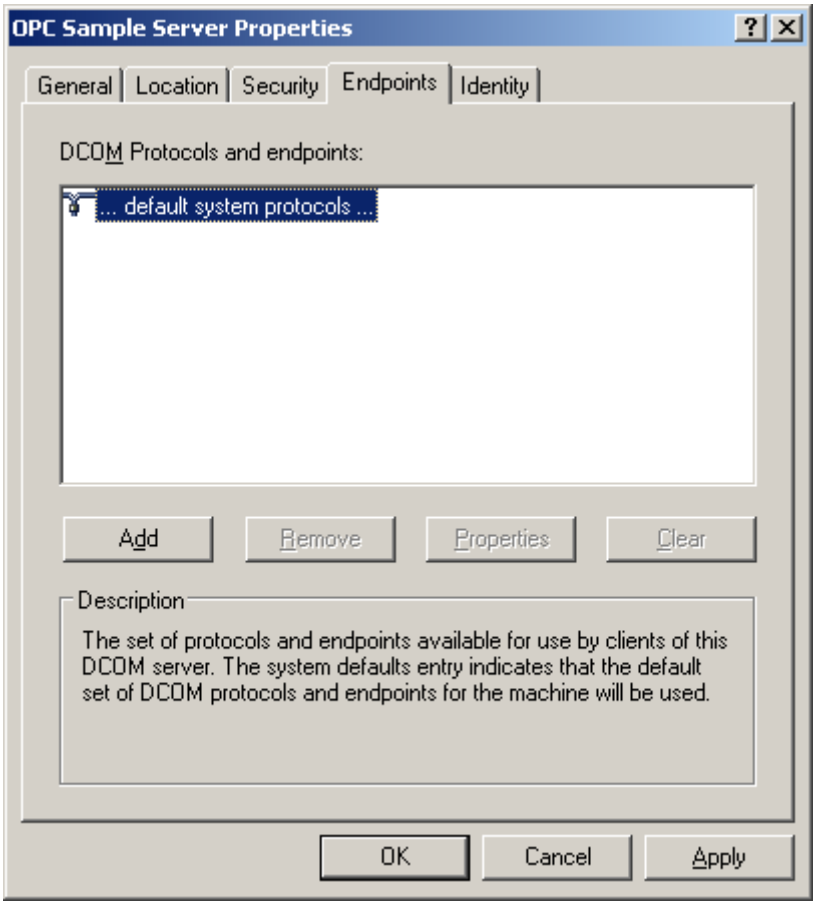
- On the "Location" tab no changes have to be made. The default settings will be correct for the OPC Server's security settings.



- 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.



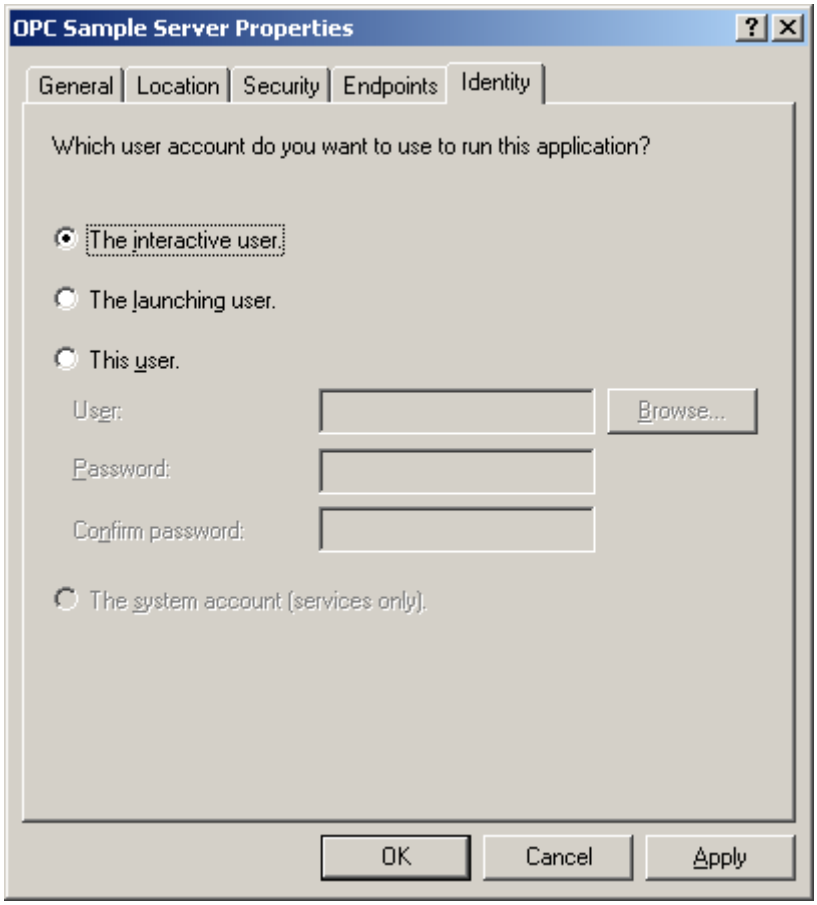
- On the "Endpoints" tab the default settings should remain. In the Default Protocols Tab for all COM-Objects *Connection-oriented TCP/IP* protocol should be moved to the top position.



- On the "Identity" tab no changes have to 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 actually 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 having always 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.



5.2.5 DCOM Permissions

5.2.5.1 Permissions

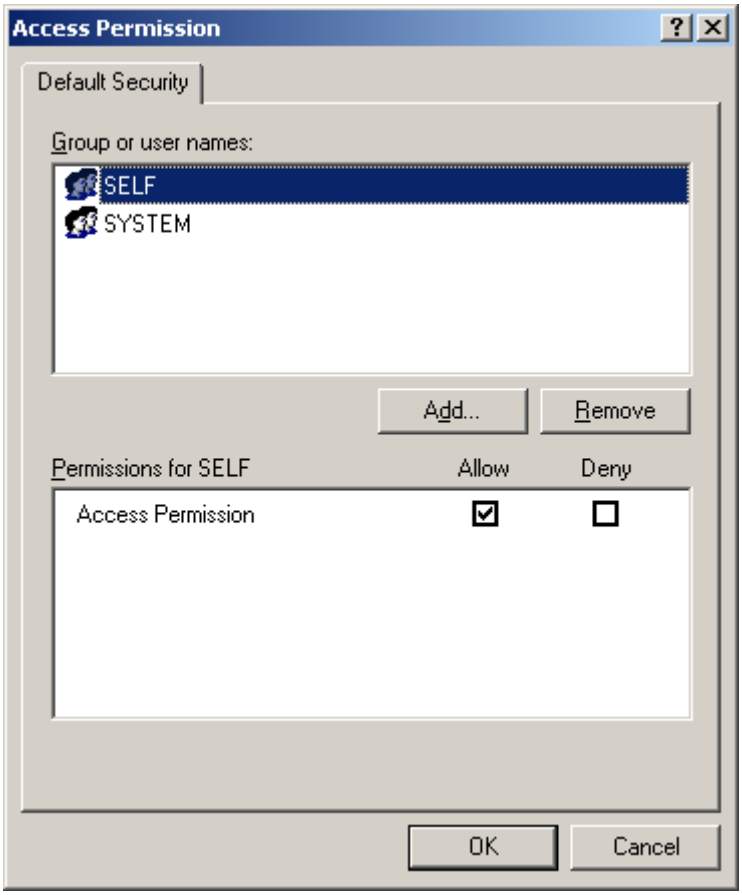
Within DCOM Security the launch permission is described as the right to start (launch) an application. As an OPC Server will be started (launched) by the Client application (CreateInstance) the launch permission for the OPC Server application must be granted for the *Security Context* in which the OPC Client is running. The *Security Context* of the Client may be different depending on the uuser actually logged in and double clicked the Client.

The access permission is described as the right to access (communicate to) an application. As an OPC Client will call functions on the OPC Server Object the *Security Context* of the Client must be known by the Server. When sending callbacks to the Client (asynchronous functions or DataChange) the Client must grant access permission to the OPC Server.

- Access Permission [▶ 66]
- Launch Permission [▶ 67]
- Select Users [▶ 68]

5.2.5.2 Access Permission

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

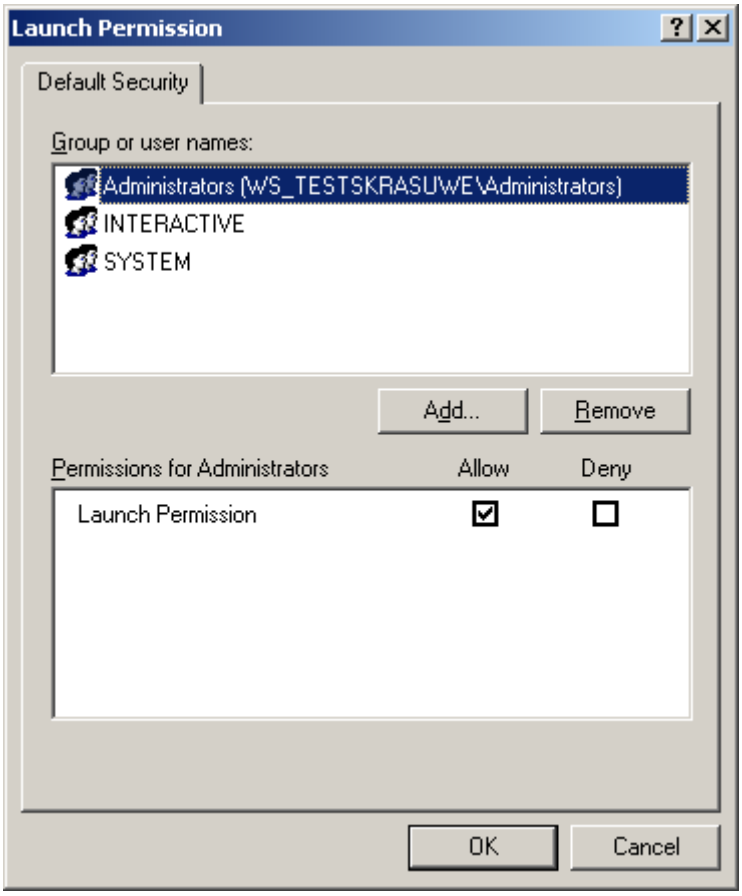


For selecting users click the *Add..* button:

Selecting Users [▶ 68]

5.2.5.3 Launch Permission

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

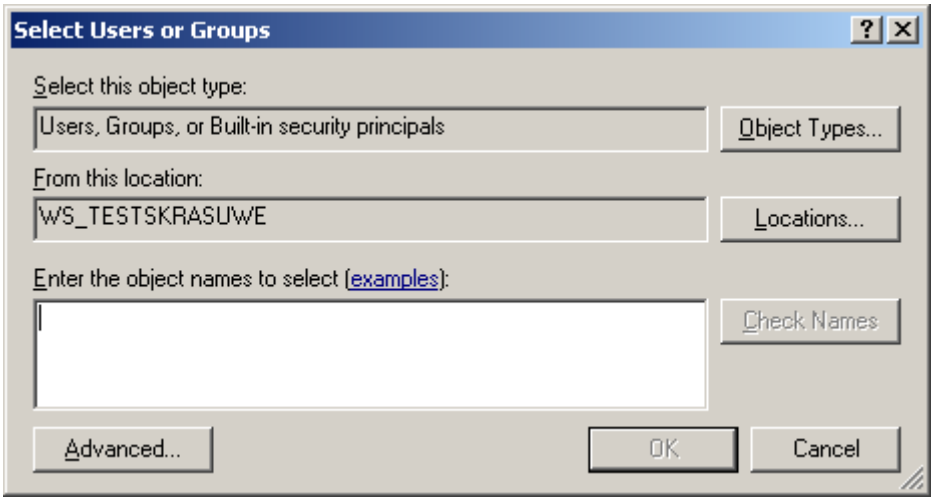


For selecting users click the *Add..* button:

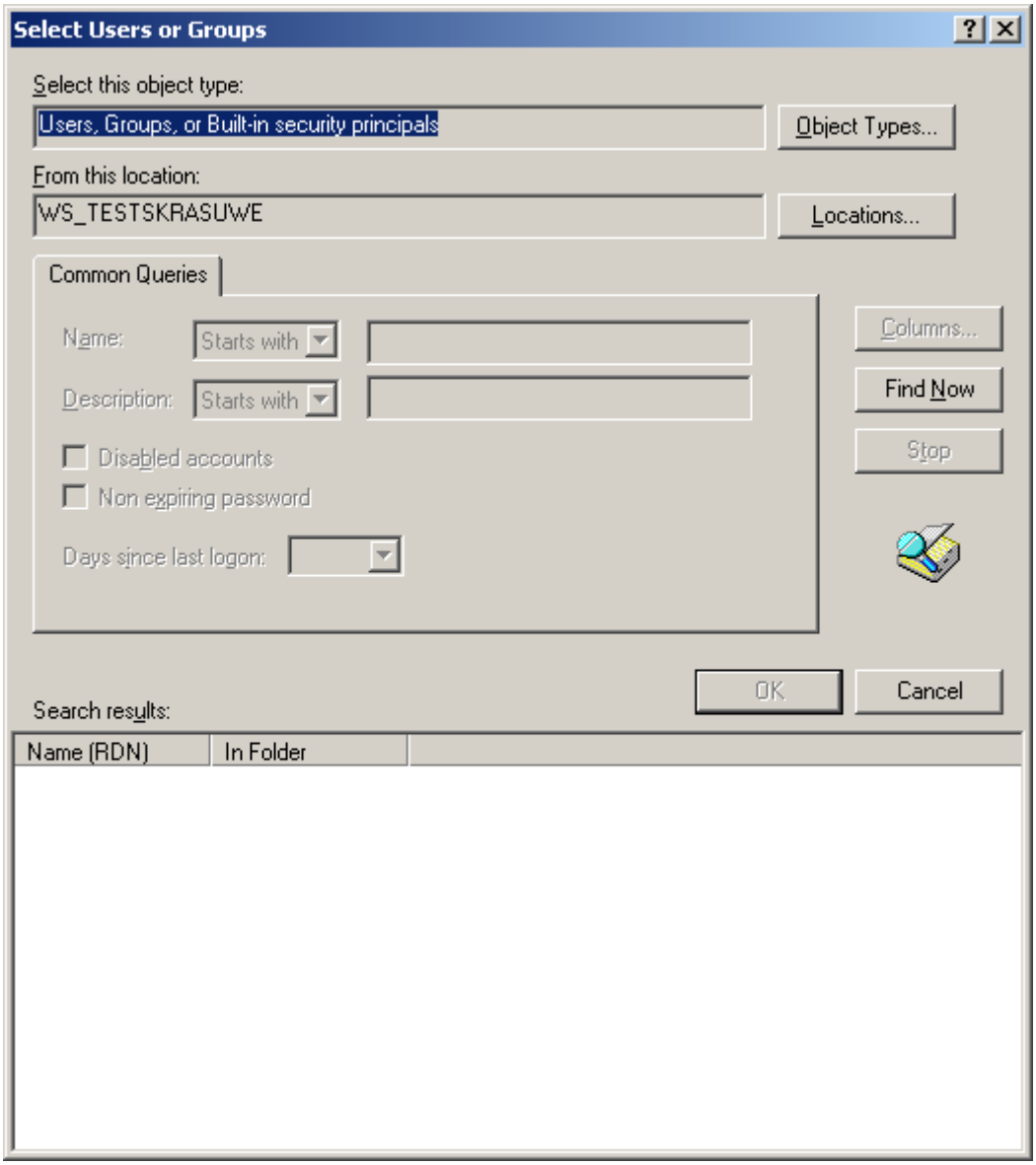
Selecting Users [▶ 68]

5.2.5.4 Select User

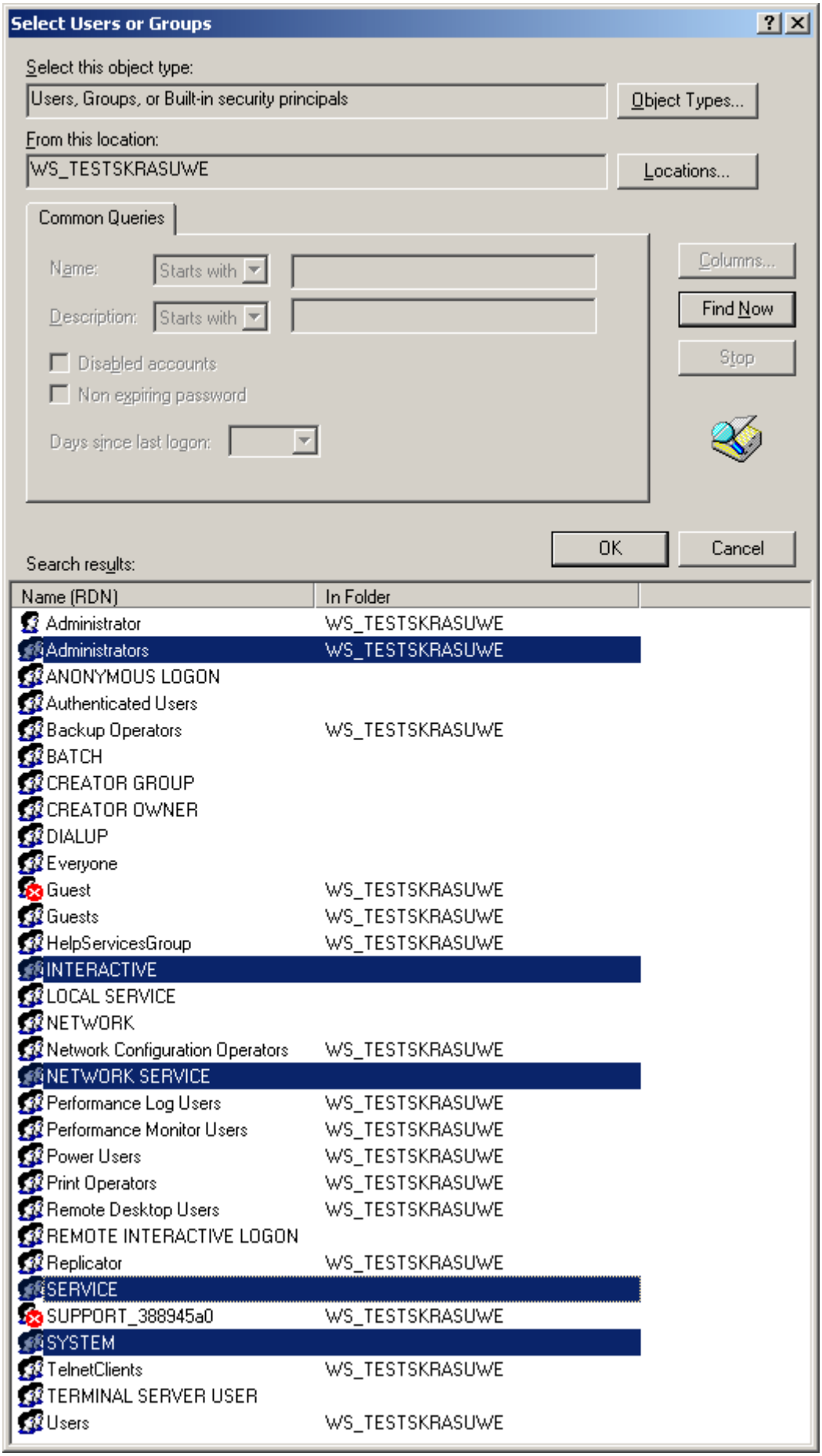
A filter on object types can be set to reduce the listing of all user names. The location states the machine name where the user account is stored, this can be the local workstation or the central domain controller machine. The edit box below the names can be typed in and verified afterwards by pressing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


Select Users or Groups
Select this object type:
Users, Groups, or Built-in security principals
Object Types...
From this location:
WS_TESTSKRASUWE
Locations...
Common Queries
Name:
Starts with
Description:
Starts with
Disabled accounts
Non expiring password
Days since last logon:
Columns...
Find Now
Stop
OK
Cancel
Search results:
Name (RDN)
In Folder
Administrator WS_TESTSKRASUWE
Administrators WS_TESTSKRASUWE
ANONYMOUS LOGON
Authenticated Users
Backup Operators WS_TESTSKRASUWE
BATCH
CREATOR GROUP
CREATOR OWNER
DIALUP
Everyone
Guest WS_TESTSKRASUWE
Guests WS_TESTSKRASUWE
HelpServicesGroup WS_TESTSKRASUWE
INTERACTIVE
LOCAL SERVICE
NETWORK
Network Configuration Operators WS_TESTSKRASUWE
NETWORK SERVICE
Performance Log Users WS_TESTSKRASUWE
Performance Monitor Users WS_TESTSKRASUWE
Power Users WS_TESTSKRASUWE
Print Operators WS_TESTSKRASUWE
Remote Desktop Users WS_TESTSKRASUWE
REMOTE INTERACTIVE LOGON
Replicator WS_TESTSKRASUWE
SERVICE
SUPPORT_388945a0 WS_TESTSKRASUWE
SYSTEM
TelnetClients WS_TESTSKRASUWE
TERMINAL SERVER USER
Users WS_TESTSKRASUWE

More Information:
www.beckhoff.com

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com