D2000 OPC UA Server

OPC Unified Architecture (OPC UA) is a protocol for industrial automation. This protocol, which is managed by OPC Foundation, is a successor of the successful and often used OPC (OPC DA or OPC Classic) protocol. Unlike its predecessor, it is not based on Windows technologies (OLE, COM), thus available also on other platforms (e.g. PLC Simatic, or Bernecker & Rainer).

D2000 OPC UA Server allows OPC UA third-party clients to access the objects of the D2000 system - to read them and to write their values.

D2000 OPC UA Server is located in the installation directory of D2000 installation under the name "opcuaserver.exe" (i.e. "opcuaserver" on Linux and Raspbian).

Characteristics of D2000 OPC UA Server

Support of **opc.tcp://** protocol.

Support for multiple OPC UA clients connected in parallel (multiserver).

Addressing of D2000 objects: numeric ID (object HOBJ, in the *D2000* branch) or text (object name, in the *D2000N* branch).

Value types on the side of the D2000 OPC UA server:

- Boolean (objects of Line type, values of boolean type - De/Di/Dout)
- DateTime (values of absolute time type - TmA/TiA/ToA)
- Double (values of real type - Re/Ai/Ao, relative time - TmR/TiR/ToR)
- Int32 (objects of Station/Alarm/Process types, values of type quadstate - Qi)
- Int64 (values of integer type - Int/Ci/Co) *
- String (values of text type - Txt/TxtI/TxtO)

Note: In version D2000 21.1.72, the Int type was changed from 32 to 64 bits, which results in the representation of Int/Ci/Co values as Int64 (formerly Int32). Therefore, new start parameters were supported, which can be used to change the behaviour of the D2000 OPC UA Server as follows:

- *--INT32INV* values of type Int/Ci/Co will be represented as Int32 in the D2000 OPC UA server, values outside the range of Int32 will be invalidated
- *--INT32SAT* values of type Int/Ci/Co will be represented as Int32 in the D2000 OPC UA server, values outside the range Int32 will be saturated

Support of identities:

- ANONYMOUS
- name:password

Support of security policies:

- None
- Basic128Rsa15
- Basic256
- Basic256Sha256

Message Security Modes:

- None
- Sign
- Sign&Encrypt

Configuration of a user for D2000 OPC UA Server

For D2000 OPC UA Server to access the individual objects of the D2000 system, it is necessary to create a user in the D2000 system under which the OPC UA Server logs in to D2000 Server. OPC UA Server receives access rights of this user. User name must be in "OPCUA_User_<*process_name_opcua*>" format. For instance, if OPC UA Server is named "SELF.OUS" (default, the process name can be changed with /W switch), then the name of the relevant user will be "OPCUA_User_SELF". It is necessary to set this user's access rights to objects of the D2000 system. These access rights will be monitored while reading/writing values by the OPC client.

Configuration of D2000 OPC UA Server

OPC UA Server configuration is read from a file. It is vital to specify the path to a configuration file by the starting parameter --cfg=<*path_to_configuration_file*>, for example "`opcuaserver.exe --cfg=c:\D2000\D2000_APP\application1\opcuaserver\opcuaserver.conf`". Sample configuration file is located in the program directory in a subdirectory `Templates\opcuaserver\opcuaserver.conf.in` (resp. `.sys\templates\opcuaserver\opcuaserver.conf.in` on Linux). In this file, some parameters are already preset. It is necessary to set at least the `pki_dir` parameter and create a directory structure for PKI.

It is possible to specify the following parameters in the configuration file:

Parameter	Value
application_name	name of the application

application_uri	URI applications
pki_dir	full path to PKI directory structure (e.g. `'c:\D2000\D2000_APP\application1\opcuaserver\pki'`)
tcp_config.host	the address of the network adapter on which the OPC UA Server accepts connections (*0.0.0.0* for all network adapters)
tcp_config.port	the port on which OPC UA Server accepts connections
user_tokens	the list of configured users under which OPC UA clients can log in
endpoints	the list of access points of the OPC UA Server

The configuration file is read during the OPC UA Server startup, so the adjustments of parameters in the file will show only after a restart. If the PKI directory structure does not exist, the OPC UA Server creates it (empty, without keys and certificates), based on the settings of `pki_dir` parameter.

Configuration of PKI (public key infrastructure)

For running a secure communication between OPC UA Server and the OPC UA client, it is necessary for OPC UA Server to create a PKI directory structure, private key, and a certificate.

The directory structure consists of the following directories:

directory name	description
`pki/`	PKI directory
`pki/private/`	directory with a private key of the OPC UA Server
`pki/own/`	directory with a public certificate of OPC UA Server
`pki/rejected/`	directory with a certificate of denied clients
`pki/trusted/`	directory with a certificate of allowed clients

It is essential to secure the private key against unauthorized access.

Private key generation and certificate signing requests using `openssl` utility :

```
openssl req -out csr.csr -new -newkey rsa:2048 -nodes -keyout pki/private/private.pem
```

Creation of self-signed certificate:

```
openssl x509 -req -days 365 -in csr.csr -signkey pki/private/private.pem -outform der -out pki/own/cert.der
```

Management of OPC UA Clients certificates

OPC UA Server sends its certificate to the OPC UA client during establishing a secured connection. When an unknown OPC UA client connects, OPC UA Server rejects the client and saves their certificate into "`pki/rejected/`" directory. After that, the administrator of the D2000 application has to manually move that certificate into "`pki/trusted/`" directory. This ensures that the server will consider the given client trustworthy and will accept the connection.

Management of OPC UA Clients names and passwords

Configuration of OPC UA clients' names and passwords is in the `opcuaserver.conf` configuration file. Only a single user token *sample_user* with user name *sample* and password *sample1* is predefined:

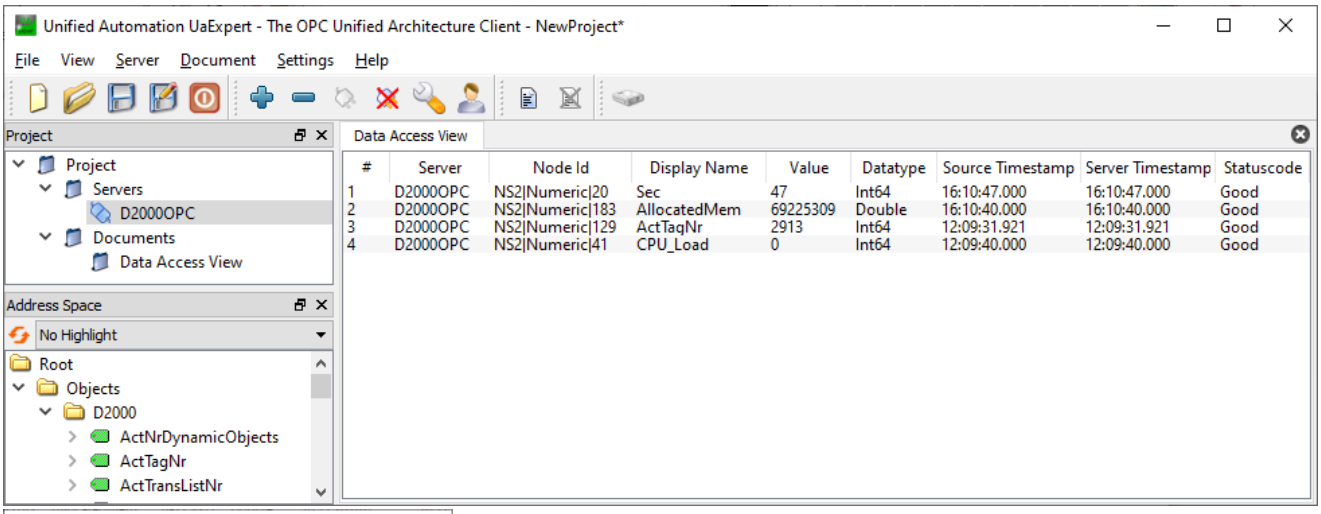
```
user_tokens:
  sample_user:
    user: sample
    pass: sample1
```

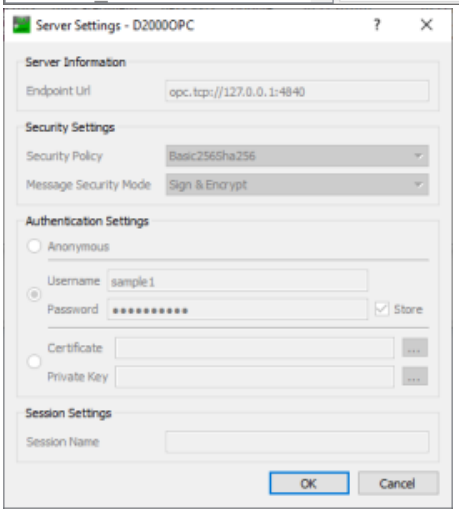
User tokens, as well as anonymous access (ANONYMOUS) permitted for individual endpoints, are defined in the definition of respective endpoints:

```
basic256sha256_sign_encrypt:
    path: /
    security_policy: Basic256Sha256
    security_mode: SignAndEncrypt
    security_level: 4
    user_token_ids:
      - ANONYMOUS
      - sample_user
```

Example configuration file (configuration contains absolute paths to PKI directory structure):

Configuration and connection of OPC UA client UaExpert by Unified Automation:


Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject*
File View Server Document Settings Help
Project
Project
Servers
D2000OPC
Documents
Data Access View
Address Space
No Highlight
Root
Objects
D2000
ActNrDynamicObjects
ActTagNr
ActTransListNr
Data Access View
Server Node Id Display Name Value Datatype Source Timestamp Server Timestamp Statuscode
1 D2000OPC NS2|Numeric|20 Sec 47 Int64 16:10:47.000 16:10:47.000 Good
2 D2000OPC NS2|Numeric|183 AllocatedMem 69225309 Double 16:10:40.000 16:10:40.000 Good
3 D2000OPC NS2|Numeric|129 ActTaqNr 2913 Int64 12:09:31.921 12:09:31.921 Good
4 D2000OPC NS2|Numeric|41 CPU_Load 0 Int64 12:09:40.000 12:09:40.000 Good


Server Settings - D2000OPC
Server Information
Endpoint Url
opc.tcp://127.0.0.1:4840
Security Settings
Security Policy
Basic256Sha256
Message Security Mode
Sign & Encrypt
Authentication Settings
Anonymous
Username
sample1
Password
Store
Certificate
Private Key
Session Settings
Session Name
OK
Cancel