Table of Contents

1 Introduction

OPCAdapter

system for connecting of a control and monitoring systems through network and language platforms, or through Internet.

What is it OPCAdapter?

OPCAdapter is application dedicated for **processing** and perhaps **even vizualization** process data obtained from equipments of various contractors. Programable logic controller (PLC), various inteligent sensors and actuating units, which have communication interfaces supported bz OPC technology, can be mention as examples of these equipments (equipments are implemented as OPC servers).

OPC technology presents very **expanded industry standard** what is great **advantage** of its use. Per contra, the **disadvantage** is the principle of this technology. It is based on the COM and DCOM principles, and therefore is delimited **almost for Microsoft platform**. **OPCAdapter removes the disadvantage of this method**. The data received from/sent to OPC servers are transfer between various applications with same communication interface, with communication protocol TCP/IP which doesn't use the COM and DCOM technology. This communication could be in progress between running application on the same computer, as well as two computers in different network domains or even through Internet.

As example of such application, we could mention the Java application (which doesn't use COM technology) running on the same computer as OPCAdapter, which have to handle the data obtain from OPC serevrs and store it to datadase. This application represents the client, and the OPCAdapter performs as a server for this application, while for connected OPC servers performs as OPC client. OPCAdapter and mentioned application communicate through sockets.

How can be OPCAdapter used?

- As self-contained application for collecting and visualllisation data from a technological process, there is posssibility to select which data will be displayed graphical and which only using charakters
- As OPC client for testing yor OPC servers with possibility to browse on local and/or remote servers too. It allows storing this browsed configuration to the XML file
- As gateway between your application and OPC servers implemented according to the specification OPC Data Access 2.0x

Which components has the OPCAdapter software package?

- OPCAdapter - NT service 37 is runtime pplication without a user interface
- The application with the user interface 20 is applicable to create and tune configuration of connected devices, or for simple monitoring of transferred data. It can provide the full OPCAdapter functionality, or it can be used only as configuration tool for OPCAdapter - NT service.
- The application OPCAdapterSocketClient 41 is an example of the client of OPCAdater, which simply presents data transferred from OPCAdapter through special communication interface TCP/IP layer. It is implemented in development environment Microsoft Visual C++ 7.7. an it is supplied with full source files.
- The help file that includes detailed description of the communication protocol and manual for creating user client applications, which are able to communicate with OPCAdapter.

Which other useful properties has the OPCAdapter?

- It is able to connect/disconnect everz OPC server extra or together
- It happens often in control and monitoring of technological process that it is necessary to power down some equipment and consequentlz its OPC server is disconnected too. OPCAdapter has ability to set the period for automatically trying to connect disconnected OPC server again.
- The configuration, which is stored in XML file, gives the possibility to use it in the user client application of OPC servers address space.

What can SAE - Automation, s.r.o. (company limited) Nová Dubnica offer in connection with the OPCAdapter?

1. Deliveries of OPCAdapter software package.
2. Implementing of complete control and monitoring systems using application OPCAdapter.
3. Development of the user client applications.
4. Extension of OPCAdapter with data collecting from different devices not using communication drivers implemented as OPC servers (as example connection of control modules of company AMIT s.r.o., limited).
5. Development communication drivers for the OPCAdapter server side by needs of zour application - it means, that we are implementing the communication between your application nad OPCAdapter.

Reference to topic:

What is it OPCAdapter? 4

1.1 What is it OPCAdapter?

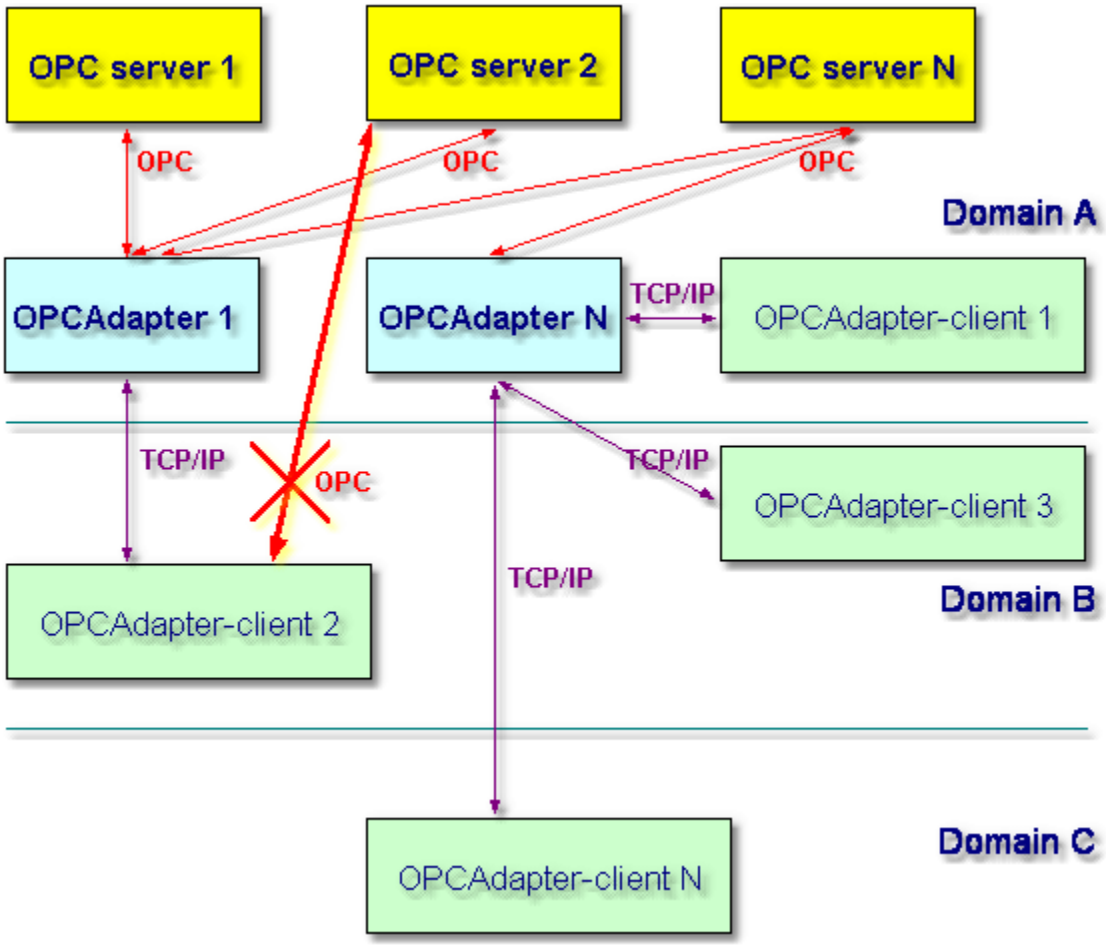
OPCAdapter is pplication which join funcionality of OPC client and TCP/IP server. It enables connection of different controll and monitoring systems through network domains 4, platforms 5, or through Internet 6. As example we could mentoion application in Java, which have to handle data from OPC server and have to store their to database.

Refernce to topics:

Connection of control and monitoring systems between network domains 4
Connection of control and monitoring systems between different language platforms 5
Connection of control and monitoring systems in Intenet 6
Connection of control and monitoring systems in network with more users 7

1.1.1 Connection of control and monitoring systems between network domains

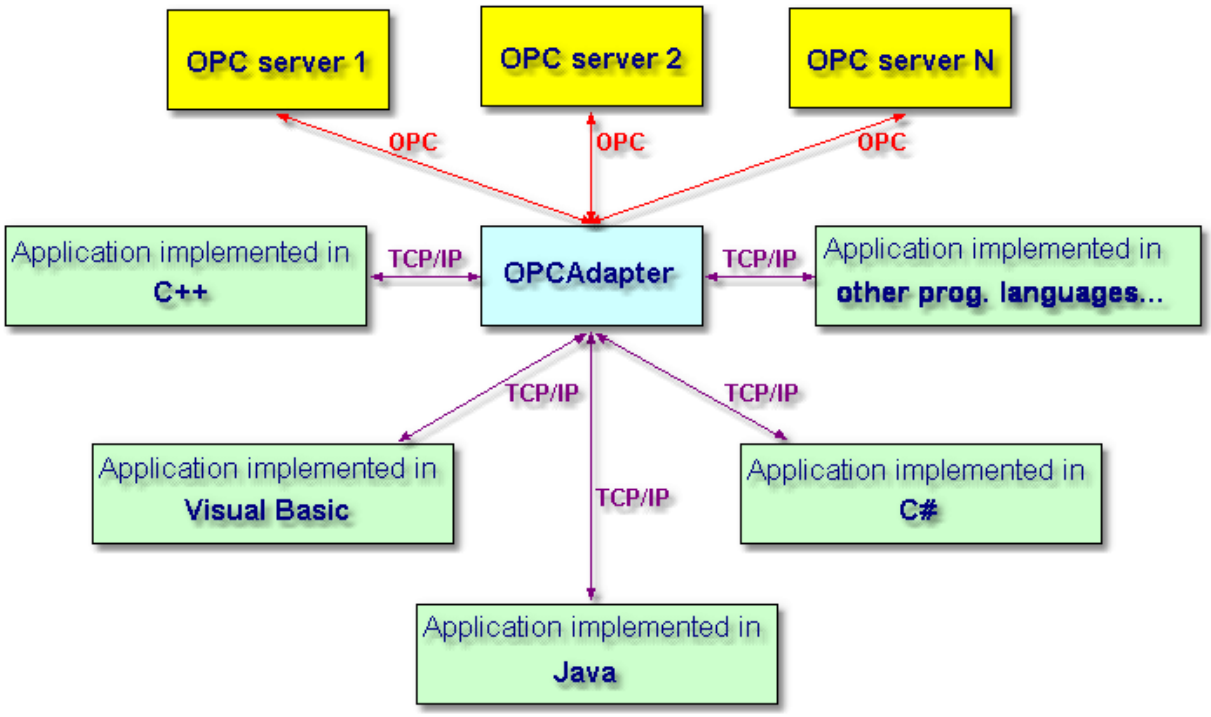
In the following picture shown communication model describes application layout in computer network. This is example of computer network with more domains (domains A, B a C). This picture shows tzpical example, when it is usefull to access from one domain to OPC server in the second domain. In this case we are faced with the serious problems connected with configuration of DCOM 12. With help of application **OPCAdapter**, we are spare from this problems, or we could minimalise them.



Obr. 1.: Connection of control and monitoring systems between network domains

1.1.2 Connection of control and monitoring systems between different language platforms

Access to **OPCAdapter** is language independent because it is realised through TCP/IP 13.



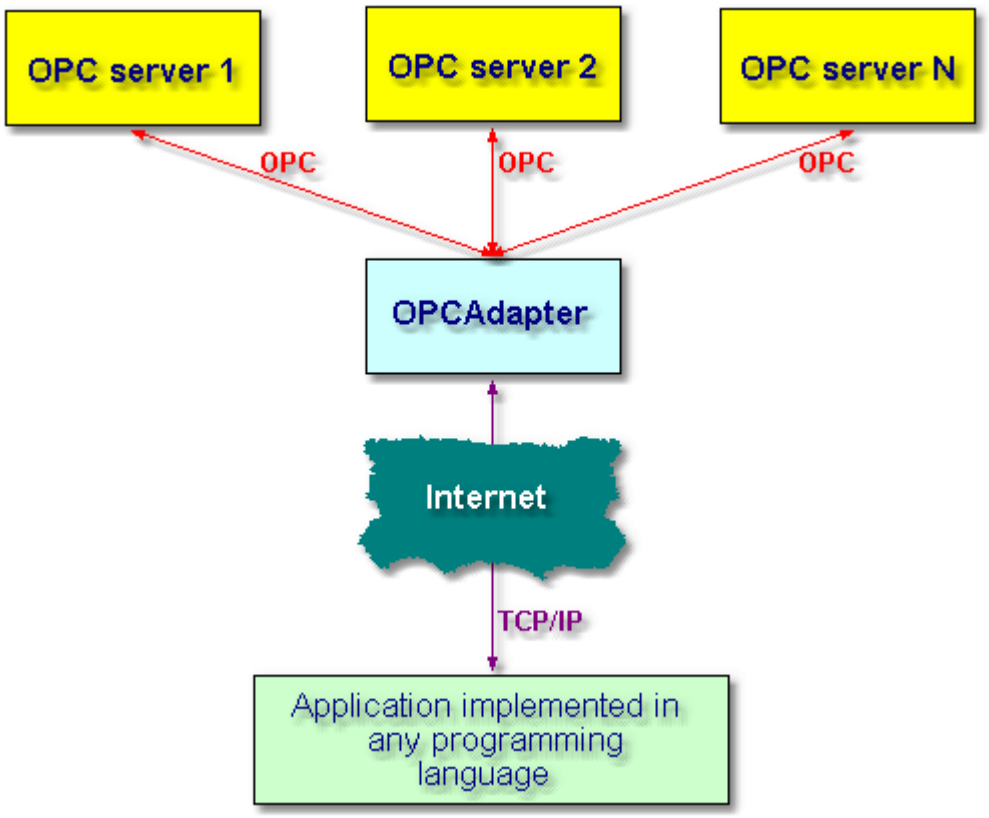
Obr. 2.: Connection of control and monitorinf systems between different language platforms

Reference to topics:

TCP/IP socket server 11
Communication TCP/IP client-server without COM, DCOM technologies 13

1.1.3 Connection of control and monitoring systems in Internet

Access to **OPCAdapter** is realised through TCP/IP 13, what makes possible to access on client application from any computer network whitin the frame of Internet.



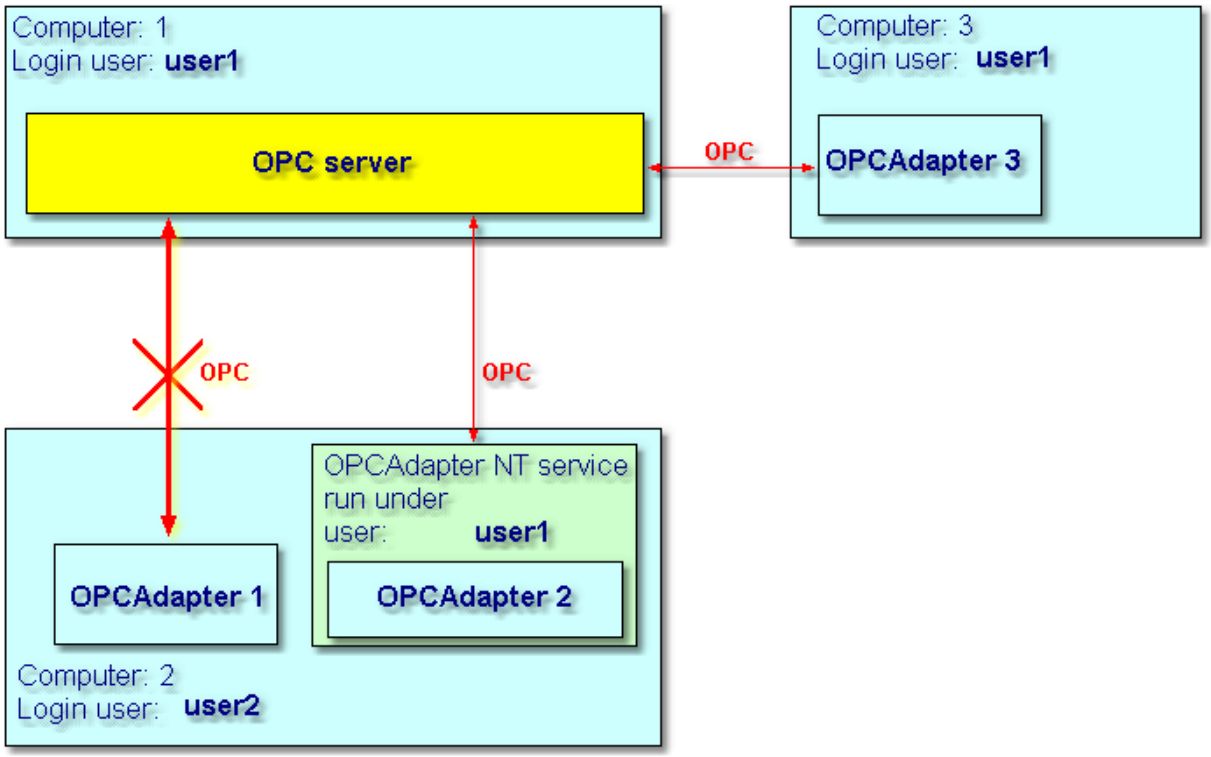
Obr. 3.: Connection of conrol and monitoring systems in Internet.

Reference to topics

TCP/IP socket server 11
Communication TCP/IP client-server without COM, DCOM technologies 13

1.1.4 Connection of control and monitoring systems in network with more users

OPCAdapter bring solution also for computer networks with more users. Access to OPC server running on different computer with different user is possible with OPCAdapter NT service 37.



Obr. 4.: Connection of control and mońitoring systems in network with more users.

Reference to topic:
OPCAdapter NT service 37

1.2 Good reason why use OPCAdapter

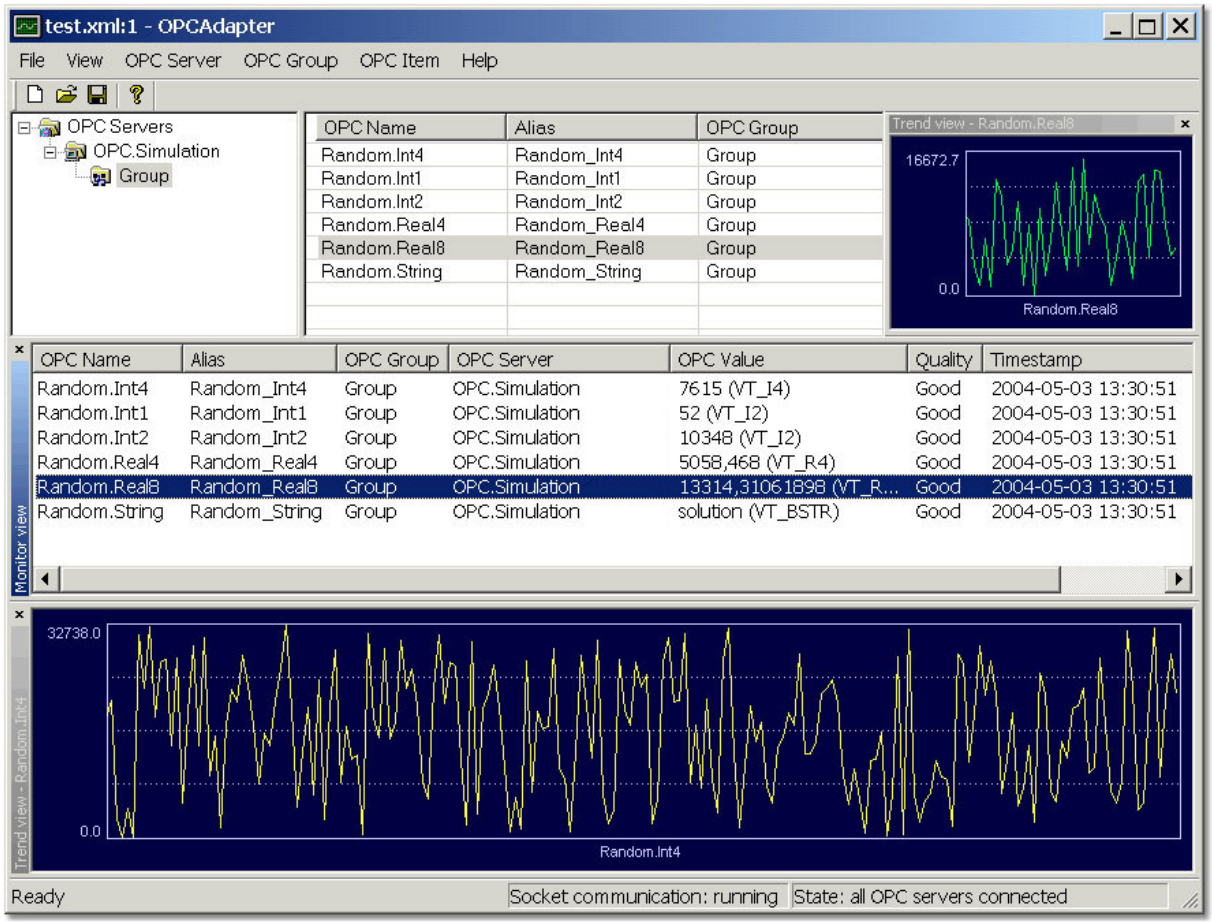
Quality concentration on important things
OPCAdapter is application, which join functionalitz of OPC client and TCP/IP server. It enables connection of different controll and monitoring systems through network domains 4, platforms 5, or through Internet 6.

System confuguration is very simple and more time applicable
System is configured with simple, but comfort configurator. It is possible to save alone configuration in file Extensible Markup Language (XML) 18. Therefore it is easy to modify or use again same configuration.

2 OPCAdapter

OPCAdapter is pplication which join funcionality of OPC client and TCP/IP server. It enables

connection of different controll and monitoring systems through network domains 4, platforms 5, or through Internet 6. As example we could mentoion application in Java, which have to handle data from OPC server and have to store their to database.



Obr. 5.: OPCAdapter with user interface, configurator for OPCAdapter NT service.

Refernce to topics:

Connection of control and monitoring systems between network domains 4
Connection of control and monitoring systems between different language platforms 5
Connection of control and monitoring systems in Intenet 6
Connection of control and monitoring systems in network with more users 7

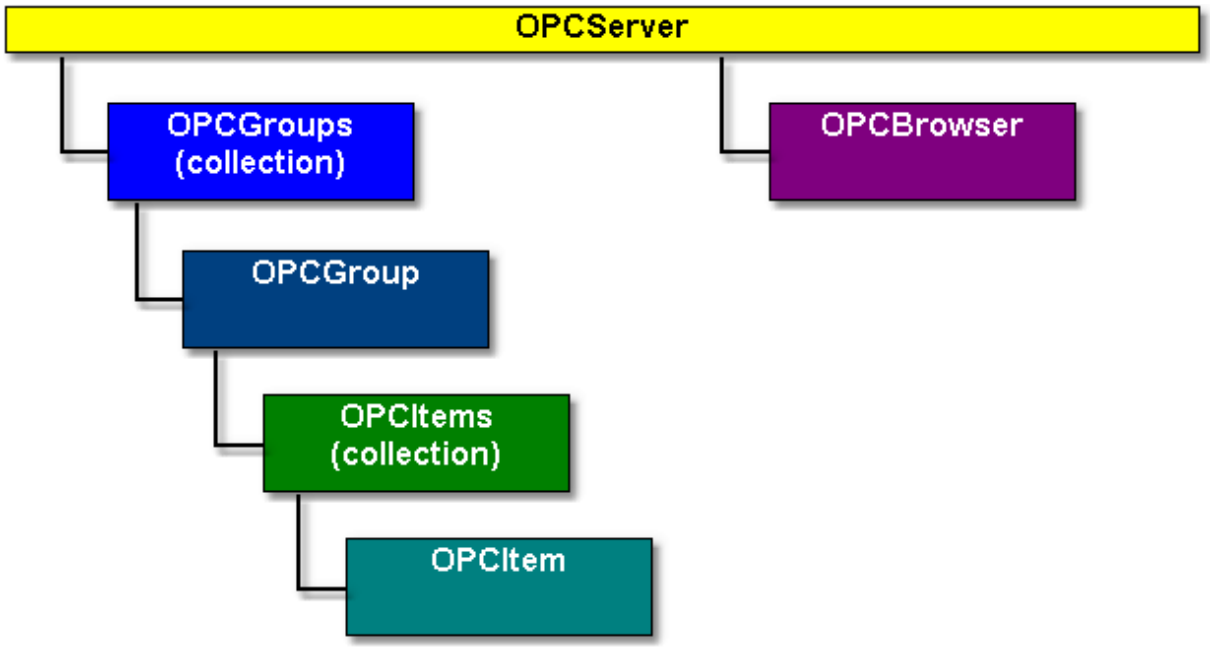
OPCAdapter with user interface 20
OPCAdapter NT service 37

2.1 OPC client compatible with OPC Data Access 2.0

OPCAdapter is program which colud assecc as OPC DA client, to various OPC Data Access Servery 11. Intefface of these servers are accesible to application **OPCAdapter** through **Automation Interface** (interface).

Automation Interface

OPCAdapter uses functionality of free accesible library DLL, for accessing OPC DA servers on **local** or **remote** computer. Automation DLL is free distributed by organization OPC Foundation. Following picture shows object modelm, which is accesible through automation DLL.



Obr. 6.: Object model accesible through automation DLL.

Reference to topics:
OPC specification 10
OPC Data Access 2.0 11

2.1.1 OPC specification

OPC (OLE for Process Control) is standard mechanizm for communication with several data sources. It is open and effective communication architecture based on data access.

Principle of OPC standard is based on technologies **OLE/COM (DCOM).** These technologies are made for data exchange between Microsoft applications.

OPC standards are free available technical specifications, which define set of standard interfaces for different application in automatization technology. These interfaces increase power and effectivity of data exchange between software components from different producers.

The most used OPC specifications

OPC Data Access	Defines interface for read and write data in real time.
OPC Alarms and Events	Defines interface for monitoring events.
OPC Historical Data Access	Defines interface for access to historical data.
OPC Batch	Defines interface for access to data, which are used for batch handle.
OPC Security	Defines interface for setup and ulitize securitz levels.
OPC and XML	Integration of OPC and XML for internwet applications.
OPC Data eXchange (DX)	Defines communication between server and client.

OPC as standard was established by organization OPC Foundation.

2.1.2 OPC Data Access 2.0

OPC Data Access specification
It defines interface for access to process data between client and server application. **Data Access Server** (OPC DA server) provide to one, or to several **Data Access Client** (OPC DA client) full transparent access to any data source. Of course, it is possible, that one OPC DA client could in same time access to more OPC DA servers.

Required interfaces
Each OPC DA server, which meets specification of OPC Data Access 2.0 have to have implemented following required interfaces:

OPCServer

IUnknown	required
IOPCServer	required
IOPCCommon	required
IConnectionPointContainer	required
IOPCItemProperties	required
IOPCServerPublicGroups	optional
IOPCBrowseServerAddressSpace	optional

OPCGroup

IUnknown	required
IOPCItemMgt	required
IOPCGroupStateMgt	required
IOPCPublicGroupStateMgt	optional
IOPCSyncIO	required
IOPCAsyncIO2	required
IConnectionPointContainer	required

2.2 TCP/IP socket server

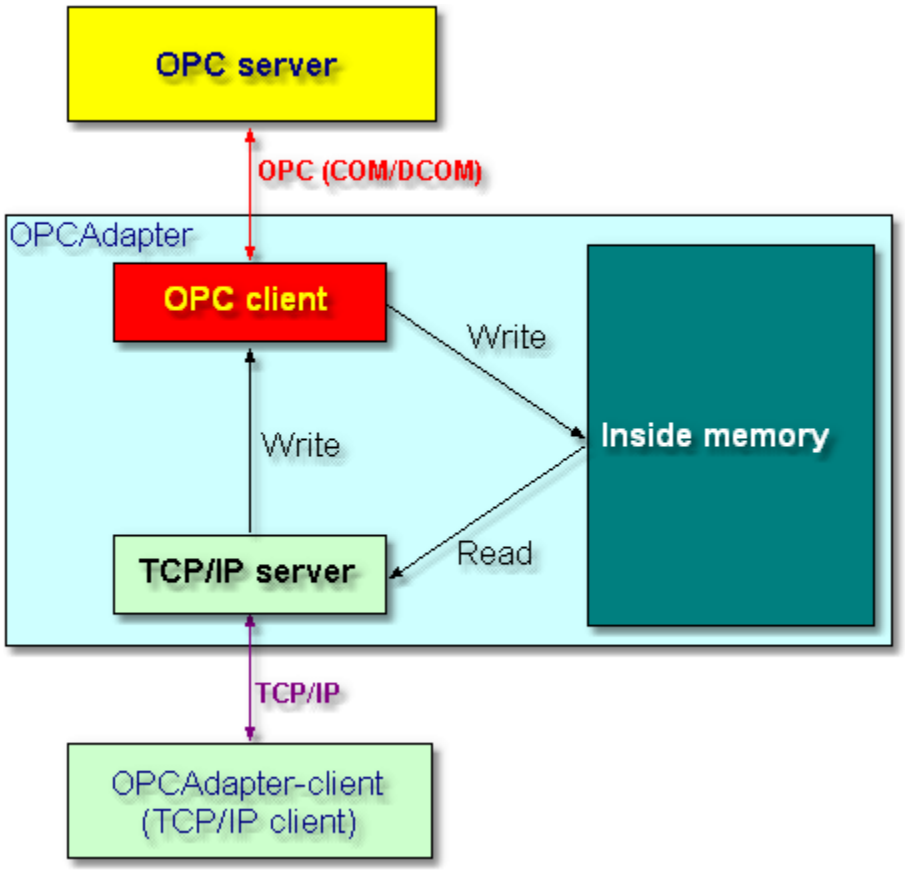
Its task is listen on specific **port** for client connections. When the client is join, server returns data and sends back to client and closes the connection.

Port
Port is defined by uniform location, to which application **OPCAdapter** can send data and receive messages form there.

Reference to topic:
Communication TCP/IP client-server without COM, DCOM technologies 13

2.3 Communication through interfaces OPC, COM, DCOM, TCP/IP

Simple communication scheme
On the one hand, it specify applied communication standard between
OPC server and OPCAdapter 12 and on the other hand communication standard between
OPCAdapter and OPCAdapter-client 13.



Obr. 7.: Simple communication scheme.

Functionality in communication scheme

OPCServer

Any OPC DA server, which is full compatible with standard OPC Data Access 2.0 11.

OPCAdapter

Its functionality we could spread on two basic modules:

- OPC DA client 9,
- TCP/IP server 11.

In its **inside memory** it stores actual values of all monitored OPC variables.

OPCAdapter-client

The application, which communikates through TCP/IP with program **OPCAdapter**.

Reference to topics:

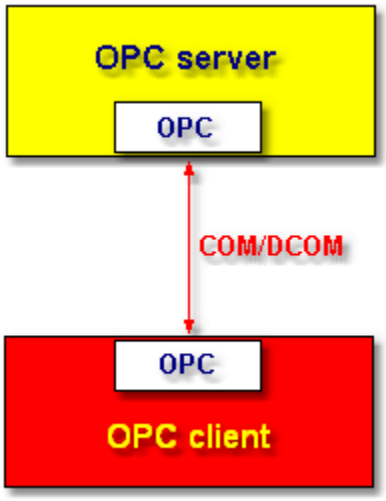
Communication OPC client-server with COM, DCOM technologies 12
Communication TCP/IP client-server without COM, DCOM technologies 13

2.3.1 Communication OPC client-server with COM, DCOM technologies

In generali now, **Ole for Process Control (OPC)** is accepting as one of the most popular

industry standard between users and developers too.

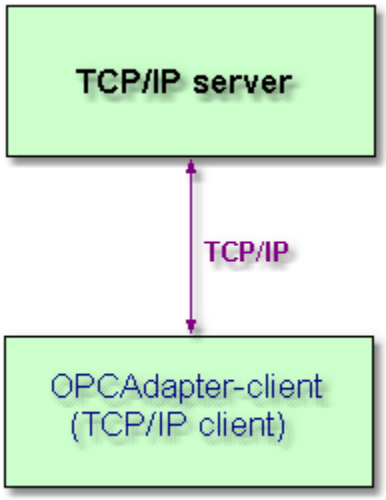
OPC is standard interface, for access to application from automation region on Windows platform. On the present is this standard based on the **Distributed Component Object Model (DCOM)**, what is technology by company Microsoft for implementing distributed systems.



Obr. 8.: Communication OPC client-server with COM, DCOM technologies.

2.3.2 Communication TCP/IP client-server without COM, DCOM technologies

Transmission Control Protocol/Internet Protpcol (TCP/IP) is standardized industry communication protocol, which defines methods for union data to packets for transfer between equipments in heterogonous network. It is standard for data exchange between several networks, including Internet.



Obr. 9.: Communication TCP/IP client-server without COM, DCOM technologies.

Reference to topic:

TCP/IP socket server 11

2.4 Special communication protocol

Bilateral communication between application**OPCAdapter** and other applications, which could be implemented in any programming language (example.: Java, C++, Delphi...), is realized by protocol TCP/IP 13.

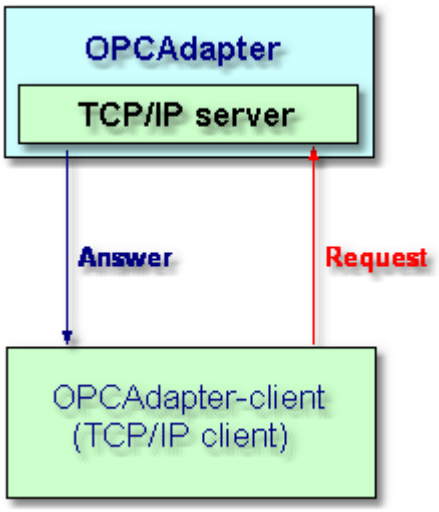
Special protocol, developed for application **OPCAdapter**, which in details shows implemented operations (methods) make application layer of protocol TCP/IP:

- Operation read 14,
- Operation write 16.

Implemented operations are always execute in two steps:

1. request for executing operation,
2. answer on request.

Following picture shows simple scheme of operation executing:



Obr. 10.: Simple scheme of operation executing.

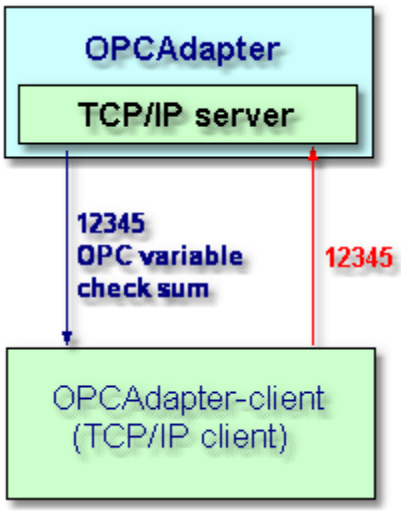
Reference topics:

Operation read 14
Operation write 16
Supported data types of OPC variables 17

2.4.1 Operation read

OPCAdapter after receiving request for **operation read**, return to client application cpmplete list of all OPC variables, which are stored in its inside memory 11.

Following pisture shows simple scheme of executing of **operation read**:



Obr. 11.: Simple scheme of operation read executing.

Protocol - operation READ

Request (any clint application)

1,2,3,4,5	header of request (5 bytes)

Answer (OPCAdapter)

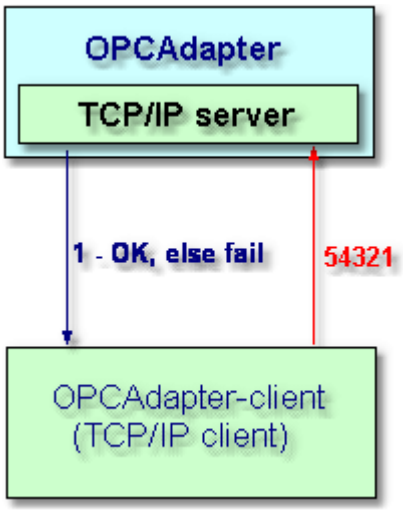
1,2,3,4,5	header of answer (5 bytes)
parita2	number of bytes including parita1 (4 bytes)
250	begin of variable frame (1 byte)
name of variable	ASCII characters of the variable name, or AliasName
251	end of variable name (1 byte)
timestamp	time stamp of last variable actualization, date+time (8 bytes)
252	end of time stamp (1 byte)
quality	quality of read variable (1 byte)
253	end of quality (1 byte)
code type	code type (1 byte): 1 - integer 2 bytes 2 - long 4 bytes 3 - real 4 bytes 4 - double 8 bytes 5 - string 6 - boolean (0 - false, FF - true) 7 - date 8 bytes 8 - unsigned integer - 2 bytes 9 - unsigned integer - 1 byte
254	end of code type (1 byte)
value	value of variables - number of bytes according to code types (if code type is 5, then end of string have to be check on ASCII char 250 or 255)
250 or 255	begin of new variable, or parita1 (1 byte): 250 - then will be another variable 255 - then will be parita1
parita1	number of variables (2 bytes, lower byte is first - LH)

Reference to topics:

2.4.2 Operation write

OPCAdapter after receiving request for **operation write** of new value for one defined OPC variable, returns to client application answer about error code of operation write.

Following picture shows simple scheme of **operation write** executing:



Obr. 12.: Simple scheme of operation write executing.

Protocol - operation WRITE

Request (any client application)

5,4,3,2,1	header of request (5 bytes)
250	begin of variable frame (1 byte)
variable name	ASCII characters of variable name, or also AliasName
251	end of variable name (1 byte)
code type	code type (1 byte): 1 - integer 2 bytes 2 - long 4 bytes 3 - real 4 bytes 4 - double 8 bytes 5 - string 6 - boolean (0 - false, FF - true) 7 - date 8 bytes 8 - unsigned integer - 2 bytes 9 - unsigned integer - 1 bytes
252	end of code type (1 byte)
value	variable value - number of bytes as defined in code type (if code type is 5, the end of the string have to be check on ASCII char 255)
255	end of value (1 byte)
parita1	number of variables (2 bytes)
parita2	total count of transfered bytes (4 bytes)

Answer (OPCAdapter)

1	header of answer (1 byte). Answer have to be received else it means some transfer error.

Reference to topics:

Special communication protocil 14
Communication TCP/IP client-server without COM, DCOM technologies 13
Operation read 14

2.4.3 Supported data types of OPC variables

Supported daty types of OPC variables

integer	2 bytes
long	4 bytes
real	4 bytes
double	8 bytes
string	undefined, results from protocol
boolean	1 byte: 00 - false, FF - true.
date	8 bytes
unsigned integer	2 bytes
unsigned integer	1 bytes

This list will be extended by another data types, as for example data type **array**, in the near future.

Reference to topics:

2.5 Composition of configuration file

The system is configuravle by simple, but very comfort configurator 20. it ispossible to save the alone configuration in form of Extensible Markup Language (XML) 18 file. We can use stored configuration again, or just modify it. Following picture shows example of configuration file.

Obr. 13.: Composition of configuratin file (example).

Analyse of configuration file

OPCServer
OPC Server, on which will be join the application**OPCAdapter**, is exactly specified:
- name of OPC Server, Prog ID (NAME=Matrikon.OPC.Simulation),
- name of computer (ComputerName=SAE99).

OPCGroup
In hierarchy under OPC Server is defined one OPC group with following properties:
- name (Name=Grupe),
- sign if is OPC group active (IsActive=TRUE)
- information if OPC clien is subscribed by OPC serever (IsSubscribed=TRUE),
- update rate (UpdateRate=1000),
- ...

Under OPC Group is section OPC Items.

OPCItems
Section which defines OPC Items.

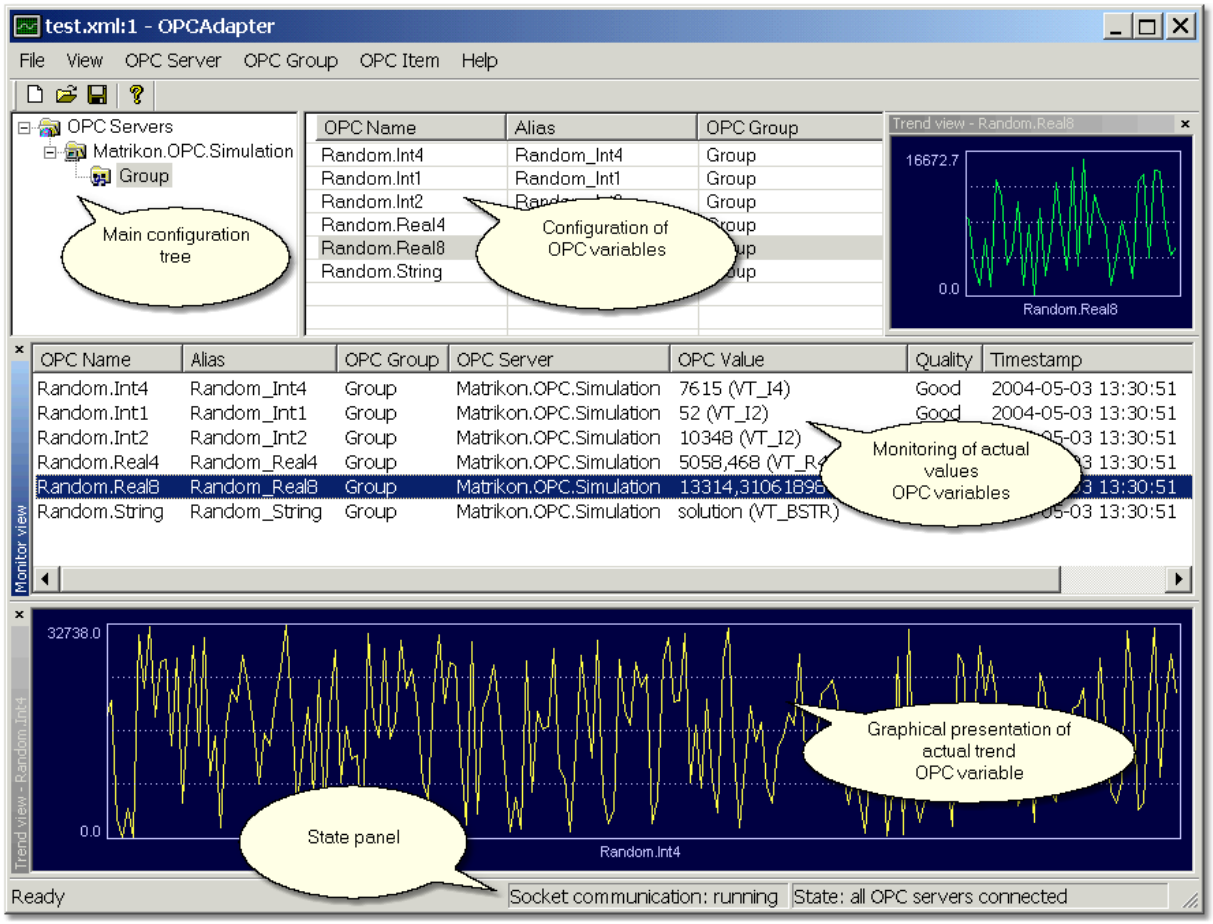
OPCItem
Definition of one OPC variable, which will be monitored. It has following properties:
- name of OPC variable (ID=Random.Int1),
- shoart name (alias) of OPC variable (AliasName=Random_Int1).

AliasName is short label of OPC variable name, which is used for identification OPC variable in special communication protocol 14.

Reference to topics:

OPC client compatible with OPC Data Access 2.0 9
Special communication protocol 14
Operation read 14
Operation write 16

2.6 OPCAdapter with user interface



Obr. 14.: OPCAdapter with user interface, configurator for OPCAdapter NT service (detail).

User interface of application **OPCAdapter** is devided in following parts:

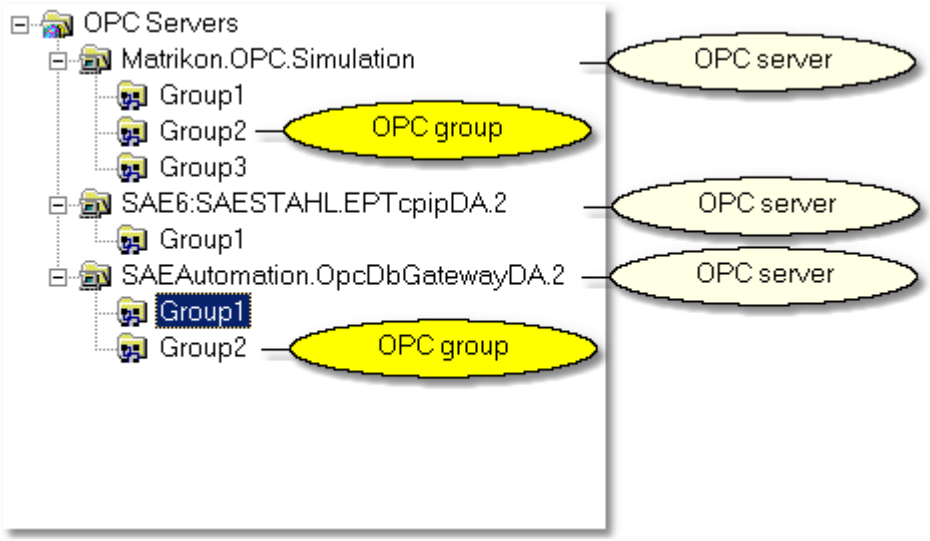
- main configuration tree 21,
- configuration of OPC variables 25,
- monitoring of actual values of OPC variables 27,
- graphical presentation of actual trend of OPC variable 28,
- state panel 30.

Reference to topics:

OPCAdapter 8
Main configuration tree 21
Configuration of OPC variables 25
Monitoring of actuale values of OPC variables 27
Grafical presentation of actual trend of OPC variable 28
State panel 30

2.6.1 Main configuration tree

It shows configured project in form of tree. On the highest level, in the root of the tree is set of **OPCServers**. Allchosen OPC servers belong to this set (example.: SAEAutomation.OpcGatewayDA.2, SAESTAHL.EPTTcpipDA2, Matrikon.OPC.Simulation, ...). Under any OPC server could by defined more **OPC groups**, which exactly define for OPC server scanning and refresh rate.



Obr. 15.: Main configuration tree.

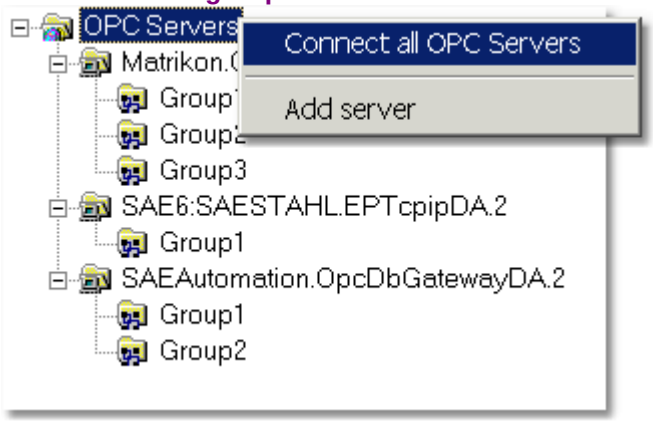
Every level of this structure have own context menu 21.

Reference to topics:

OPCAdapter with user interface 20
Main configuration tree, context menu 21

2.6.1.1 Main configuration tree, context menu

Context menu for group of OPC servers

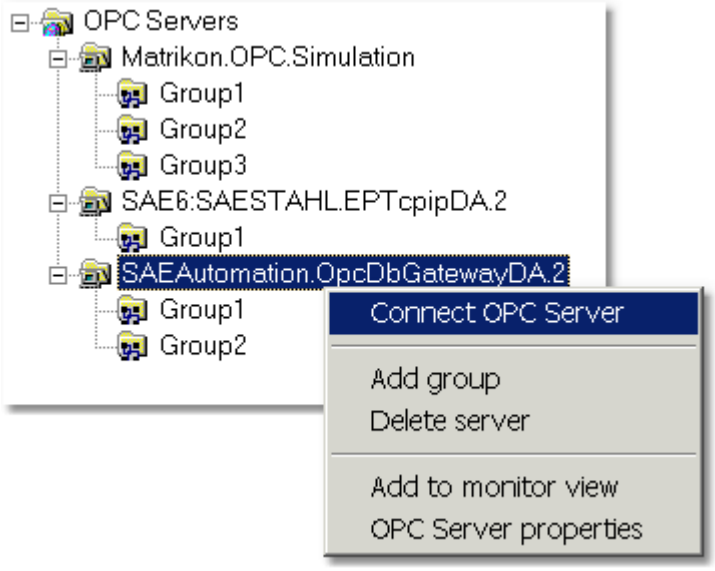


Obr. 16.: Context menu for group of OPC servers.

Connect to all OPC severs
This item allows to **OPCAdapter** to connect to all OPC servers, which are defined in the set of **OPC Servers**.

Add server
This item adds new OPC server to set of **OPC Servers**.

Context menu for OPC server



Obr. 17.: Context menu for OPC server.

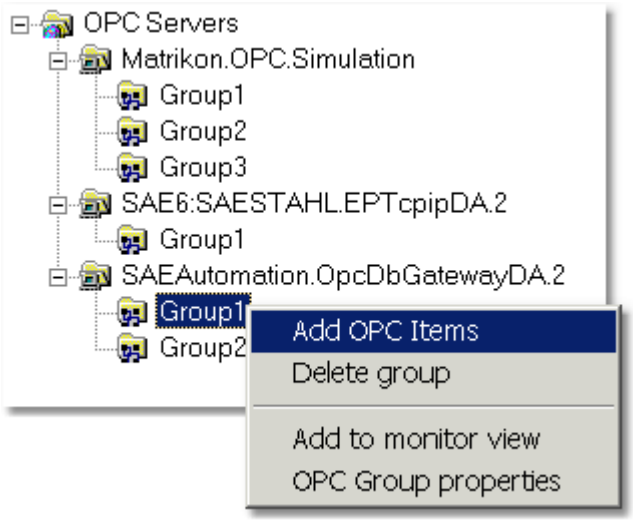
Connect to OPC server
This item allows to **OPCAdapter** to connect to one selected OPC server (exam.: SAEAutomation.OpcDbGatewayDA.2).

Add group
It add the new group foe selected OPC server.

Delete server
It delete selected OPC server from configuration.

Add to monitor
For selected OPC server, it add all OPC variables to main monitoring window.

Context menu for OPC group



Obr. 18.: Context menu for group of OPC servers.

Add OPC Items
It add new OPC variables to OPC group.

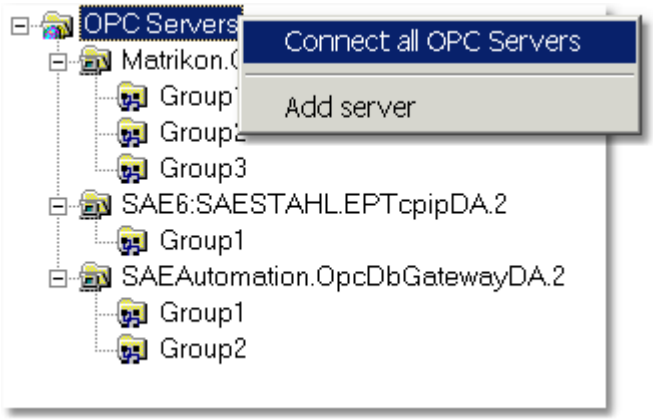
Delete group
It delete selected OPC group from configuration.

Add to monitor view
It add all OPC variable to main monitoring window.

OPC Goup properties
It shows dialog for setting the properties of OPC group.

Reference to topics:
OPCAdapter with user interface 20
Main configuration tree 21

2.6.1.1.1 Context menu for group of OPC servers

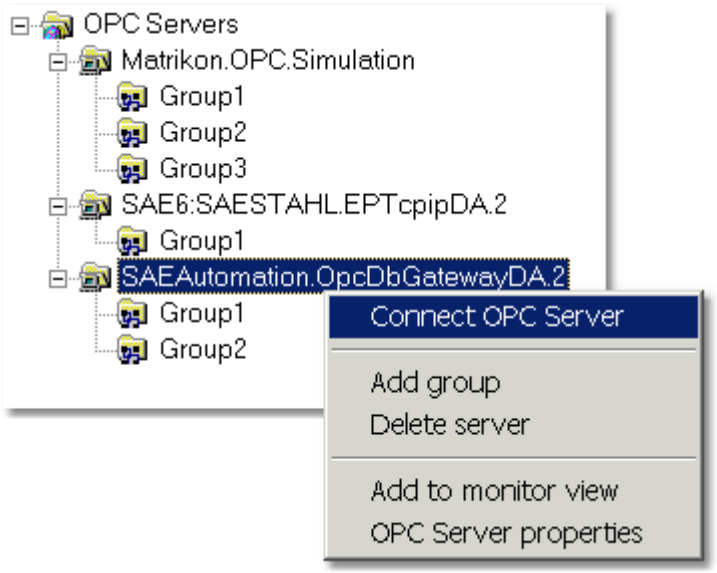


Obr. 16.: Context menu for group of OPC servers.

Connect to all OPC severs
This item allows to **OPCAdapter** to connect to all OPC servers, which are defined in the set of **OPC Servers**.

Add server
This item adds new OPC server to set of **OPC Servers**.

2.6.1.1.2 Context menu for OPC server



Obr. 17.: Context menu for OPC server.

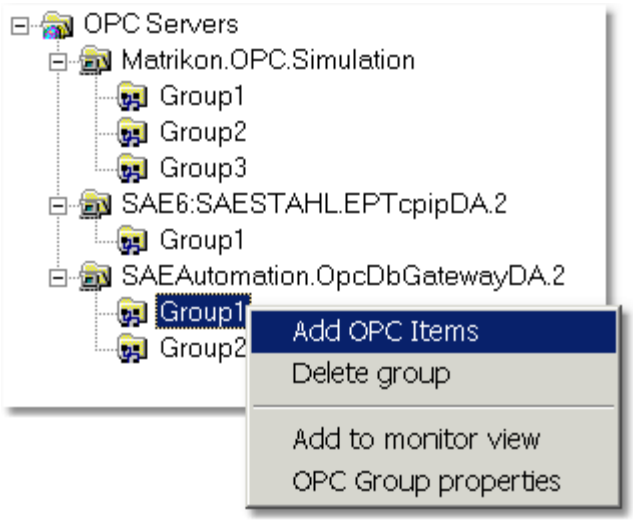
Connect to OPC server
This item allows to **OPCAdapter** to connect to one selected OPC server (exam.: SAEAutomation.OpcDbGatewayDA.2).

Add group
It add the new group foe selected OPC server.

Delete server
It delete selected OPC server from configuration.

Add to monitor
For selected OPC server, it add all OPC variables to main monitoring window.

2.6.1.1.3 Context menu for OPC group



Obr. 18.: Context menu for group of OPC servers.

Add OPC Items
It add new OPC variables to OPC group.

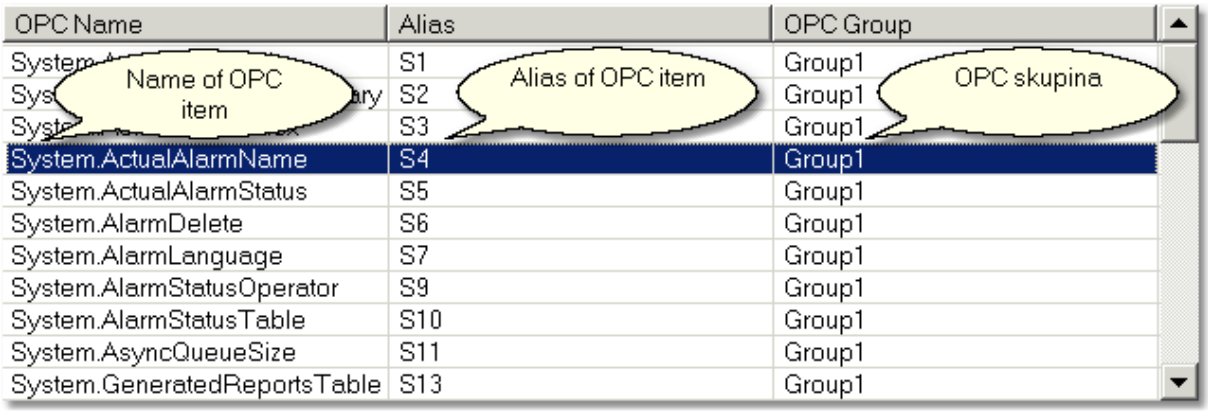
Delete group
It delete selected OPC group from configuration.

Add to monitor view
It add all OPC variable to main monitoring window.

OPC Goup properties
It shows dialog for setting the properties of OPC group.

2.6.2 Configuration of OPC Items

Following picture shows list of **OPC Items**, which are in the one **OPC group** (example.:Group1). Configuration of OPC Items is supported by context menu and keybort shortcuts 26.



Obr. 19.: Configuration of OPC Items.

OPC Neno
Defines all access path to OPC Item.

Alias
Defines short label for name of OPC Item.
- If **OPCAdapter** will be run only as OPC client then it is not necessry to define this item.
- It is necessary to define alias in case, that **OPCAdapter** will be run as TCP/IP server 11. In this case is **Alias** as unique identifier for OPC Item.

After mouse clicking, or pressing the **F2** key, it is possible to modify right in the list of OPC Items.

Obr. 20.:Modification of Alias right in the list of OPC Items.

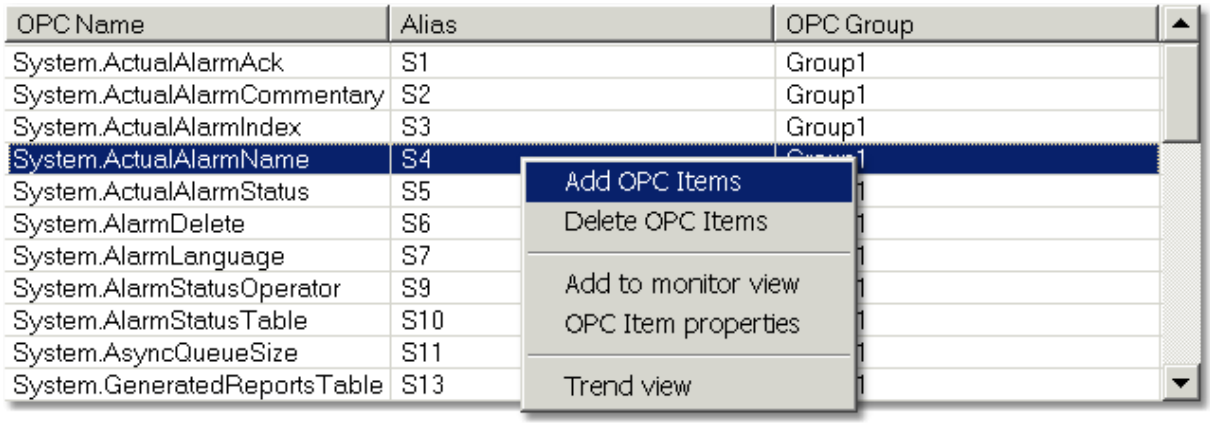
OPC Group
Defines OPC group to which belong the OPC Items.

Reference to topics:

OPCAdapter wtih user interface 20
Configuration of OPC Items, context menu and shortcuts 26
Communication through interfaces OPC, COM, DCOM, TCP/IP 11
Communication TCP/IP client-server without COM, DCOM technologies 13

2.6.2.1 Configuration of OPC Items, context menu and shortcuts

Context menu



Obr. 21.: Configuration of OPC Items, context menu.

Add OPC Items
It add new OPC Items to OPC Group.

Delete OPC Items
It delete selected OPC Items from configuration.

Add to monitor view
It add selected OPC Items to main monitoring window.

OPC Item properties
It shows dialog for settingproperties of OPC Item.

Trend view
It shows dialog for monitoring trend of OPC Item.

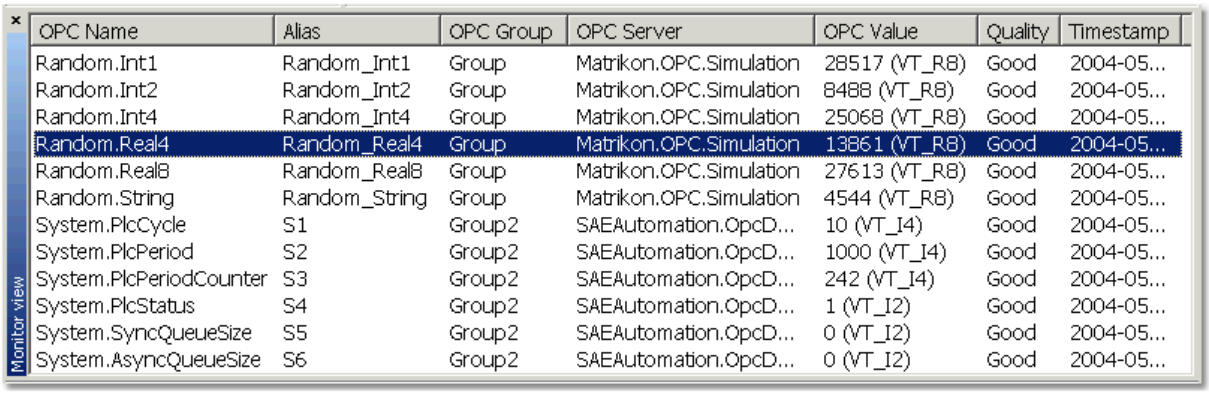
Shortcut keys

Ctrl+A	Select all OPC Items
Del	Delete all selected OPC Items from configuration
F2	Enable cell editing for modify **Alias** of one OPC Item
F5	Reload configuration form project file

Reference to topic:
Configuration of OPC Item 25

2.6.3 Monitoring actual values of OPC variables

Following picture shows actual values of selected **OPC Items.** In monitoring window is possible to watch **OPC Items** several different **OPC Servers** or **OPC Groups**. Configuration of OPC Items is supported by context menu and shortcut keys 28.



OPC Name	Alias	OPC Group	OPC Server	OPC Value	Quality	Timestamp
Random.Int1	Random_Int1	Group	Matrikon.OPC.Simulation	28517 (VT_R8)	Good	2004-05...
Random.Int2	Random_Int2	Group	Matrikon.OPC.Simulation	8488 (VT_R8)	Good	2004-05...
Random.Int4	Random_Int4	Group	Matrikon.OPC.Simulation	25068 (VT_R8)	Good	2004-05...
Random.Real4	Random_Real4	Group	Matrikon.OPC.Simulation	13861 (VT_R8)	Good	2004-05...
Random.Real8	Random_Real8	Group	Matrikon.OPC.Simulation	27613 (VT_R8)	Good	2004-05...
Random.String	Random_String	Group	Matrikon.OPC.Simulation	4544 (VT_R8)	Good	2004-05...
System.PlcCycle	S1	Group2	SAEAutomation.OpcD...	10 (VT_I4)	Good	2004-05...
System.PlcPeriod	S2	Group2	SAEAutomation.OpcD...	1000 (VT_I4)	Good	2004-05...
System.PlcPeriodCounter	S3	Group2	SAEAutomation.OpcD...	242 (VT_I4)	Good	2004-05...
System.PlcStatus	S4	Group2	SAEAutomation.OpcD...	1 (VT_I2)	Good	2004-05...
System.SyncQueueSize	S5	Group2	SAEAutomation.OpcD...	0 (VT_I2)	Good	2004-05...
System.AsyncQueueSize	S6	Group2	SAEAutomation.OpcD...	0 (VT_I2)	Good	2004-05...

Obr. 22.: Monitoring of actual values of OPC Items.

OPC Name
It defines all access path to OPC item.

Alias
It defines short name of OPC Item.

OPC Group
It defines OPC Group to which allows OPC Items.

OPC Server
If defines OPC Server to which allows OPC Items.

OPC Value
It defines actual value of OPC Item.

Quality
It defines quality of OPC Item.

Time

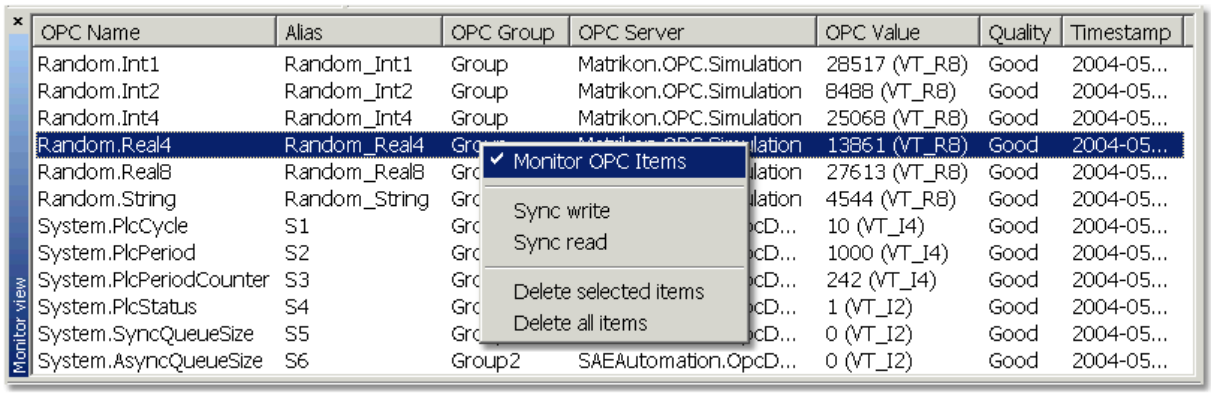
Time stamp of last change of OPC Item.

Reference to topics:

OPCAdapter with user inteface 20
Monitoring of actual values of OPC Items, context menu and shortcut keys 28

2.6.3.1 Monitoring of actual values of OPC Items, cotext menu and shortcut keys

Context menu



Obr. 23.: Monitoring of actual values of OPC Items, context menu.

Monitor OPC Items
It starts or stops monitoring of OPC Items.

Sync write
It provides to user change the value of OPC Item.

Sync read
It provides to user read the actual value of OPC Item.

Delete selected items
Delete all selected OPC Items from monitoring window.

Delete all items
Delete all OPC Items from monitoring window.

Shortcut keys

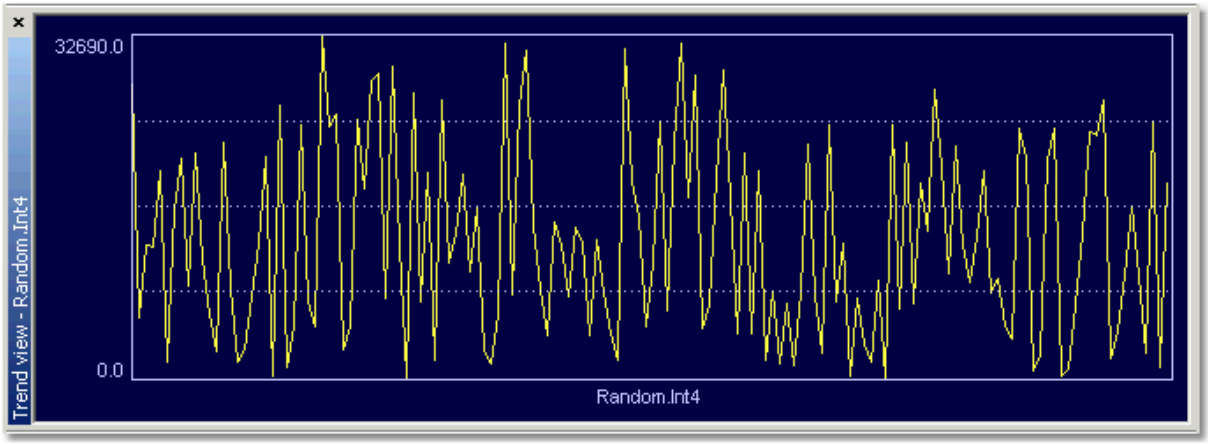
Ctrl+A	Select all OPC Items
Del	Delete selected OPC Items from monitoring window

Reference to topic:

Monitoring of actual values of OPC Items 27

2.6.4 Grpahic presentation of actual trend of OPC Item

Picture shows graph of actual trend of one OPC Item. Vizual properties iof graphic presentation of actual trend could be defined in configuration dialog 29.



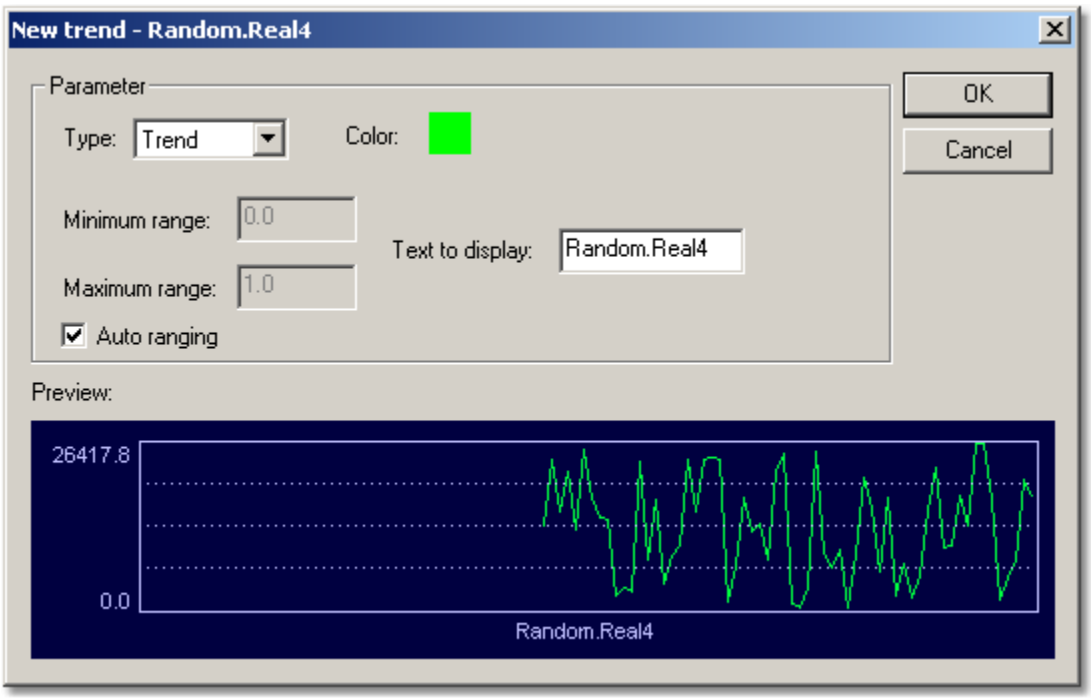
Obr. 24.: Graphic presentation of actual trend of OPC Item Random.Int4.

Reference to topics:

OPCAdapter with user interface 20
Configuration dialog of graphical presentation of acual trend 29

2.6.4.1 Configuration dialog of grafical presentation of actual trend

Following picture shows dialog window which defines property of the **graphical presentation of actual trend**.

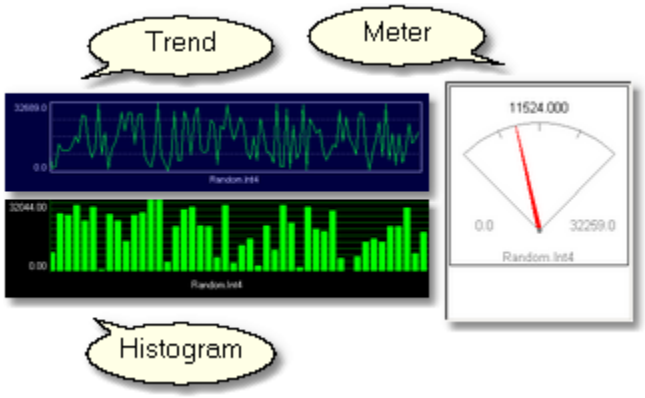


Obr. 25.: Configuration dialog of grafical presentation of actual trend of OPC variable Random.Real4.

Type

It define type of showing graph. The application provides following types of graph:

- Trend
- Meter
- Histogram



Minimal value
It defines minimal value of OPC Item which will by shown in the graph.

Maximal value
It defines maximal value of OPC Item which will by shown in the graph.

Color
It defines color of trend.

Computer setting of range
Sign, that tell us about usein of automatic range of values for actual graph.

OK
New window with defined graph properties will be creat for selected OPC Item.
Zruš
Configuration dialog will be destroye and any trend will not be shown.

Reference to topic:
Graphical presentation of actual trend of OPC Item 28

2.6.5 State panel

Following picture shows **actual state of communication** through interface OPC a TCP/IP 11 in application **OPCAdapter**.

Obr. 26.: Actual state of communication through interface OPC and TCP/IP.

Socket communication
It shows actual stat of communication TCP/IP client-server 13 between application **OPCAdapter** and **TCP/IP client** this application. Communication could have one of the following states:

- **stopped** (communication through interface TCP/IP is stopped),
- **running** (communication through interface TCP/IP is running).

State

It shows actual stat of communication OPC client-server 12 between application **OPCAdapter** and **OPC server**. Communication could have one of the following state:

- **All OPC Servers are disconnected**
  The communication through OPC interface is disconnected for all OPC Servers.
- **All OPC Servers are connected**
  The communication through OPC interface is running.
- **Server 'SAEAutomation.OpcDbGatewayDA.2' disconneted**
  The communication through interface OPC for defined OPC server is disconnected. Defined OPC server 'SAEAutomation.OpcDbGatewayDA.2' is not connected or connection failed.
- **Server 'SAEAutomation.OpcDbGatewayDA.2' connected**
  The communication through interface OPC for defined OPC server is running. Defined OPC server 'SAEAutomation.OpcDbGatewayDA.2' is now connected, connection established.

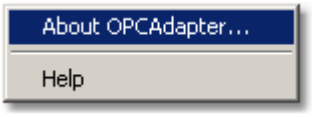
Reference to topics:

OPCAdapter with user interface 20
Communication through interfaces OPC, COM, DCOM, TCP/IP 11
Communication OPC client-server with COM, DCOM technologies 12
Communication TCP/IP client-server without COM, DCOM technologies 13

2.6.6 About application OPCAdapter

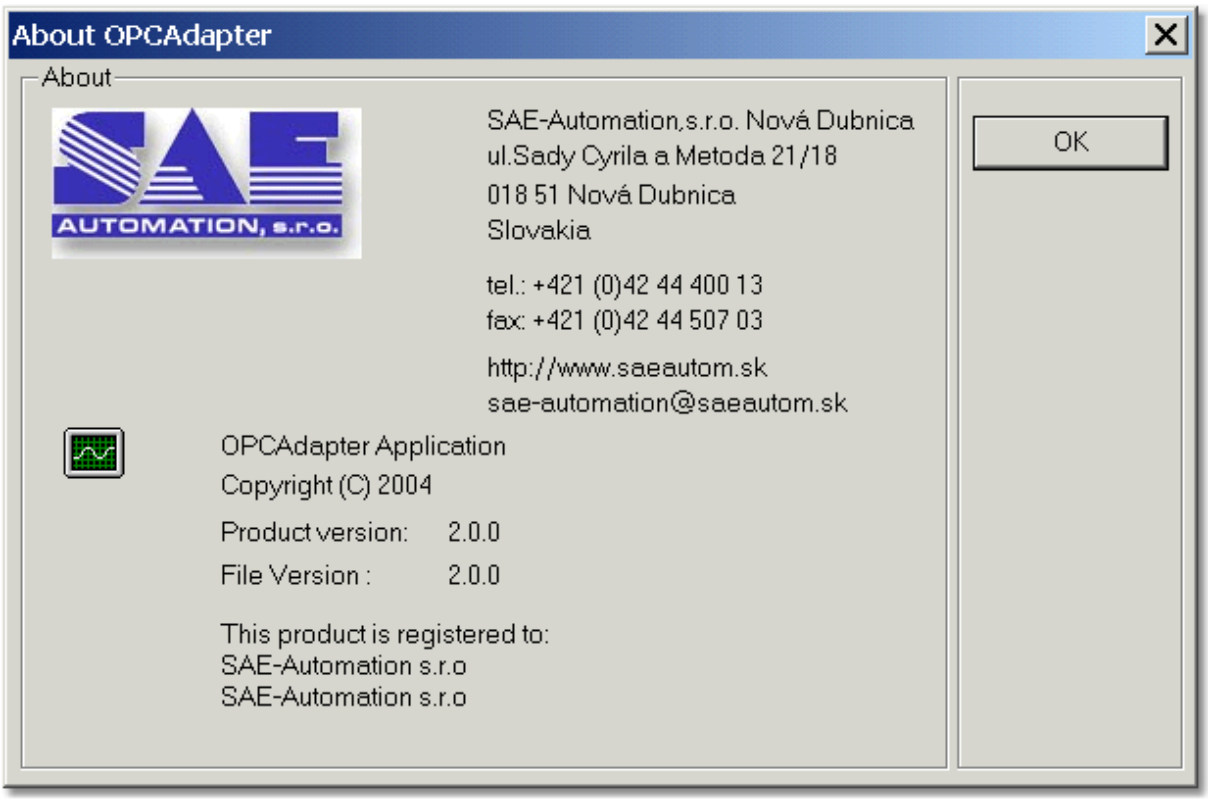
It is possible to reach information about application right from main menu, or by mouse-click on relevant icon in tool panel.

Menu



Icon in tool panel

Dialog with information will by display:



Obr. 27.: About application OPCAdapter.

Reference to topic:
OPCAdapter 8

2.6.7 Main menu

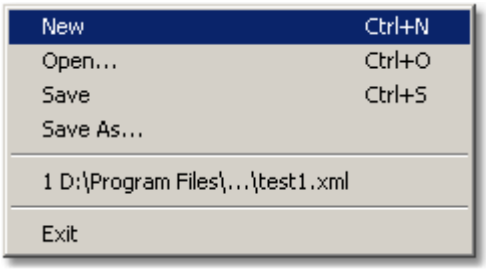
Main menu contains following items:

- File 32,
- View 33,
- OPC Server 34,
- OPC Group 34,
- OPC Item 35,
- Parameters 35,
- Help 37.

Reference to topic:
OPCAdapter with user interface 20

2.6.7.1 Main menu, item File

Following picture shows item **File** for working with configuration file XLM 8.



Obr. 28.: Item File.

New
It makes new configuration.

Open...
It open some stored configuration.

Save
It save actual configuation to configuration file.

Save as...
It save actual configuration to new configuration file.

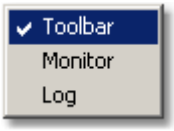
List of last opened configuratio files
He is list of last opened configuration files. After mouse-click on any from list will be this configuration load to the application.

Exit
It end the application.

Reference to topic:
Main menu 32

2.6.7.2 Main menu, item View

Following picture shows item **View** for turn on and off some windows.



Obr. 29.: Item View.

Tool panel
It shows or hide tool panel.

Monitor view
It shows or hide monitor window.

Log view
It shows or hide main log window of application.

Reference to topic:
Main menu 32

2.6.7.3 Main menu, item OPC Server

Following picture shows item **OPC Server** for configuration and control of OPC servers.

Obr. 30.: Item OPC Server.

Connect ot all OPC servers
It starts connection to all defined OPC serevrs in configuration.

Connect to OPC Server
It starts connection to OPC Server, which is selected in the main configuration tree 21.

Add server
It adds new server to configuration.

Delete server
It delete selected OPC Server from configuration.

Reference to topics:
Main menu 32
Main configuration tree 21

2.6.7.4 Main menu, item OPC Group

Following picture shows item **OPC Group** for configuration of OPC Groups.

Obr. 31.: Item OPC Group.

Add group
It adds new OPC group to actual selected OPC Server.

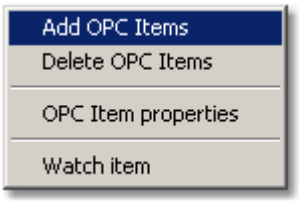
Delete group
It deletes actual selected group from configuration.

Group properties
It defines properties of OPC Group, which is actual select in the main configuration tree 21.

Reference to topics:
Main menu 32
Main configuration tree 21

2.6.7.5 Main menu, item OPC Item

Following picture shows item **OPC Item** for configuration of OPC Items.



Obr. 32.: Menu OPC Item.

Add new OPC Items
It allows to usethe address area of OPC server and adds the new OPC Items to selected OPC Server.

Delete items
It delete selected items from configuration.

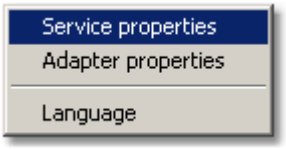
Item properties
It defines properties of OPC Item, which is selected in the main configuration tree 21.

Monitor item
Graphical presentation of actual trend 28 OPC variable.

Reference to topics:
Main menu 32
Main configuration tree 21

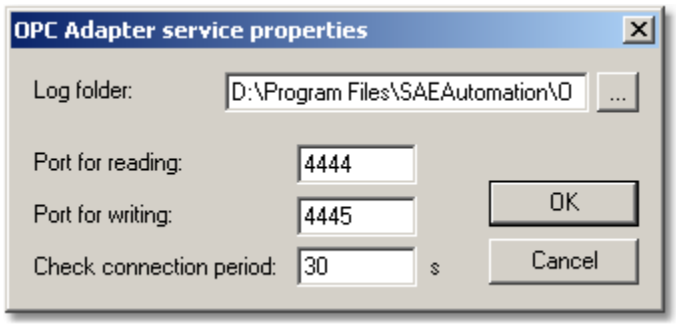
2.6.7.6 Main menu, item Parameters

Following picture shows item **Parameters** for configuration of application parameters.



Obr. 33.: Menu Parameters.

Service properties
Setting of NT service properties of application **OPCAdapter**. The application allows defined the directory for logging file, number of ports for read and write and range of time for checking conection.



Obr. 34.: Setting dialog for properties of NT service OPC Adapter

Range of connection checking - it defines tiem interval for checking connection to OPC server. This time interval is set up in seconds (example.: 60 seconds). If OPCAdapter find, that connection is failed or the OPC server is disconnected, it will try to reconnect to this OPC server after defined time interval.

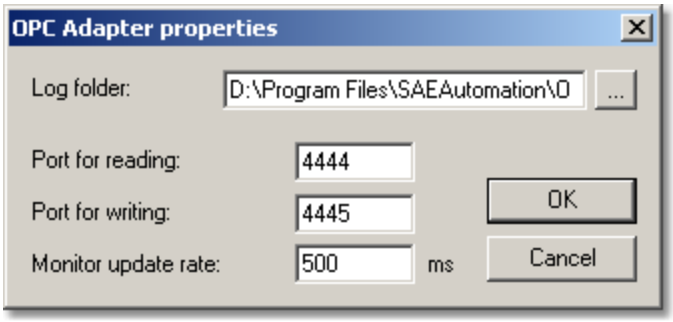
Log directory - it defines directory for logging files.

Port for readin - it defines number of port 11 for operation read 14.

Port for white - it defines number of port 11 for operation write 16.

Programm settings

Property settings for application **OPCAdapter**. The application allows to user set-up the directory for logging file, define ports for reading and writing and define interval for monitor reloade.



Obr. 35.: Setting dialog for application OPC Adapter

Time for monitor reload - it defines time for monitor of OPCAdapter reload. The period is set-up in the miliseconds (example.: 500 ms).

Log directory - it defines directory for logging files.

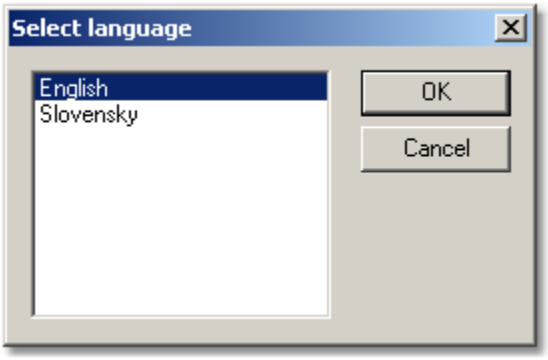
Port for readin - it defines number of port 11 for operation read 14.

Port for white - it defines number of port 11 for operation write 16.

Language

Language setting of application **OPCAdapter**. Today we support two language version of OPCAdapter

- English
- Slovak

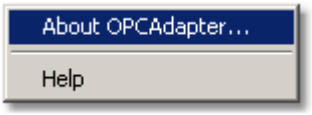


Obr. 36.: Language setting dialog

Reference to topics:
Main menu 32
OPCAdapter NT service 37

2.6.7.7 Main menu, item Help

Following picture shows item **Help**.



Obr. 37.: Menu Help .

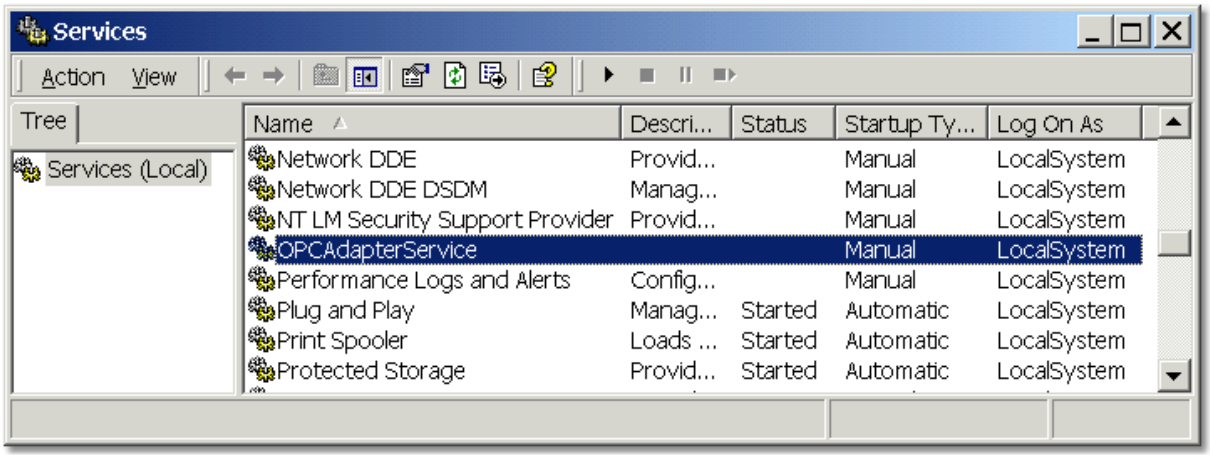
About OPCAdapter...
It shows informaton about OPCAdapter.

Help
It shows help file of application OPCAdapter.

Reference to topics:
Main menu 32

2.7 OPCAdapter NT service

Following picture shows panel of Windows **Services**. Here you can see that the application **OPCAdapter** is installed as NT service with name **OPCAdapterService**.



Obr. 38.: OPCAdapter NT service.

Configuration of application

OPCAdapter NT service is application without user interface. Configuration file for NT service is possible to make by appication OPCAdapter with user interface 20, which have function of configurator.

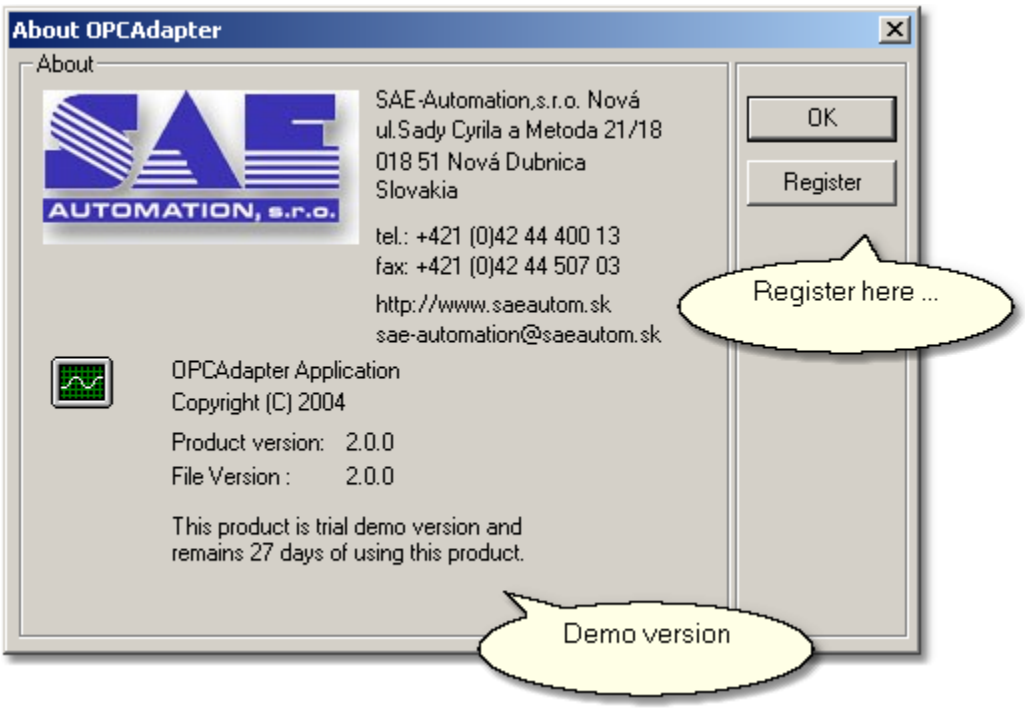
Reference to topics:

OPCAdapter with user interface 20

3 OPCAdapter registration

The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual properly laws and treaties. The SOFTWARE isn't sold, but only the laws for its using are transfer to its user (license 40 is grant)

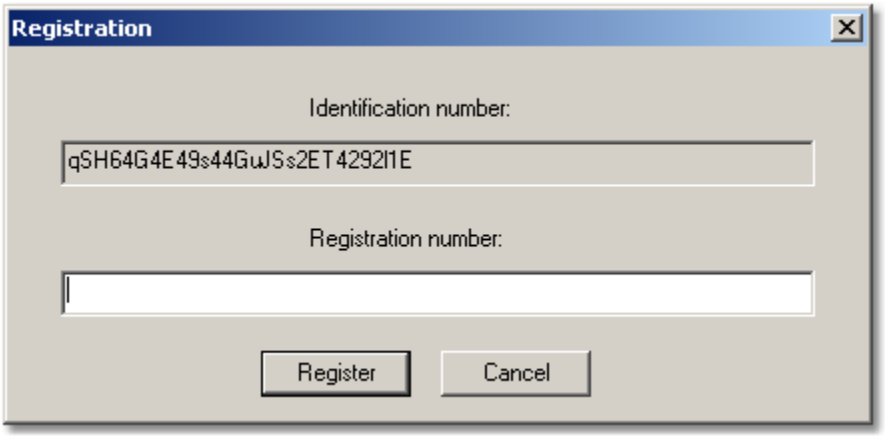
Following picture shows dialog with information about application 31.



Obr. 40.: Application OPCAdapter, demo version.

Dialog informs, that installed application **OPCAdapter** is demo version, which will be deactivated after 30 days.

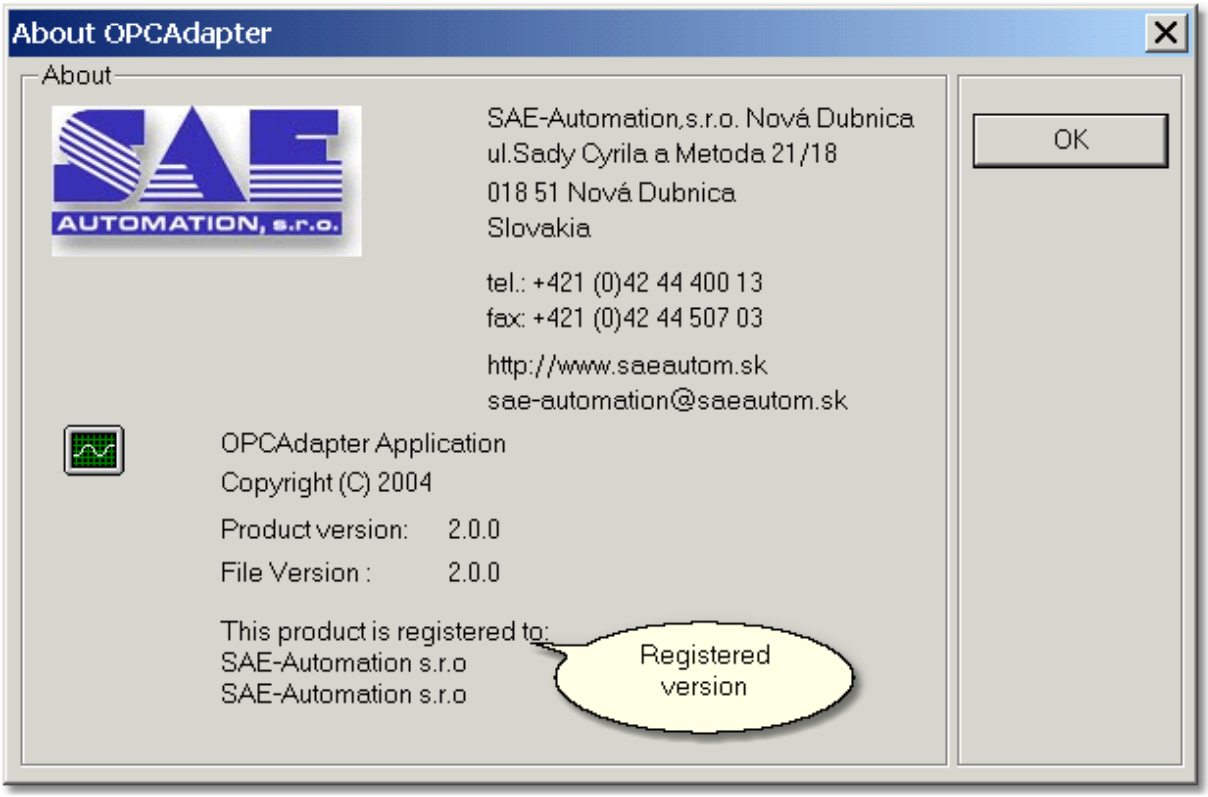
Product registration



Obr. 41.: Dialog for registration of OPC Adapter.

1. In dialog about application OPCAdapter 31 pusth the button **Register**. This will show thwe dialog for registration of your OPCAdapter version.
2. The **identification number** from the dialog copy to the mail and sendas to the e-mailo address sae-automation@saeautom.sk. We SAE-Automation,s.r.o. 58 will send you right **registration number**.
3. Registration number from us, copy to the registration dialog.
4. Push the button **Register**.
5. OK.

The application wiil be now right registered and the dialod about application OPCAdapter will be in following style.



Obr. 42.: Application OPCAdapter, registered version.

3.1 License agreement

SOFTWARE LICENSE AGREEMENT FOR END-USER FOR SOFTWARE FROM COMPANY SAE - Automation, s.r.o., Nová Dubnica

This software license agreement for end-user is legal agreement between you (person or corporation) and company SAE - Automation, s.r.o. Nová Dubnica for software products of company SAE - Automation, s.r.o., which includes computer software and associated storage media with this computer software and printed materials, and may include "online" or electronic documentation delivering on storage media ("SOFTWARE PRODUCT" or "SOFTWARE"). By installing, copying, or otherwise using the SOFTWARE, you agree to be bound by the terms of this software license agreement. If you don't agree to the terms of this software license agreement, promptly return the unused SOFTWARE to company SAE - Automation, s.r.o. and money you paid for SOFTWARE, will be return to you.

SOFTWARE LICENSE

The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual properly laws and treaties. The SOFTWARE isn't sold, but only the laws for its using are transfer to its user (licence is grant).

GRANT OF LICENSE

This software license agreement grants to you following rights:
1. You can use one copy of the SOFTWARE from company SAE - Automation, s.r.o. on one computer. The SOFTWARE is used on computer, when it's loaded in operation memory (RAM) or

installed on storage media (hard-disk, CD-ROM or another storage media).
2. You can print documentation or copy it in arbitrary number under following conditions:
a) All text has to be copy without correction and with all pages.
b) All copies have to have sign of copyright laws of company SAE - Automation, s.r.o. and all another attentions present in document.
c) This documentation can't be distributed in order to make a profit.

UPGRADE
If the SOFTWARE is an upgrade of a product of company SAE - Automation or another company, you now may use or sell that upgraded software only in accordance with SOFTWARE on which was upgrade grant.

COPYRIGHT
All titles and copyrights in and to the SOFTWARE, the accompanying printed materials, and any copies of the SOFTWARE, are owned by SAE - Automation, s.r.o. or its suppliers. The SOFTWARE is protected by copyright laws and international treaty provisions. You have to treat with SOFTWARE as with otherwise product under copyright with the exception of:
a) You can make one copy only for backup you investment.
b) Install SOFTWARE on one computer and first copy remain as backup copy.

FUTHER LAWS AND RESTRICTIONS
1. You can not divide and use separately single SOFTWARE components on several computers.
2. You can not rent or lend the SOFTWARE.
3. Copyright transfer. You can transfer copyright for SOFTWARE using to third person, including this license agreement. You can not remain any copies and you have to remove entire SOFTWARE, including all components, data media and printed materials. Third person, on which you transfer SOFTWARE, have to agree with the terms of this software license agreement. If the SOFTWARE is an upgrade, you have to transfer all previous versions, on which the upgrade was grant.
4. If you don't fulfil the terms and conditions of this software license agreement SAE - Automation, s.r.o. reserves the right to fail this license agreement for SOFTWARE. In that case you have to destroy all your copies of the SOFTAWRE.
5. Company SAE - Automation does not take over any furthers warranties resulted from using the SOFTWARE.
6. If any further cooperation or exploitation from third party software is needed for functionality of the SOFTWARE form SAE - Automation, s.r.o., you accept the responsibility for observation of license agreement of third party supplier, while it was not agreed differently by special agreement between you and SAE - Automation, s.r.o.
7. The end-user responds to claims made by breach of contract.

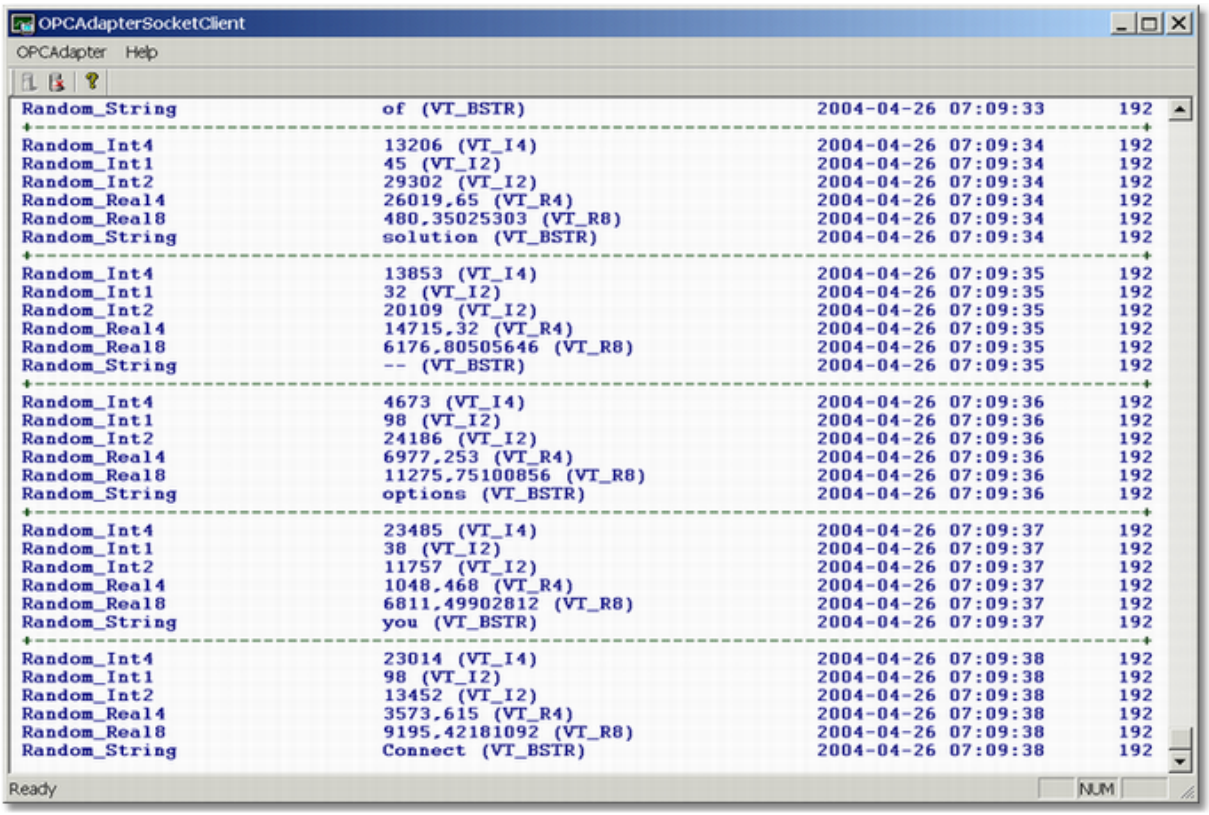
4 OPCAdapterSocketClient

What is it OPCAdapterSocketClient?
It is application which communicate with application **OPCAdapter** throughl TCP/IP 13 protocol. Through user interface it shows actual state of inside memory 11 of application **OPCAdapter**.

In communication TCP/IP client-server 13 it represents client, which in defined time interval sending requests to TCP/IP server 11. After sending the request 14 it is waiting for answer 54, which shows in the main window.

What we offer to you?
We offer you free very effectively tool for monitoring different types of variables. On our web presentation www.saeautom.sk you can find not only this application but also its complete source

codes.



Obr. 43.: OPCAdapterSocketClient - simple example of monitoring application.

Reference to topic:
Introduction 3

4.1 Communication protocol

Communication between **OPCAdapter** and **OPCAdapterSocketClient** is realized through TCP/IP 13 protocol. Connection is defined by **IP address** and Port 11 number.

IP address

It uniform identified computer on which the application **OPCAdapter** is running. Client application have to know the IP Address to make connection with this computer. The postme will not bring zou a leter without the right adress.

Port

The Port is defined by unique location, to which the application **OPCAdapter** could send messages and receive the answers or different messages.
On one computer run more applications in one time so definition of IP Adress is not comfortable. Also in the real life it so, that at the one address, one house, or one flat live more people. And the postman have to know who can read this letter.

Application layer of TCP/IP protocl, special communication protokol
Special communication protocol 14 create application layer of TCP/IP 13 protocol, which exactly defines available operations.

Reference to topics:
Special communication protocol 14
Operation read 14
Operation write 16

4.2 Source code

Short look to source code
In this part we brings you short look to the implementation of main parts of application **OPCAdapterSocketClient**.

In application **OPCAdapterSocketClient** is implemented only communication one-way communication (read-only), source code included only subset of communication protocol. It is part for operation read 14. Example on the operation write 16 one variable is not implementad, but is very simple and similar to operation read.

Within basic functions, which **OPCAdapterSocketClient** have to have:
- sending of request 43 (read actual list of variables),
- answer receive (list of variables),
- decoding and checking the answer 47 (checking the header of answer),
- decoding one transfered OPC variable 49,
- showing answer.

Reference to topics:
Main working thread 43
Answer decoding 47
Decoding one transfered OPC variable 49

4.2.1 Main working thread

OPCAdapterSocketClient send request for readin actual state of inside memory 11 of application **OPCAdapter**.

Answer to request is represented by following technique:
- OPCAdapter is accesible 45,
- OPCAdapter is accesible, but recieved answer has wrong format 46,
- OPCAdapter is not accesible 47.

Sending request and following recieving answer is implemented in the main function of working thread `Thread_PeriodicallyReadServerData`. Receive answer is decoded in the function CEngine::ParseResponse 47.

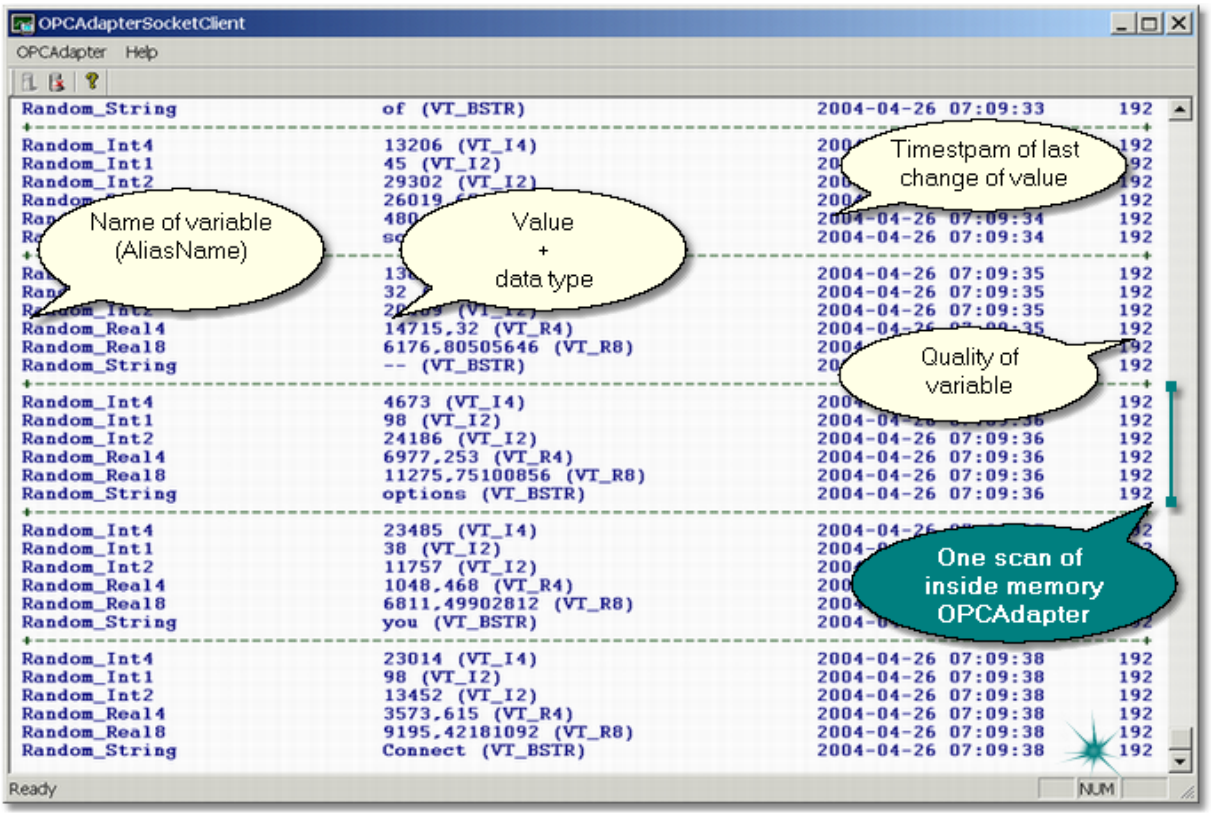
```
UINT Thread_PeriodicallyReadServerData(LPVOID lpParam)
{

	CEngine*		pEngine = (CEngine*) lpParam;
	int			nBytes = 0;

	CBlockingSocket		bsClient;
	CSockAddr		saServer;
	char			request[_REQUEST_LENGTH];
	BYTE			response[_RESPONSE_LENGTH];


	try
	{
		// @flow0 | create a socket server address
		saServer = CBlockingSocket::GetHostByName(	pEngine->m_strIPAddress,
								pEngine-
>m_nPort);
	}
	catch(CBlockingSocketException* e)
	{

		// Error: A socket server address not created!
		char error[200];
		*error=0;
		e->GetErrorMessage(error, sizeof(error));
		e->Delete();

		if(pEngine)
		{
			pEngine->WriteBlock("\r\nServer not available! ", _COLORREF_ERROR);
			pEngine->WriteBlock(error,  _COLORREF_ERROR);
		}

		Beep(500, 100);
		return 1;
	}


	// @flow0 | create the request header '12345'
	for(BYTE i=0;i<5;i++)
	{
		request[i] = i+1;
	}


	while(1)
	{

		// @flow1 | wait for a timeout or stop event (stop data reaging)
		DWORD dwRet = WaitForSingleObject(	g_eventCloseDataReading, // stop the data
reading
							pEngine->m_nUpdateRate); // Timeout

		if (dwRet == WAIT_OBJECT_0)			// stop event (stop data reaging)
		{
			// @flow1 | if socket already exists
			bsClient.Close();

			break;
		}

		if (dwRet == WAIT_TIMEOUT)			// timeout
		{

			try
			{

				// @flow1 | if socket already exists
				bsClient.Close();

				// @flow1 | create a socket
				bsClient.Create();

				// @flow1 | connect to a socket server
```

Reference to topics:

OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47
Answer decoding 47

4.2.1.1 OPCAdapter is accesible

If the asked server accesible and communication is correct, then on the main screen of the application are presents all transfered OPC variables, which are stored in the inside memory 11 of **OPCAdapter**.



Obr. 45.: Communication is correct, list of transfered OPC variables is shown on the screen.

Reference to topics:

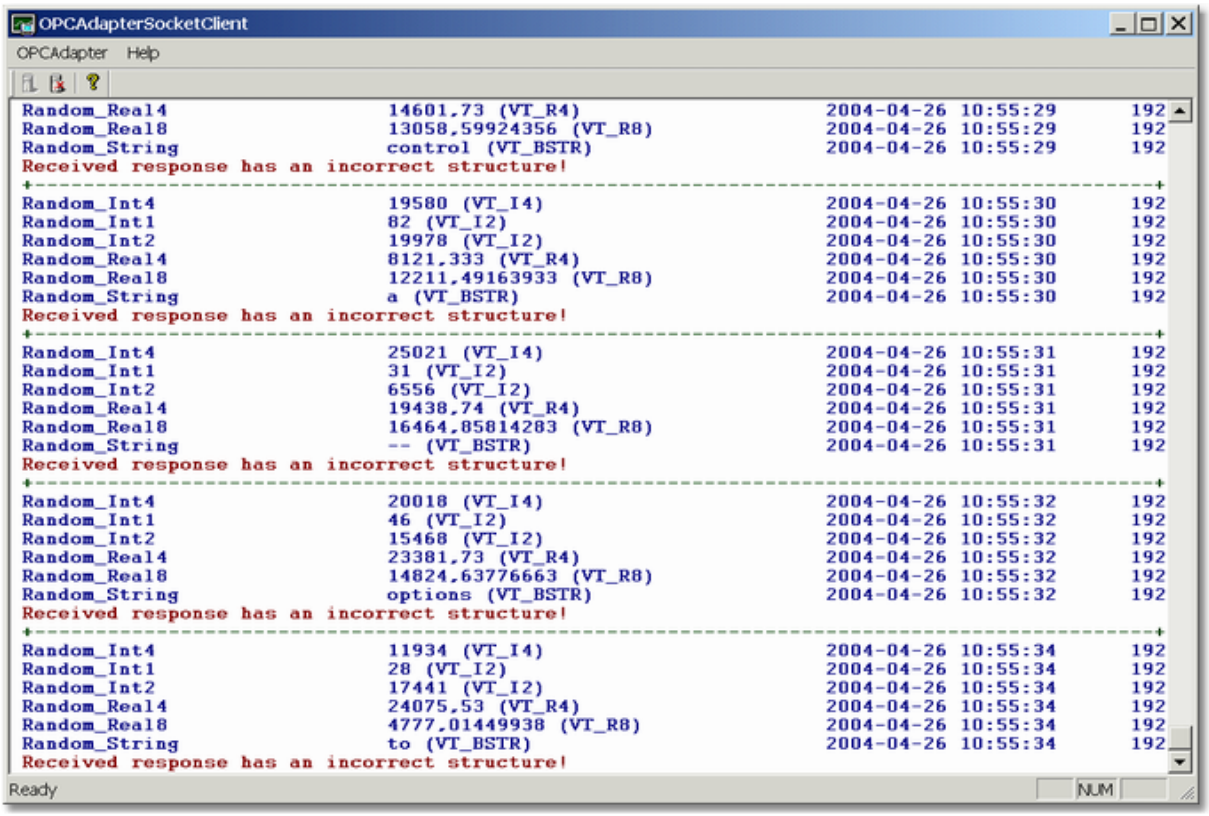
Main working thread 43
Answer decoding 47
Decoding of one transfered OPC variable 49

Operation read 14
Operation write 16

OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47
Supported data types of OPC variables 17

4.2.1.2 OPCAdapter is accesible, but received answer has wrong format

If the askedserver isaccesible, but on screen is error message **Received response has an incorrect structure!**, then the client detects that the receiving message has bad format. One of the possible problems could be, that the receive message includes some not supported data type of OPC variable 17.



Obr. 46.: Receive message has wrong format.

Reference to topics:

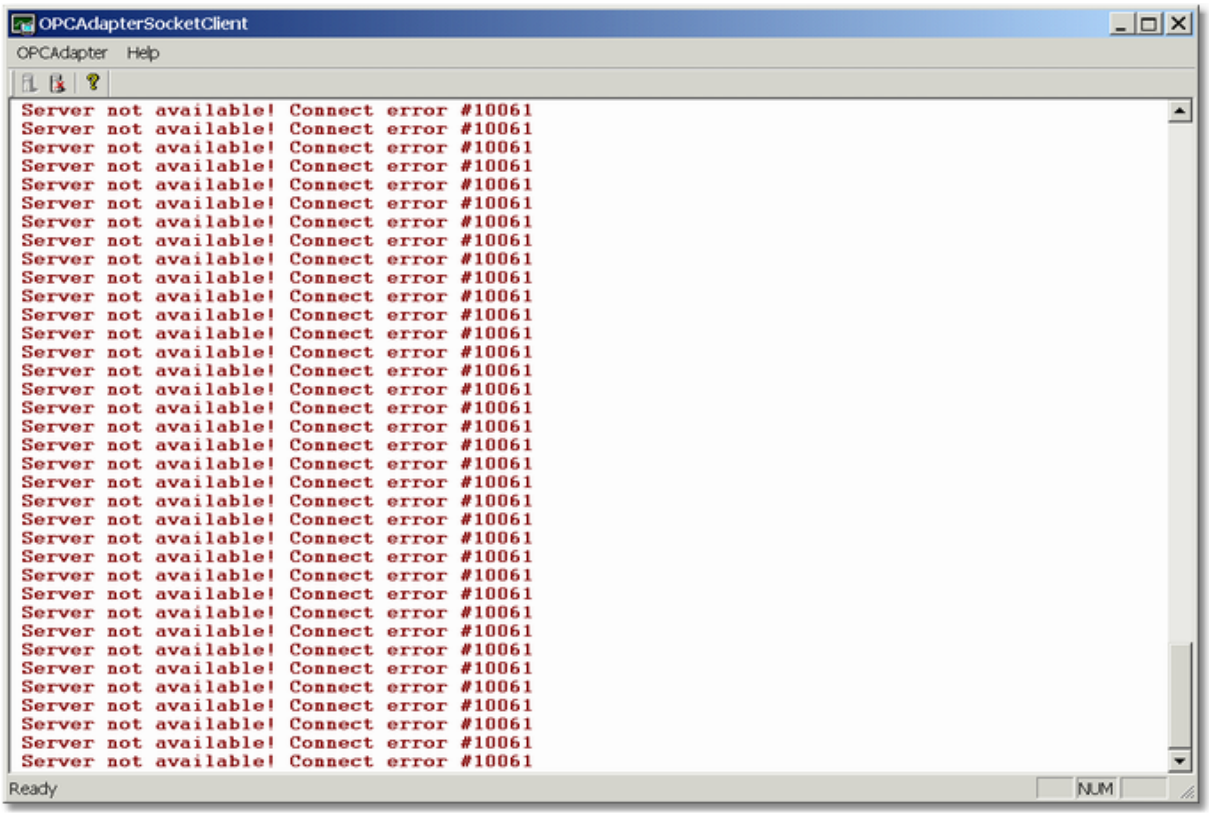
Main working thread 43
Answer decoding 47
Decoding of one transfered OPC variable 49

Operation read 14
Operation write 16

OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47
Supported data types of OPC variables 17

4.2.1.3 OPCAdapter is not accesible

If the asked server is not accesible, than ther is detected following error message in the main screen of the application **Server not available! Connect error #10061**.



Obr. 47.: OPCAdapter (server) is not accesible.

Reference to topics:

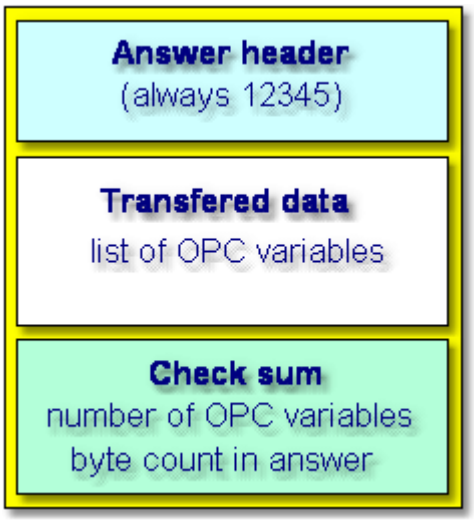
Main working thread 43
Answer decoding 47
Decoding of one transfered OPC variable 49

Operation read 14
Operation write 16

OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47
Supported data types of OPC variables 17

4.2.2 Operation read, decoding answer

The following picture shows the structure of the answer for operation read 14.



Obr. 48.: Operation read, answer format.

:Answer format for operation read is:

- header - (always 12345),
- transfered data,
- check sum.

In the function `CEngine::ParseResponse` are transfered date decoding to single OPC variables. Detailed decoding of single variable is made in the function CEngine::ParseDataItem 49.

```
BOOL CEngine::ParseResponse(BYTE* response)
{

      // @flow0 | Response
/*    response[0] = 1;              // Header - 12345 (operation read)
      response[1] = 2;
      response[2] = 3;
      response[3] = 4;
      response[4] = 5;
      response[5] = ?;              // Parita2 4 bytes (the count of all bytes in response)
      response[6] = ?;
      response[7] = ?;
      response[8] = ?;*/


      DATA_ITEM data_item;

      // @flow0 | others bytes
      int pos=9; // cursor position
      while((pos < _RESPONSE_LENGTH) && (response[pos] != 255))
      {

            // @flow1 | reset a data item
            ::ZeroMemory(&data_item, sizeof(DATA_ITEM));

            // @flow1 | get a new data item
            if(!ParseDataItem(response, pos, &data_item))
            {
                  // Error: bad response structure
                  return FALSE;
            }

            // @flow1 | the formated data item
            CString sDataItem = Format(&data_item);

            // @flow1 | write the formated data item to view
            WriteBlock(    sDataItem,     _COLORREF_DATAITEM);
      }

      return TRUE;
}
```

Reference to topics:

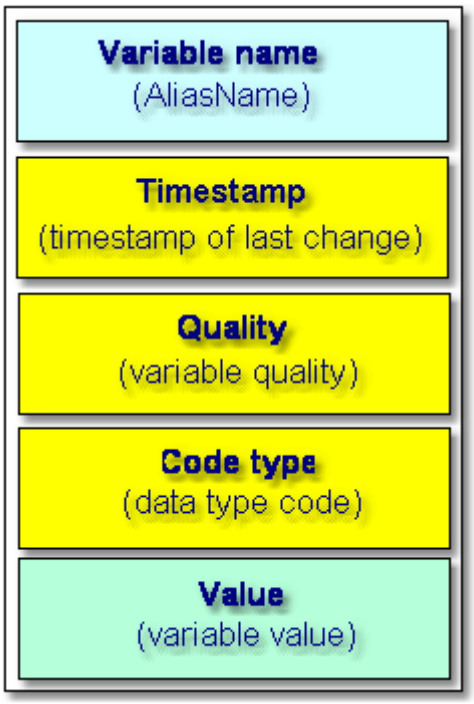
Main working thread 43
Answer decoding 47
Decoding of one transfered OPC variable 49

Operation read 14
Operation write 16

OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47
Supported data types of OPC variables 17

4.2.3 Operation read, decoding one transfered OPC variable

Following picture shows format of the transfered OPC variable for operation read 14.



Obr. 49.: Opaeration read, format of transfered OPC variable.

Format of transfered OPC variable for operation read:

- name of variable,
- timestamp,
- quality,
- code type,
- value.

Detailed design of format you can find in the part of operation read 14. Single items of structure OPC variable are decding in function `CEngine::ParseDataItem`.

```
BOOL CEngine::ParseDataItem(BYTE* response, int & pos, DATA_ITEM* pItem)
{

	// 250
	if(FAILED_POINTER(pos) || (response[pos] != 250)) // Data item frame - Start (250)
	{
		// Error: bad response structure
		return FALSE;
	}

	// @flow0 | Alias - data item alias (ascii characters of name)
	if(FAILED_POINTER(++pos))                          // Alias - Start
	{
		// Error: bad response structure
		return FALSE;
	}

	for(int j=0; j<_MAX_DATA_ITEM_ALIAS_NAME_LENGTH; j++)
	{

		if(response[pos] == 251)                   // Alias - Stop (251)
		{
			pItem->Alias.Length = j;
			pItem->Alias.Name[j] = 0;
			break;
		}

		pItem->Alias.Name[j] = (char)response[pos];

		if(FAILED_POINTER(++pos))
		{
			// Error: bad response structure
			return FALSE;
		}
	}

	if(FAILED_POINTER(pos) || (response[pos] != 251))// alias was not found!
	{
		// Error: bad response structure
		return FALSE;
	}


	// @flow0 | Timestamp
	if(FAILED_POINTER(++pos))
	{
		// Error: bad response structure
		return FALSE;
	}
	pItem->Timestamp = *((DATE*)&response[pos]);

	if(FAILED_POINTER(pos += 8) || (response[pos] != 252))// timestamp was not found!
	{
		// Error: bad response structure
		return FALSE;
	}


	// @flow0 | Quality
	if(FAILED_POINTER(++pos))
	{
		// Error: bad response structure
		return FALSE;
	}
	pItem->Quality = response[pos];

	if(FAILED_POINTER(++pos) || (response[pos] != 253))  // quality was not found!
	{
		// Error: bad response structure
		return FALSE;
	}


	// @flow0 | Type
	if(FAILED_POINTER(++pos))
```

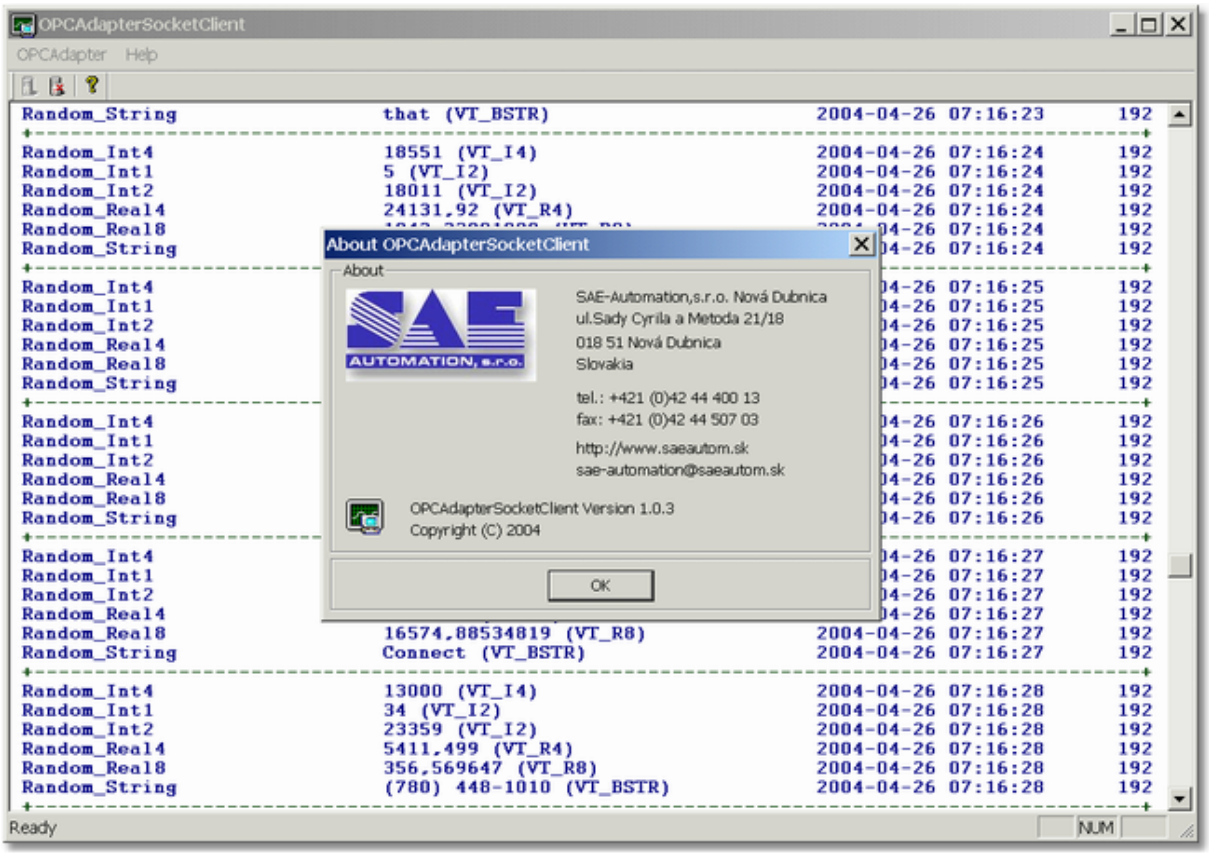
Reference to topics:

Main working thread 43
Answer decoding 47
Decoding of one transfered OPC variable 49

Operation read 14
Operation write 16

OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47
Supported data types of OPC variables 17

4.3 User interface



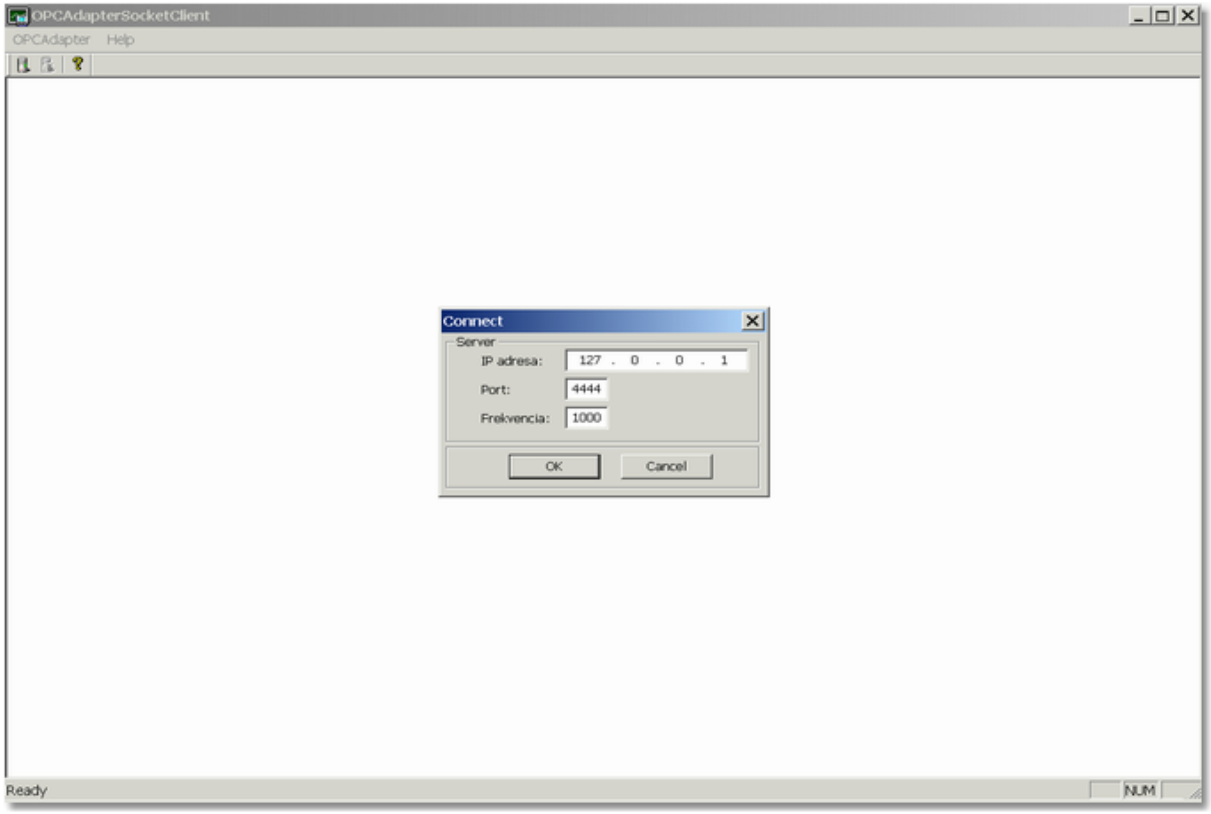
Obr. 50.: OPCAdapterSocketClient - simple application.

Odkazy na èlánky:

OPCAdapterSocketClient 41
OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47

4.3.1 First run of OPCAdapterSocketClient

Following picture shows user interface which will be shown after first run of the application **OPCAdapterSocketClient.exe**.



Obr. 51.: First run of application OPCAdapterSocketClient.

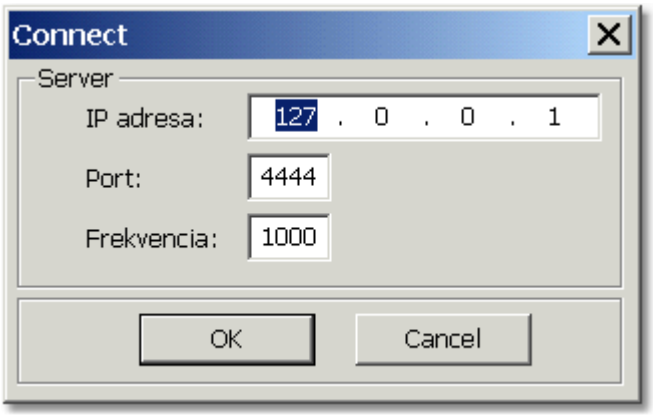
In front is dialog Connect 54, which will be shown always before start communication.

After pushing button **OK**, client will try to establish communication with server 54.

Reference to topics:

Interpretation of receiving answer 54
OPCAdapterSocketClient 41
OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47

4.3.1.1 Dialog Connect



Obr. 52.: Dialog Connect.

IP adresa
Expressly identify computer on which is running application **OPCAdapter**. Default value 127.0.0.1 is address of local machine.

Port
Number which defined location for reading messages from application **OPCAdapter**. Default value for operation read 14 is 4444.

Frequency
It is time interval, on wich the request to the server will be send. Default value is 1000, what means 1000 miliseconds (1 second).

OK
After push the button, the client start the communication with server.
Cancel
After push the button the dialog will be cancel.

Reference to topics:
Interpretation of receiving answer 54
OPCAdapterSocketClient 41
OPCAdapter ia accesible 45
OPCAdapter is accesible, but received answer has wrong format 46
OPCAdapter is not accesible 47

4.3.1.2 Interpretation of receive message

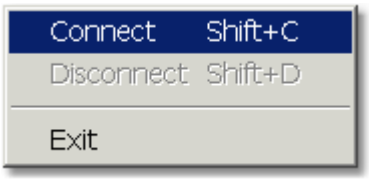
Answer to the request is implemented by foolowing:
- OPCAdapter is accesible 45,
- OPCAdapter is accesible, but the message format is wrongò 46,
- OPCAdapter nie je dostupný 47.

Reference to topics:
Interpretation of receiving answer 54
OPCAdapterSocketClient 41

4.3.2 Start of communication

Strat of communication is possible to make from main menu, or by mouse-click on the corresponding icon in the tool panel.

Menu



With mouse-click to item in the menu, or with shortcut key **OPCAdapter** ⇨ **Connect** (Shift+C).

Tool panel

After pushing one of the previous button the dialog Connect 54 will be display.

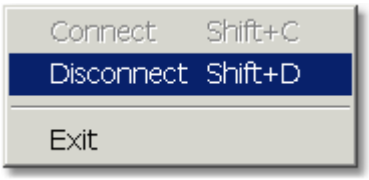
Reference to topics:

Dialog Connect 54
Stop of communication 55
About application OPCAdapterSocketClient 56
End of application 57

4.3.3 Stop of communication

Communication disconnect is possible to make from main menu or by mouse-click on the icon in the tool panel.

Menu



By mouse-click on the item in the menu or with shortcut key **OPCAdapter** ⇨ **Disconnect** (Shift+D).

Tool panel

Reference to topics:

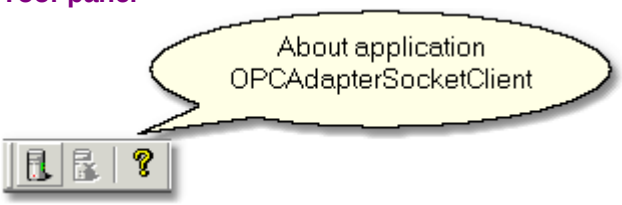
Dialog Connect 54
Stop of communication 55
About application OPCAdapterSocketClient 56
End of application 57

4.3.4 About appliation OPCAdapterSocketClient

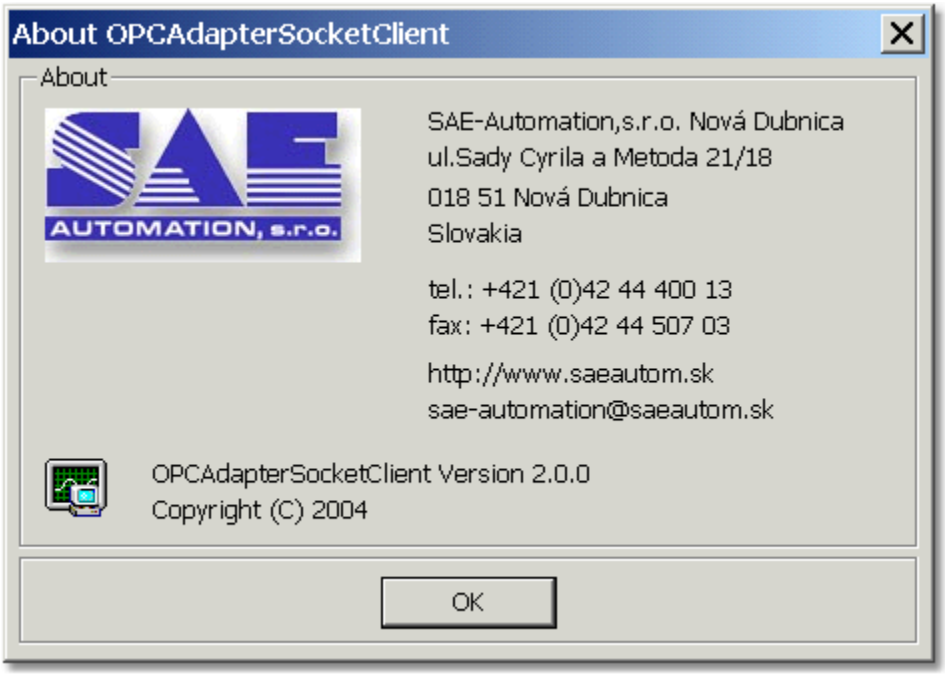
Information about application could be display from main menu or from tool panel.

Menu

Tool panel



The following dialog with information will be display:



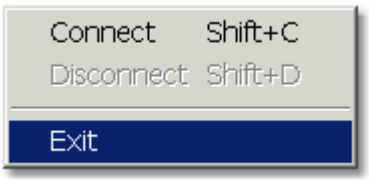
Obr. 53.: About application OPCAdapterSocketClient.

Reference to topic:
OPCAdapterSocketClient 41

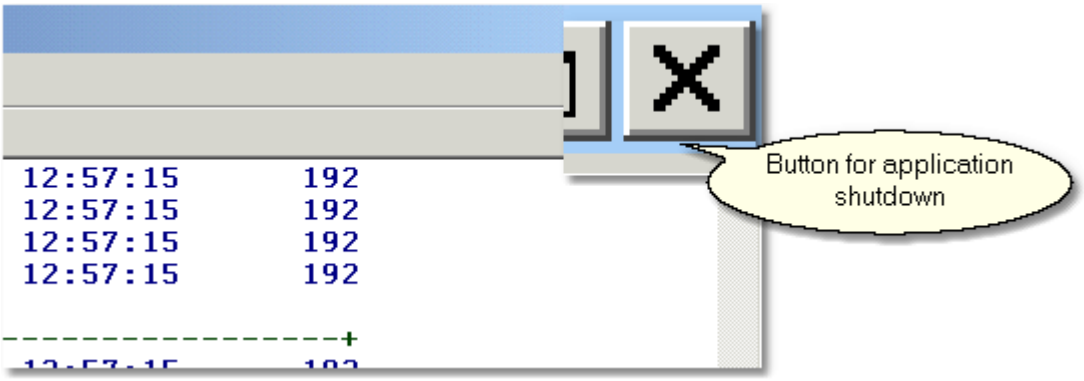
4.3.5 End of application

From main menu or with button for application shutdown, it is possible to finish work with the OPCAdapterSocketClient application.

Menu



Button for application shutdown



5 Final summary

Program package **OPCAdapter** from company SAE-Automation s.r.o. brings on the market new multi-use software, which provide simple, steady and executive process for data transfer between application, computers and alone platforms.

Utilize
Program package **OPCAdapter** was develeped and implemented in regards to fulfillment following requirements:

- simplify access to OPC servers,

- enable data from OPC servers to outside world (computer and platform independence),

Helpful properties

- full-value OPC client (OPC Data Access 2.0),
- possibility of connection to several OPC servers in one time,
- display of actual values from OPC servers,
- graphical presentation of actual trends,
- accesible data from OPC server to outside world through TCP/IP.

6 Contact

SAE-Automation,s.r.o. Nová Dubnica
ul.Sady Cyrila a Metoda 21/18
018 51 Nová Dubnica
Slovakia

tel.: +421 (0)42 44 400 13
fax: +421 (0)42 44 507 02

e-mail: sae-automation@saeautom.sk
internet: http://www.saeautom.sk