

D2000 OPC UA Server

OPC Unified Architecture (OPC UA) je protokol pre priemyselnú automatizáciu. Tento protokol, ktorý spravuje organizácia OPC Foundation, je nástupcom úspešného aasto používaného protokolu OPC (OPC DA alebo OPC Classic). Na rozdiel od svojho predchodcu nie je postavený na Windows technológiách (OLE, COM) a je tak dostupný aj na iných platformách (napr. PLC Simatic, alebo Bernecker & Rainer).

D2000 OPC UA Server umožňuje OPC UA klientom tretích strán pristupovať k objektom systému D2000 - íta a zapisovať ich hodnoty.

D2000 OPC UA Server sa nachádza v programovom adresári D2000 inštalácie pod menom "opcua_server.exe" (resp. "opcua_server" na Linuxe a na Raspbiane).

Vlastnosti D2000 OPC UA Servera

podpora protokolu **opc.tcp://**

podpora viacerých paralelne pripojených OPC UA klientov (multiserver)

adresácia D2000 objektov: numerické ID (HOBX objektu, vo vetve D2000) alebo textové (meno objektu, vo vetve D2000N).

typy hodnôt na strane D2000 OPC UA servera:

- Boolean (objekty typu Linka, hodnoty typu boolean - De/Di/Dout)
- DateTime (hodnoty typu absolútny as - TmA/TiA/ToA)
- Double (hodnoty typu real - Re/Ai/Ao, relatívny as - TmR/TiR/ToR)
- Int32 (objekty typu Stanica/Alarm/Proces, hodnoty typu štvorstav - Qi)
- Int64 (hodnoty typu Int/Ci/Co) *
- String (hodnoty typu reazec - Txt/TxtI/TxtO)

Poznámka: vo verzii D2000 21.1.72 došlo k zmene Int typu z 32 na 64 bitov, oho dôsledkom je reprezentácia hodnôt typu Int/Ci/Co ako Int64 (predtým Int32).

Preto boli podporené štartovacie parametre, pomocou ktorých je možné zmeniť chovanie D2000 OPC UA Servera nasledovne:

- `--INT32INV` hodnoty typu Int/Ci/Co budú v D2000 OPC UA serveri reprezentované ako Int32, hodnot mimo rozsah Int32 budú zneplatnené
- `--INT32SAT` hodnoty typu Int/Ci/Co budú v D2000 OPC UA serveri reprezentované ako Int32, hodnot mimo rozsah Int32 budú saturované

podpora identít:

- ANONYMOUS
- meno:heslo

podpora bezpečnostných politík:

- None
- Basic128Rsa15
- Basic256
- Basic256Sha256

módy zabezpečenia správ (Message Security Modes):

- None
- Sign
- Sign&Encrypt

Konfigurácia užívateľa pre D2000 OPC UA Server

Aby D2000 OPC UA Server mohol pristupovať k jednotlivým objektom systému D2000, je potrebné vytvoriť v systéme D2000 užívateľa, pod ktorým sa OPC UA Server prihlási k serveru D2000. OPC UA Server dostane prístupové práva tohto užívateľa. Meno užívateľa musí byť vo formáte „OPCUA_User_<meno_procesu_opcua>“. Napríklad, ak je OPC UA Server nazvaný „SELF.OUS“ (default, meno procesu je možné zmeniť prepínaním /W), tak meno príslušného užívateľa bude „OPCUA_User_SELF“. Tomuto užívateľovi je potrebné nastaviť k objektom systému D2000 prístupové práva, ktoré budú kontrolované pri operáciách čítania/zápisu hodnôt OPC klientom.

Konfigurácia D2000 OPC UA Servera

Konfiguráciu OPC UA Server číta zo súboru. Cestu ku konfiguračnému súboru je nutné špecifikovať štartovacím parametrom `--cfg=<cesta_ku_konfiguracnemu_u_saboru>`, napríklad "opcua_server.exe --cfg=c:\D2000\D2000_APP\application1\opcua_server\opcua_server.conf". Vzorový konfiguračný súbor sa nachádza v [programovom adresári](#) v podadresári `Templates\opcua_server\opcua_server.conf.in` (resp. `sys\templates\opcua_server\opcua_server.conf.in` na Linuxe). V tomto súbore sú niektoré parametre už predvyplnené, je nutné nastaviť aspoň parameter `pk_i_dir` a vytvoriť adresárovú štruktúru pre [PKI](#).

V súbore je možné špecifikovať nasledujúce parametre:

Parameter	Hodnota
application_name	meno aplikácie

application_uri	URI aplikácie
pki_dir	plná cesta k adresáru PKI štruktúry (napríklad 'c:\D2000\D2000_APP\application1\opcuaserver\pki')
tcp_config.host	adresa sieového adaptéra na ktorom OPC UA server prijíma spojenia (0.0.0.0 pre všetky sieové adaptéry)
tcp_config.port	port na ktorom OPC UA server prijíma spojenia
user_tokens	zoznam nakonfigurovaných používateľov, pod ktorými sa môžu prihlasovať OPC UA klienti
endpoints	zoznam prístupových bodov OPC UA servera

Konfiguračný súbor je ťatý len pri štarte OPC UA servera, takže úpravy parametrov v súbore sa prejaví až po jeho reštarte. Ak adresárová štruktúra PKI neexistuje, OPC UA server ju vytvorí (prázdnu, bez kúov a certifikátov) podľa nastavenia parametra pki_dir.

Konfigurácia PKI (public key infrastructure)

Pre prevádzkovanie zabezpečenej komunikácie medzi OPC UA serverom a OPC UA klientami je potrebné pre OPC UA server vytvoriť PKI adresárovú štruktúru, súkromný kľúč a certifikát.

Adresárová štruktúra pozostáva z adresárov:

názov adresára	popis
pki/	adresár PKI
pki/private/	adresár s súkromným kľúčom OPC UA servera
pki/own/	adresár s verejným certifikátom OPC UA servera
pki/rejected/	adresár s certifikátmi zamietnutých klientov
pki/trusted/	adresár s certifikátmi povolených klientov

 Súkromný kľúč je nutné zabezpečiť proti neoprávnenému prístupu.

Generovanie súkromného kľúča a certificate signing request pomocou utility openssl:

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

Vytvorenie self-signed certifikátu:

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

Manažment certifikátov OPC UA klientov

Pri vytváraní zabezpečeného pripojenia OPC UA klient pošle OPC UA serveru svoj certifikát. Po pripojení neznámeho OPC UA klienta OPC UA server klienta odmietne a jeho certifikát uloží do adresára "pki/rejected/". Správca D2000 aplikácie následne musí manuálne presunúť daný certifikát do adresára "pki/trusted/". To zabezpečí, že daného klienta bude server považovať za dôveryhodného a spojenie prijme.

Manažment mien a hesiel OPC UA klientov

Konfigurácie mien a hesiel OPC UA klientov sú v konfiguračnom súbore `opcuaserver.conf`. Preddefinovaný je jediný užívateľský token `sample_user` s menom `sample` a heslom `sample1`:

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

V definíciách jednotlivých endpointov sú vymenované povolené užívateľské tokeny, prípadne je povolený aj anonymný prístup (ANONYMOUS):

Príklad konfiguraného súboru (konfigurácia obsahuje absolútne cesty k PKI adresárovej štruktúre):

