OPC Unified Architecture Data Access Client

OPC Unified Architecture Data Access Client communication protocol

Protocol specification

The OPC UA protocol is other generation of OPC standard, which provides the cohesive, secure and reliable platform-independent framework for access to the real time data. The main difference in comparison with the previous versions is that the interprocess communication uses TCP/IP communication instead of COM/DCOM technology. This protocol supports two methods of data encoding (binary and XML). The existing OPC client implementation supports only binary encoding.

Basic concepts

Endpoint: a physical address on a network, which allows the client to access to one or more of the services provided by the server.

Session: it is an abstract connection of OPC UA server and a client on OSI application layer.

Monitored item: an entity on OPC UA server that is defined by the client. It is used for monitoring the values.

Subscription: an object on OPC UA server that is defined by the client. It returns the notifications about change of monitored items.

Initiation of communication

When initialing the communication, the client changes the messages with the server. In binary encoding, **"Hello Message"** is the first message sent from the client to server. The message defines the size of receiving and sending buffers and maximum size of messages that may be swapped during TCP communication between client and server. It also defines URL address of endpoint. The server will send **"Acknowledge message"**, in which it will confirm the suggested parameters or modify them according to its limits.

The client will send next message **"OpenSecureChannel message"** . It helps to establish the communication channel to swap data. In this message, the client and server will agree what type of encrypting mode will be used (either "sign and encrypt" or "decrypt" only). OPCUA client in D2000 System supports only decrypted mode.

After establishing the communication channel, the client can send the message to create a session, **"CreateSession Message"**. It is the connection on OSI application layer. After the server confirms the request, the session must be activated by message **"ActivateSession Message"**. In this message, the client will agree the algorithm for signing and encoding with the server but only if the modes have been defined when establishing the communication channel.

After activating the session, all types of messages that are intended for object management in the address space of OPC UA server can be swapped. In this step the client creates a subscription with the parameters set on the communication station level for all stations within the communication line. The monitored items should be then inserted to these subscriptions. They correspond with the instances of I/O tags, which contain the parameters defined in the address dialog window.

From this moment, the server informs the client about the changes on the monitored objects by **"Publish message"** in the periodic intervals (that have been set in the parameters of subscription). If the monitored objects have not been changed, the server will send **"Publish message"** once in (Max KeepAlive Count * Publishing Interval) seconds. The message informs the client that the subscription is still active. A similar check mechanism is also on the client side - it will send **"Publish message"** once every (Max KeepAlive Count * Publishing Interval). If the client does not send the acknowledgement **"Publish message"** within (LifeTime Count * Publishing Interval), the subscription will expire on the server side.

Communication line configuration

Communication line category: OPC UA Client
Host address: OPC UA server address. You may set the name according to UNC convention (e.g. "\\server" or "server", DNS names (e.g. "domain.com", "example.company.com") or IP address ("196.54.23.113").
TCP port: TCP port of OPC UA server (e.g. 4840).
EndpointUrl: Endpoint address (e.g. *opc.tcp://localhost:4840*)
Encoding type: Type of encoding that is used at dat exchange (currently only *Binary encoding* is supported).

Protocol configuration on the communication line

Parameter name	Meaning	Unit	Default value
Session Name	Session text identifier. Session identifier should be a unique within the client instance, making it possible to search problems faster in the client or server logs.	String	Kom process
Requested Channel Lifetime	The channel must be reopened before this time limit elapses. If the time is exceeded, the channel will be closed and unable to change data.	hh:mm:ss	01:00:00
Requested Session Timeout	Any message should be changed between client and server before this time limit elapses. If it is not sent, the sources within the session that are kept on the server are released. The primary work of this parameter is to remove the sessions that became inactive because of some unexpected reason.	mm:ss	01:00
Authentication Type	Type of autentication used with OPC UA server. Supported types are: • Anonymous: logon is anonymous • Username : logon uses user name and password	Anonymous / Username	Anonymous
Token User Name	User name used in authentication if Authentication type = Username.		
Token Password	Password used in authentication if Authentication type = Username.		
Debug Mode	It changes the number of information about the communication. We recommend to enable the mode Extended/Full only when detecting the problems and debugging the communication.	Normal /Extended/Full	Normal
Debug Threads	Parameter defines the thread(s) that will send the debug info about the communication.	Receiving /Sending/Others treads/All threads	All threads

Protocol configuration on the communication station

The parameters on the level of communication station correspond with the setting of one subscription. It means the one communication station is equivalent to one instance of subscription within session.

Full name	Description	Unit	Default value
Requested Publishing Interval	Defines the time interval for server to send the information about the change of monitored items within the instance subscription by **"Publish message"**. Note: This parameter defines a proposed value which the OPC UA server can change, e.g. Bernecker-Rainer always returned a value of "Publishing Interval" at least 50 ms, although the requested interval was smaller.	mi:ss.mss	00:05.000
Requested LifeTime Count	If the client does not send the request for data till the time defined by (LifeTime Count * Publishing Interval), the subscription expires. The value should be minimally 3 times higher than the "Requested Max KeepAlive Count". Note: This parameter defines a proposed value which the OPC UA server can change, e.g. Bernecker-Rainer always returned as a value of "LifeTime Count" a maximum of 600, although the requested value was greater.	Number	1000
Requested Max KeepAlive Count	If the objects of subscription are not changed, the server will send keep-alive message after elapsing the time (Max Notifications Per Publish * Publishing Interval). The client will confirm this message when it sends a new request for data. Note: This parameter defines a proposed value which the OPC UA server can change, e.g. Bernecker-Rainer has always returned as a value of "Max KeepAlive Count" a maximum of 200, although the requested value was greater.	Number	5
Max Notifications Per Publish	Parameter defines the maximum number of notifications about the object change, which the server can send in one **"Publish message"**. Zero indicates that the number of notifications is unlimited.	Number	0
Publishing Enabled	Parameter enables/disables the publishing within the subscription.	YES /NO	0
Priority	It defines a relative priority of one subscription. If server should send more notifications, the subscription with higher priority is preferred.	0-255	0
Samples Queue Size	This parameter enables to create the object queue with the defined length on OPC UA server's side for each monitored item in subscription.	Number	0

I/O tag configuration

I/O tag configuration dialog window is used for setting the monitored objects.

Object address setting

Name	Meaning	Unit	Default value
ID	The identifier in text format, which is, in dependence on **ID type,** converted to the required native type.	String	
ID type	Enumerated types of identifiers. They help to access to the the objects in OPC UA address space. *Numeric-1B ID*: Identifier limited to 1-byte value (0-255) *Numeric-2B ID*: Identifier limited to 2-byte value (0-65535) *Numeric-4B ID*: 4-byte identifier *String*: Text identifier *Guid -16B ID*: 16-byte (128-bit) number that is usually divided into four parts. For example 3F2504E0-4F89-11D3-9A0C-0305E82C3301. *ByteString*: Identifier that is represented as a sequence of bytes.	Numeric-1B ID / Numeric-2B ID/ Numeric-4B ID/String/Guid -16B ID /ByteString	Undefined
Namespace	Numerical identifier of name space of OPC UA server. Each OPC UA server can have N name spaces. However, the object identifier must be unique in one name space.	Numeric	
Variable type	Value type of objects that can be processed by OPC UA client. *Variable type* should be used only if I/O tag is intended for writing. As regards the reading of the object value, the information about type is sent together with the value.	Undefined / Boolean / Byte / SByte / Integer16 / Unsigned16 / Integer32 / Unsigned32 / Integer64 / Unsigned64 / Float / Double / String / UTC Time / Boolean array / Byte array / SByte array / Integer16 array / Unsigned16 array / Integer32 array / Unsigned32 array / Integer64 array / Unsigned64 array / Float array / Double array / String array / UTC Time array	Undefined

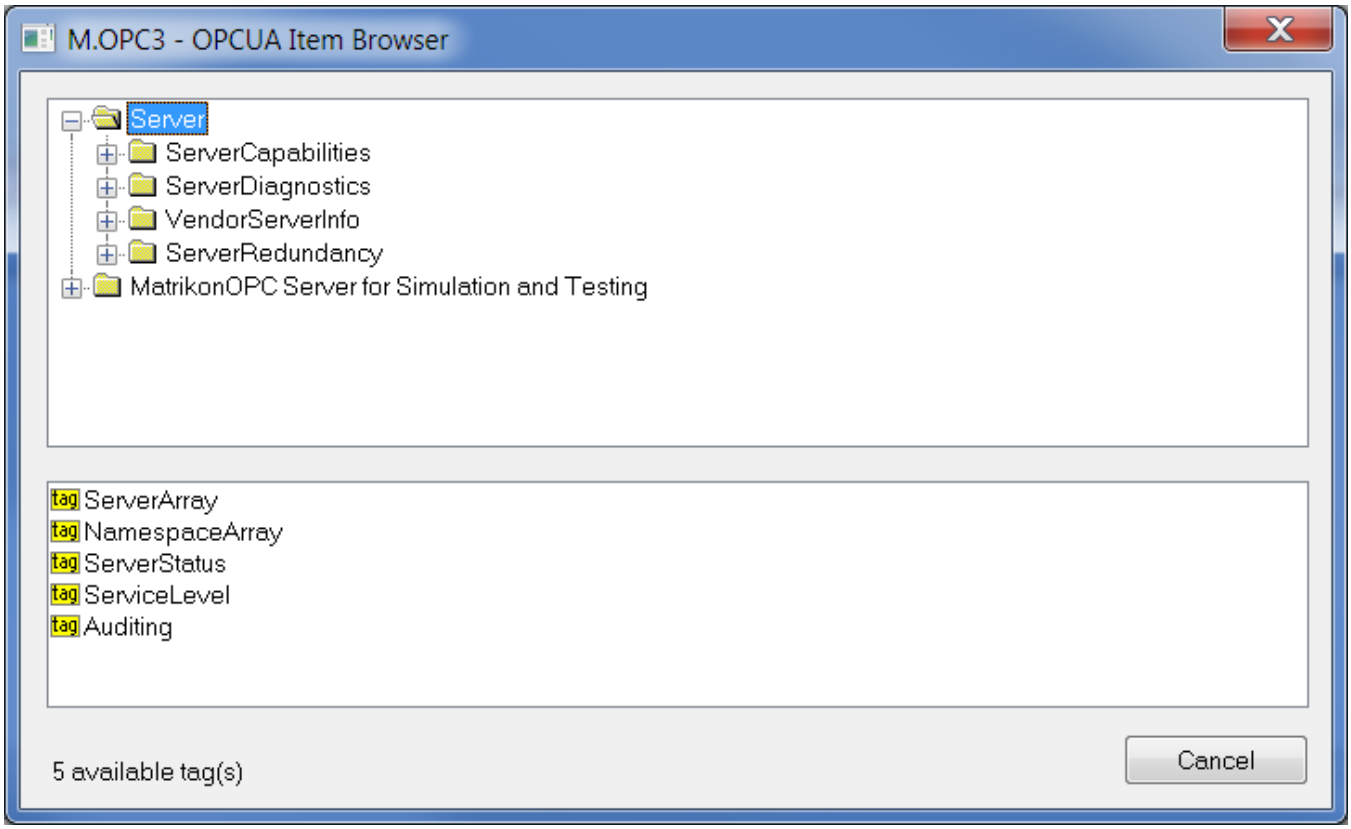
Array index	If the object value is represented as a value array *(Boolean array / Byte array / SByte array / Integer16 array / Unsigned16 array / Integer32 array / Unsigned32 array / Integer64 array / Unsigned64 array / Float array / Double array / String array / UTC Time array)*, the parameter defines its range or value of a particular item. A text representation of array index may be in several formats: • Separate integer, e.g. **"6"** - when you want to obtain only one value from the array. • Two integers separated by a colon, e.g. **"6:7"**, - if you want to obtain the range of values. • The expression separated by comma in case of multidimensional array, e.g. **"6,7"** - when you want to obtain the particular value of the item on 2D array. If you want to define the range, you should use the expression separated by semi-colon, e.g. **"6:8,7:10"**.	String	
Write only	It sets if I/O tag is a part of subscription. Its value will be sent periodically from the server in "Publish message".	Unchecked/checked	Unchecked
Expanded Node ID	If it is checked, it enables to address the ExpandedNodeId. Unlike the classic identifier in the OPC UA address space, ExpandedNodeId is supplemented by **NameSpace URI** and **Server index.**	Unchecked/checked	Unchecked
NamespaceUri	Text identifier of name space of OPC UA server that is used instead of the numerical representation of namespace.	String	
ServerIndex	Numerical identifier that address the server number when using the ExpandedNodeID identifier.	Numeric	0

Settings of other parameters

Name	Meaning	Unit	Default value
Sampling type	Parameter defines a sampling frequency of monitored object. When using "Publishing rate", the frequency is equivalent to time Requested Publishing Interval, which is set on the communication station level. "Practical fastest rate" sets the sampling frequency on the maximum value. "Custom rate" enables to specify the custom sampling interval, which may be defined in "Sampling Time".	Publishing rate /Practical fastest rate /Custom rate	Publishing rate
Sampling time	Parameter allows you to set the custom sampling frequency if "Sampling type" is "Custom rate".	ss.ms	0.0
DeadBand type	Deadband is a band in which the change of value does not cause Data Change Notification, which is the part of Publish Message. When using "None", this band is ignored. Otherwise, there is used the relative or absolute value ("Percent"/" Absolute") from "DeadBand value".	None/Absolute /Percent	None
DeadBand value	Parameter defines the custom value of deadband if you chose the relative/absolute value ("Percent"/"Absolute").		0.0
Trigger type	Parameter specifies the condition which causes Data Change Notification. When using "Status", only the status change is reported. Change of value and time stamp is ignored. When using "Status,Value", the change of time stamp is ignored. "Status,Value,Timestamp" ensures the reporting in all options, i.e. when changing the status, value or time stamp.	Status/Status, Value/Status, Value, Timestamp	Status, Value, Timestamp

Browser dialog window

This dialog window is intended for browsing and inserting the OPC UA objects into the address parameter of I/O tag. The upper part contains the tree structure of address space. When clicking on the object, the lower part of window displays the direct descendents of the object.
Double click on one of the descendents transfers the address parameters of object to the address dialog window of I/O tag.



Literature

OPC Foundation manuals are placed on http://www.opcfoundation.org/.

- OPC UA Part 1 - Overview and Concepts 1.01 Specification
- OPC UA Part 2 - Security Model 1.01 Specification
- OPC UA Part 3 - Address Space Model 1.01 Specification
- OPC UA Part 4 - Services 1.01 Specification
- OPC UA Part 5 - Information Model 1.01 Specification
- OPC UA Part 6 - Mappings 1.00 Specification
- OPC UA Part 7 - Profiles 1.00 Specification
- OPC UA Part 8 - Data Access 1.01 Specification

Changes and modifications

- May 10, 2012 - creating the document

Document revisions

- Ver. 1.0 – May 10, 2012

Related pages:

Communication protocols