

CyBro OPC Server User Manual

version 33

applies to CyBro OPC Server v3.0.8 and later



© 2003-2019 Cybrotech Ltd.

Index

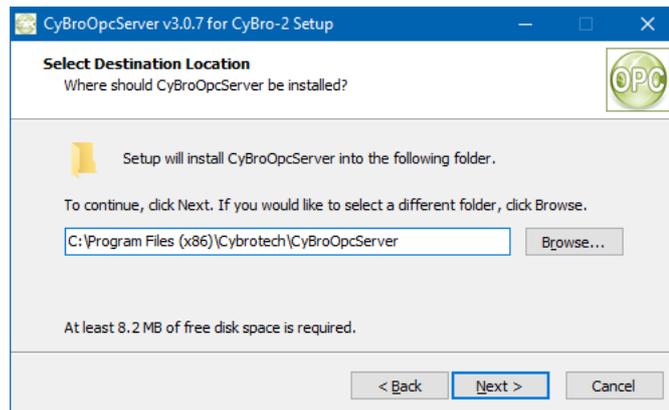
Index	2
Installation	3
Activation code	3
Controllers	4
Options	4
Toolbar	5
Status panel.....	5
Server status	5
OPC interface.....	6
Network monitor	6
Controller status	6
Animation	6
Tag status	6
Process priority.....	7
Tag quality	7
System tags.....	8
DCOM setup.....	9
Mutual user accounts	9
System-wide DCOM settings	9
Specific DCOM settings	10
Firewall exception rules.....	11
Troubleshooting	11
Keyboard shortcuts.....	12
Technical specifications	13

Installation

OPC (OLE for Process Control) is a specification standardized by OPC foundation, which enables clients to access hardware data via OPC servers in a common, well defined way.

CyBro OPC Server enables OPC clients to access Cybro controllers. Clients may read and write whole memory space, including the attached IEX-2 modules.

To install CyBro OPC Server, start the installation and follow instructions. Recommended directory is "C:\Program Files\Cybrotech\CyBroOpcServer".



Installation does the following:

- unpack CyBro OPC files into specified directory
- create start menu group and icons
- register OPC server to be visible for clients
- install redistributable OPC core components

Note: administrator rights are required.

To upgrade server, install new version into the same directory, without uninstalling previous one. User settings will be preserved. Before upgrading, close SCADA and shut down OPC server.

To uninstall server, start Control Panel, Add or Remove Programs, select CyBro OPC Server and press Remove button. OPC core components must be uninstalled separately.

Activation code

To use CyBro OPC Server, valid activation code is required. For activation code contact Cybrotech. Regarding the application size, activation code may be:

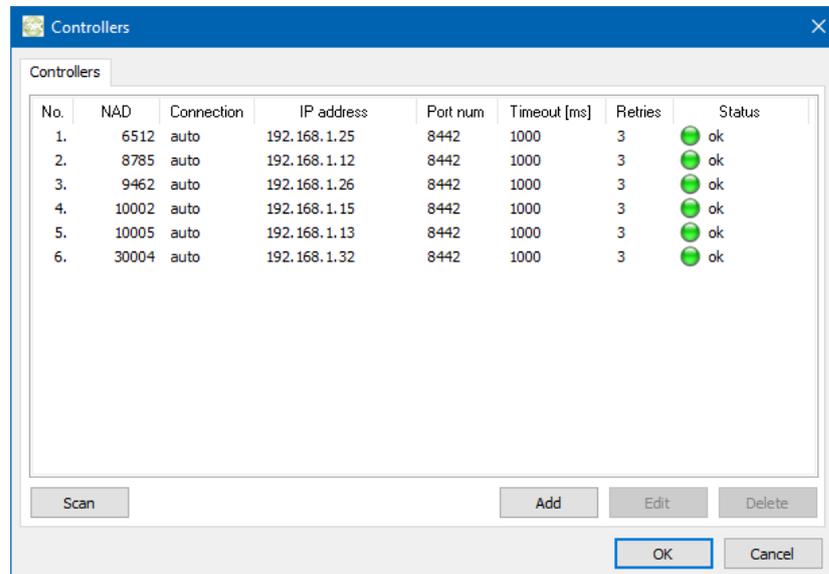
- small up to 100 tags
- medium up to 1000 tags
- large unlimited number of tags

Using the OPC server with 10 or less tags is free, no activation code is needed. There is no limit to number of reads or writes.

Server can be used for development and testing without activation code. After 2 hours, warning message will pop up and server will stop. Server can be restarted manually. Number of restarts is not limited.

Controllers

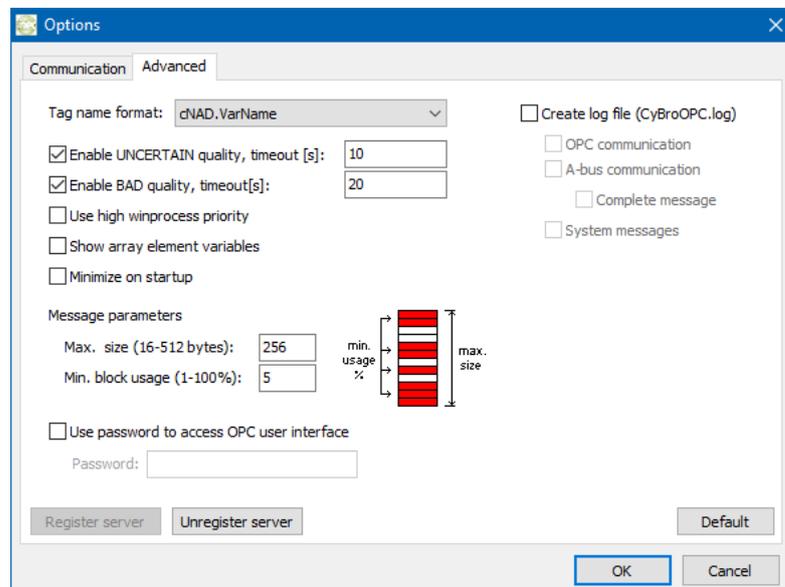
To create list of controllers that will be used with server, use [Scan](#) or enter them manually.



Status column display result of last scan. Sort order is retained in main window.

Options

Use Options dialog to set communication options and advanced details.



Default tag format is cNAD.VarName. Other formats are for compatibility with older versions and should not be used for new projects.

High process priority may slightly improve performance when server is handling large number of controllers. Log file is used for debugging. It may significantly affect server performance. File size is not limited.

Toolbar



Start/stop (Ctrl-D) Open TCP/IP socket and start communication.



Refresh tag (F9) Refresh selected tags. Refresh is performed by setting read request on selected tags. Tags will be read in next communication cycle.



Refresh all (F10) Read all tags from current controller. Refresh is performed by setting read request on all tags.



Set value (Enter) Write value to selected tags. It is performed by setting write request on selected tags. Value will be written in next communication cycle.



Add to Mon (Ins) Add selected tags to monitor. Monitor is a small OPC client, for both controller and system variables. Unlike background refresh, monitor affects tag status and communication statistics.



Controllers (F5) Open list of available controllers.



Options (F4) Open program settings dialog.

Status panel

Server status



Active, communication between server and all controllers is up and running.



Error, socket binding failed or at least one controller is not responding.



Idle, no communication. No controllers configured, or no read/write requests.



Stopped. Press [Start](#) to open TCP/IP socket and restart communication.

Server status is visible in [Sys.Status](#) tag. For more details, use [Sys.OpcStatus](#) tag.

OPC interface



Active, at least one client connected.



Idle, no clients connected.

Network monitor

Controller status



Active, communication is up and running, no errors detected.



Idle, no active read/write requests.



Communication error.

Controller status is visible in [cXXXX.Sys.Status](#) tag. For more details, use [cXXXX.Sys.PlcStatus](#).

Animation



Read operation is currently executed.



Write operation is currently executed.



A request from OPC client is pending.

Tag status



Read request pending, tag will be updated in next communication cycle.



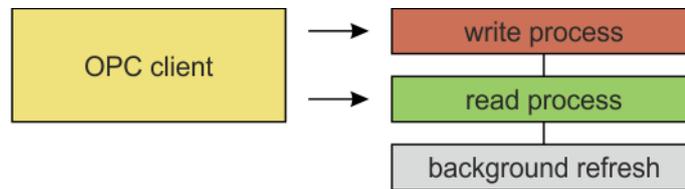
Write request pending, tag will be written in next communication cycle. Tag quality becomes LOCAL_OVERRIDE until next read is performed.



No requests pending, tag will be refreshed with next background refresh. When communication is fully utilized, background refresh is stopped, idle tags are not updated. Tag quality drops to LAST_KNOWN.

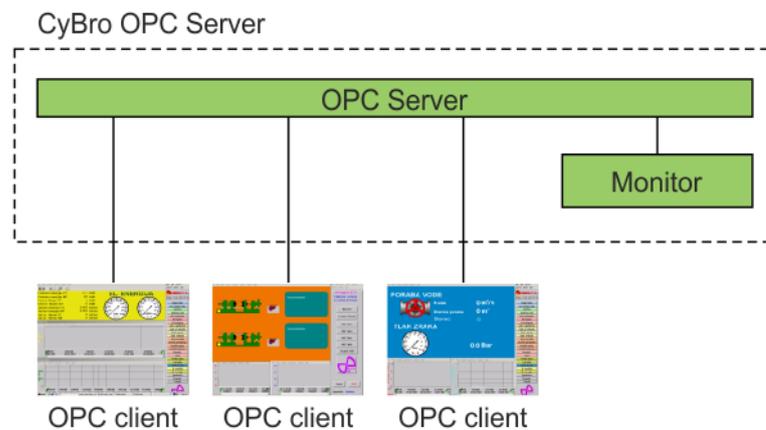
Process priority

OPC server execute three separate processes: read, write, and background refresh. Write process is top priority, read process is lower, and background refresh is lowest priority.



Read and write process is started by request from OPC client. When no requests are pending, background refresh is processed. Heavy traffic may temporary stop background refresh.

OPC data monitor is a local OPC client. Monitor activity affect system tags.



Tag quality

According to OPC specification, each tag has **value** and **quality**.

Value is an actual tag value, quantity. Range of possible values depends on tag type (bit, integer, long, real).

Quality is a property independent of value, and specifies how reliable the actual value is. Quality depends on how and when the value is obtained. When communication is uninterrupted, quality is always "Good". If communication channel is broken, quality first become "Uncertain", then "Bad".

A range of possible qualities is defined by OPC specification. There are three main categories, "Good", "Uncertain" and "Bad. Each category can contain additional info field, which may give a more detailed explanation about the problem cause.

CyBro OPC server implements the following qualities:

Good.....Value is updated regularly.

Good, local overrideValue is locally overridden using function "Set value". Tag status is red (write pending). After the write command is completed, status becomes green (reading) or gray (idle). When tag value is read next time, quality changes back to "Good".

Uncertain, last usable First timeout, last updated value is more than 10 seconds old.
Possible reason may be a slower network update time, or communication errors. This is a warning, not error.

Bad Value is unusable, no successful read was performed since OPC server started.

Bad, last known Second timeout, last updated value is more than 20 seconds old.
Communication is broken, value is not reliable any more.

Bad, out of service Activation code expired. More tags than allowed by license is used,
development timeout (2 hours) expired.

To adjust "Uncertain" and "Bad" timeouts, open [Options/Advanced](#).

System tags

System tags are virtual tags created by OPC server, used to show information about server and network. Two types are available: those belonging to server, and those belonging to controller.

Some tags are resettable, writing zero (or reset) will clear the tag.

To get more details about system tags, check tag description.

The screenshot shows the CyBro OPC Server v3.0.8 interface. The main window displays a list of system tags. The tags are organized into a tree view on the left, with 'CyBro OPC Server (running)' expanded to show 'Monitor (23)'. The main table lists various system tags with their respective types, timestamps, quality, values, and descriptions. The tags include system status, server information, license details, and communication statistics.

Name	Type	Timestamp	Quality	Value	Description
Sys.Status	string	13.3.2019. 21:43:33	Good	ok	Server status: "disconnected", "idle", "error", "ok".
Sys.OpcStatus	string	13.3.2019. 21:43:33	Good	ok	Server status details: "disconnected", "idle: no communication", "idle: no communication".
Sys.ServerStartedAt	string	13.3.2019. 21:43:33	Good	13.3.2019. 19:30:27	Time and date when OPC server was started.
Sys.SystemTime	string	13.3.2019. 21:43:33	Good	13.3.2019. 21:43:33	System time.
Sys.ServerUptime	string	13.3.2019. 21:43:33	Good	02:13:06	Time elapsed since OPC server was started.
Sys.LicenseType	string	13.3.2019. 21:43:33	Good	large	License type: "not licensed" (<10 tags), "small" (<100 tags), "medium" (<1000 tags), "large" (>1000 tags).
Sys.LicenseStatus	string	13.3.2019. 21:43:33	Good	ok	License status: "ok", "xxx min left", "expired".
Sys.TagLimit	string	13.3.2019. 21:43:33	Good	40000000	Total number of monitored tags allowed by license.
Sys.ClientsConnected	long	13.3.2019. 21:43:33	Good	1	Number of connected OPC clients.
Sys.TagCount	long	13.3.2019. 21:43:33	Good	2460	Total number of tags.
Sys.MonitoredTags	long	13.3.2019. 21:43:33	Good	139	Total number of tags monitored by all connected OPC clients.
Sys.PlcCount	long	13.3.2019. 21:43:33	Good	6	Total number of controllers in all projects.
Sys.ComReceiveCount	long	13.3.2019. 21:43:33	Good	201284	Total number of good communication messages received.
Sys.ComTransmitCount	long	13.3.2019. 21:43:33	Good	201681	Total number of communication messages sent.
Sys.ComErrorCount	long	13.3.2019. 21:43:33	Good	0	Total number of communication errors (including timeouts, bad messages).
Sys.LastReceivedAt	string	13.3.2019. 21:43:33	Good	13.3.2019. 21:43:33	Timestamp of last good communication message.
Sys.LastReceivedNad	long	13.3.2019. 21:43:33	Good	6512	Controller from which last good communication message is received.
Sys.LastResponseTime	string	13.3.2019. 21:43:33	Good	-	Response time of last good communication cycle.
Sys.MaximumResponseTime	string	13.3.2019. 21:43:33	Good	202 ms	Maximum response time encountered.
Sys.MaximumResponseTimeAt	string	13.3.2019. 21:43:33	Good	13.3.2019. 21:41:09	Timestamp when maximum response time is encountered.
Sys.MaximumResponseTimeNad	long	13.3.2019. 21:43:33	Good	8785	Controller for which maximum response time is encountered.
Sys.ComTimeoutCount	long	13.3.2019. 21:43:33	Good	0	Total number of communication timeouts.
Sys.LastTimeoutAt	string	13.3.2019. 21:43:33	Good	13.3.2019. 19:43:58	Timestamp of last detected OPC communication timeout.
Sys.LastTimeoutNad	long	13.3.2019. 21:43:33	Good	99999	Last controller that caused communication timeout.
Sys.BadMessageCount	long	13.3.2019. 21:43:33	Good	0	Total number of bad communication messages received.
Sys.LastBadMessageAt	string	13.3.2019. 21:43:33	Good	-	Timestamp of last detected bad communication message.
Sys.LastBadMessageNad	long	13.3.2019. 21:43:33	Good	0	Last controller from which bad communication message is received.
Sys.NegativeAckCount	long	13.3.2019. 21:43:33	Good	0	Total number of negative acknowledges received.
Sys.LastNegativeAckAt	string	13.3.2019. 21:43:33	Good	-	Timestamp of last detected negative acknowledge.
Sys.LastNegativeAckNad	long	13.3.2019. 21:43:33	Good	0	Last controller from which negative acknowledge is received.
Sys.TagsToRead	long	13.3.2019. 21:43:33	Good	81	Total number of tags waiting to be read (excluding background refresh).
Sys.TagsToWrite	long	13.3.2019. 21:43:33	Good	0	Total number of tags waiting to be written.
Sys.ReadCount	long	13.3.2019. 21:43:33	Good	145227	Number of OPC read cycles (excluding background refresh).
Sys.LastReadAt	string	13.3.2019. 21:43:33	Good	13.3.2019. 21:43:33	Timestamp of last successful OPC read (excluding background refresh).
Sys.LastReadNad	long	13.3.2019. 21:43:33	Good	10002	Controller accessed in last successful OPC read (excluding background refresh).
Sys.ReadCycleCount	long	13.3.2019. 21:43:33	Good	0	Number of OPC read cycles (excluding background refresh).

Click column name to sort tags. To restore default tag order, click Status column.

DCOM setup

OPC 2.0 technology uses Microsoft's COM/DCOM model to exchange data between a client and a server, so DCOM permissions must be set to allow communication between objects on different computers.

As a prerequisite, latest version of OPC Core Components Redistributables must be installed (included in CyBroOpcServer instalation).

Mutual user accounts

To ensure a successful communication between OPC client and server computers, it is necessary to setup same user accounts on both computers. There are two things to note:

- user account must have a password
- user account must have the same username/password on both computers

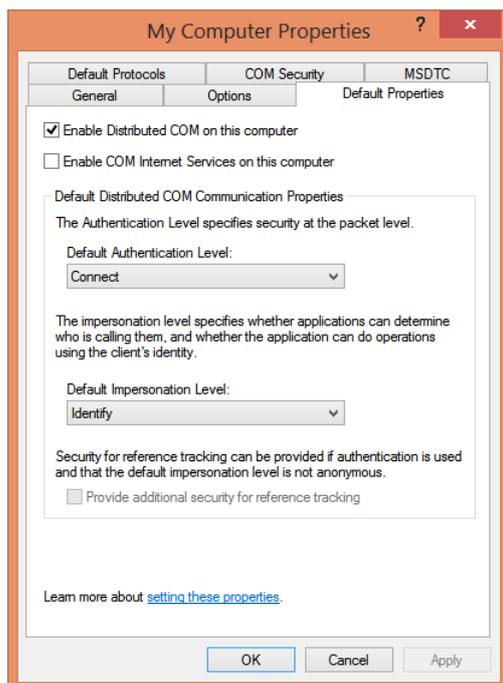
On Windows 7 and later it is also necessary to set the local security policies. Go to Control Panel / Administrative Tools / Local Security Policy, or open "secpol.msc". Next, navigate to Security Settings / Local Policies / Security Options and find the "Network access: Sharing and security model for local accounts" option and set to "Classic - local users authenticate as themselves".

System-wide DCOM settings

Open "dcomcnfg.exe", navigate to Component Services / Computers, right-click on My Computer and select Properties.

On the Default Properties tab:

1. Set "Enable Distributed COM on this computer"
2. Set Default Authentication Level to "Connect"
3. Set Default Impersonation Level to "Identify"



On the COM Security tab:

1. Under Access Permissions, click Edit Default and add the following Group or user names:

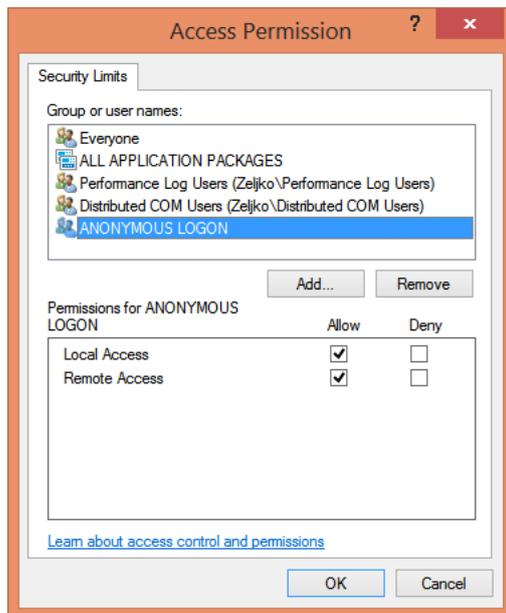
Anonymous Logon

Everyone
Interactive
Network
System

2. Ensure that both Local and Remote Access are allowed for all groups/users above.

3. Do the same for Edit Limits option (if the button is not disabled).

4. Repeat the above three steps for Launch and Activation Permissions.



Specific DCOM settings

Open "dcomcnfg.exe", navigate to Component Services / Computers / My Computer / DCOM Config. Find OPCEnum or OPC server in the list, right-click and select Properties.

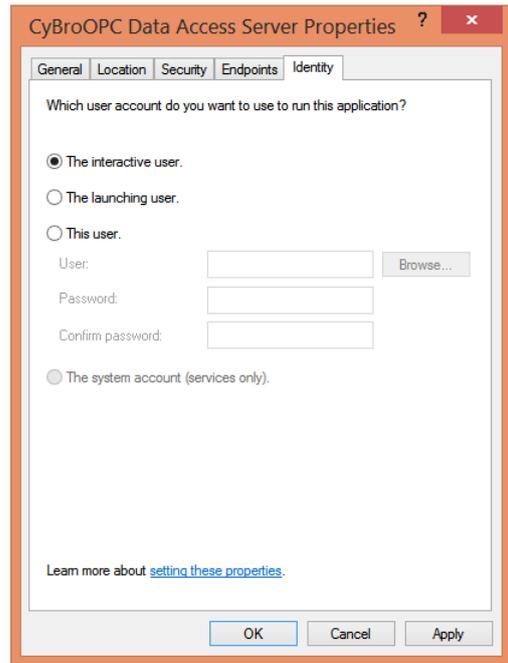
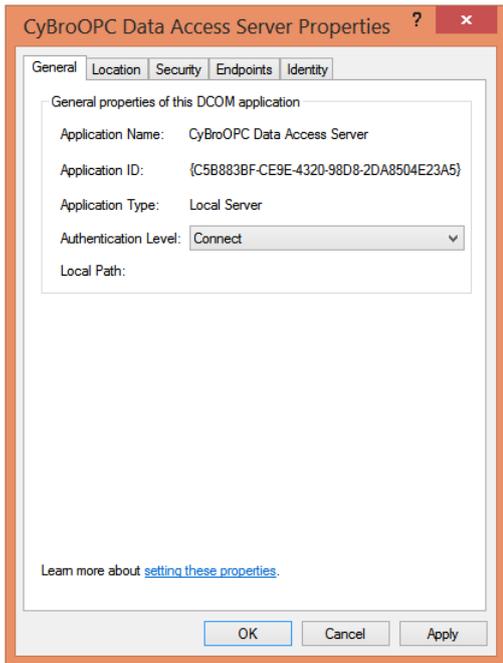
1. On the General tab, set Authentication Level to "Connect".

2. On the Security tab, under "Launch and Activation Permissions", select Customize, Edit. Add the following users and ensure that all permissions are allowed for them:

Everyone
Interactive
Network
System

3. Repeat the procedure for "Access Permissions".

4. On the Identity tab, select the user under which your OPC server will run (in case of OPCEnum, set it to "The system account"). Since CyBroOPCServer wasn't developed to run as a service, it should be set to Interactive user or This user. If Interactive user is selected, it is necessary to remain logged on at the computer in order for the OPC server to run.



Firewall exception rules

To enable successful communication with the OPCenum and OPC Server from the remote computer, they should be added to the firewall exception list. This task is specific to the firewall used, so it is not covered here.

Troubleshooting

If you does not succeed in connecting to the remote OPC server, even after DCOM permissions are configured, there is a troubleshooting guide available at OPC Training Institute. Also you can find a small utility, OPC Rescue, which will help to identify error cause.

Keyboard shortcuts

Ctrl-A	Select all tags
Insert.....	Add selected tags to monitor (controller only)
Delete	Remove selected tags from monitor (monitor only)
Enter	Set tag value
Space	Toggle tag value (bit only)
Ctrl-Up/Down	Move selected tags up/down (monitor only)
F4	Options dialog
F5	Controllers dialog
F9	Refresh selected tags
F10	Refresh all PLC tags / all monitor tags
Alt-F4	Close OPC window
Ctrl-Alt-F4	Shutdown OPC server

Technical specifications

Server IDCyProOPC.DA2
OPC version 1.0, 1.0a and 2.0
OPC interfacesynchronous and asynchronous
Supported OS.....Win7, Win10
Supported controllersCybro-2 and Cybro-3
Communication media.....Ethernet/LAN
Communication protocol.....A-bus symbolic (variables read directly from PLC)