OPC UA server pro systém REXYGEN
Referenční manuál

REX Controls s.r.o.

Verze 3.0.4
Plzeň
27.3.2025

Obsah

Kapitola 1

OPC UA a systém REXYGEN

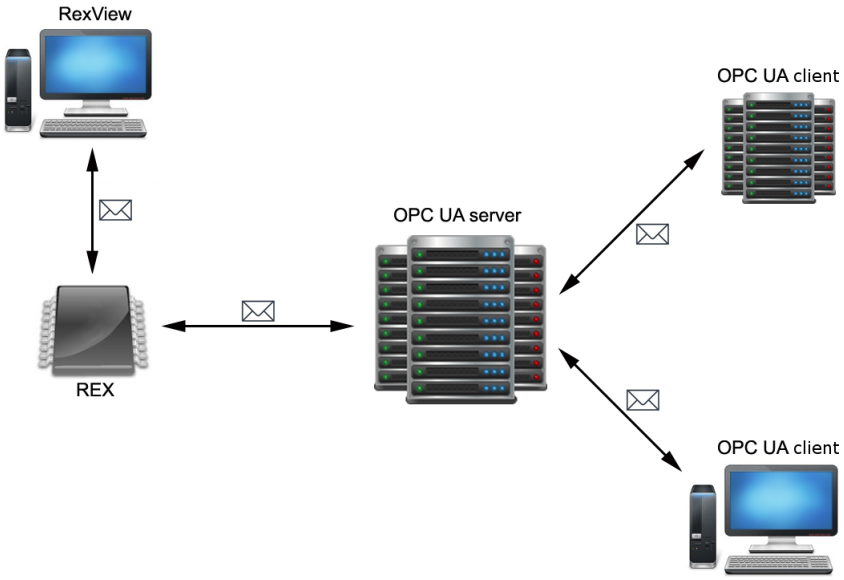
OPC UA je otevřený komunikační protokol určený pro průmyslovou automatizaci. Na rozdíl od klasického OPC je multiplatformní, může fungovat i jako webová služba a podporuje kromě přístupu k datům a událostí i další funkce jako volání metod, diagnostiku, různé stupně zabezpečení či autorizaci. OPC UA získává od svého vytvoření na oblibě a čím dál více firem ho používá ve svých výrobcích jako jedno z komunikačních rozhraní.

OPC UA není vhodné pro vytváření spojení mezi řídícími jednotkami v reálném čase, ale je použitelné pro téměř reálný čas. Jeho hlavní použití je však v propojení různých aplikací, ve vytváření "Internet of Things" a průmyslové revoluci 4.0.

1.1 Úvod

OPC UA server pro REXYGEN je samostatná aplikace, která komunikuje s exekutivou REXYGENu pomocí diagnostického protokolu. Není nutné, aby byl server na stejné výpočetní jednotce jako REXYGEN , ale je vhodné co nejvíce zkracovat jejich vzájemnou dobu odezvy. Je výhodnější, aby byl server z pohledu síťového připojení co nejblíže běžící instanci REXYGENu, než aby byl blízko ostatním OPC UA klientům. Server implementuje připojení pouze pomocí opc.tcp, což je běžná praxe u serverů, které shromažďují data z řídících jednotek běžících v reálném čase. Propojení serveru a ostatních zúčastněných aplikací je zobrazeno na obrázku 1.1.

Server získává informace o licencích přímo z exekutivy REXYGENu. Samotný OPC UA server běží neomezeně. Pokud je nějaká exekutiva příliš velká (má příliš mnoho proměnných), pak ji OPC UA server odmítne zobrazit. V tomto případě je nutné nainstalovat na cílové zařízení vyšší OPC UA licenci.



Obrázek 1.1: OPC UA server jako mezičlánek OPC UA klientů a REXYGENu

1.2 Funkce serveru

Běžící OPC UA server je připojen k exekutivě REXYGENu a ve svém adresním prostoru zobrazuje všechny její bloky i s proměnnými. Struktura Adresního prostoru je podobná struktuře tasků v diagnostice programu REXYGEN Studio. Po připojení k REXYGENu server vytvoří celou stromovou strukturu bloků a jejich parametrů a následně již pouze synchronizuje hodnoty jednotlivých parametrů. Ty jsou navíc synchronizovány pouze pokud je klient čte nebo do nich zapisuje. Tímto způsobem je možné se připojit i k několika exekutivám naráz.

Pokud je server odpojen, pokouší se opakovaně navázat spojení s exekutivou. Pokud se spojení ztratí a obnova se delší dobu nedaří, server znemožní klientům zápis do uzlů spojených s exekutivou a při jejich čtení poskytne poslední platnou hodnotu. Tento stav trvá až do opětovného připojení. Pokud dojde v REXYGENu k výměně exekutivy, server smaže a znovu nahraje strukturu bloků a vytvoří událost o změně Adresního prostoru.

1.3 Nastavení a spuštění serveru

Server je možné nastavit pomocí INI konfiguračního souboru, jehož umístění lze specifikovat pomocí parametru -c.

RexOpcUa [-c <configFile>]

V OS Windows je možné nastavit umístění standardního konfiguračního souboru pomocí příkazu s parametrem -i a cestou k novému konfiguračnímu souboru.

RexOpcUa -i <configFile>

V Linuxu je cesta ke standardnímu konfiguračnímu souboru pevně daná:

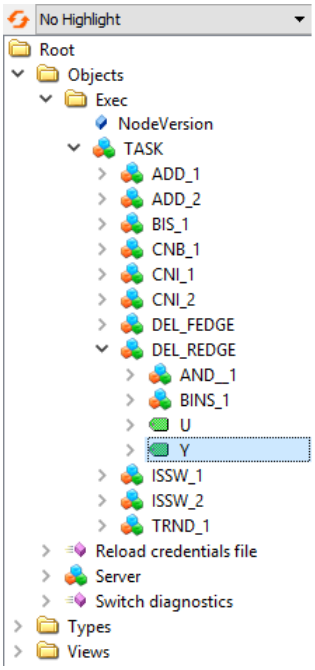
/rex/OpcUa/RexOpcUa.ini

Konfigurace je podrobně popsána v kapitole 3. Server lze spustit i jako službu (viz kapitolu 6.2). Návod na rychlé spuštění je popsán v kapitole 6.

Kapitola 2

Adresní prostor

Adresní prostor serveru obsahuje všechna data dostupná klientům. Některé uzly a jejich vazby jsou povinné pro všechny OPC UA servery, některé byly vytvořeny speciálně pro tento server. Addresní prostor obsahuje metody pro manipulaci se serverem a složky 'Exec', které obsahují stromovou strukturu úkolů REXYGENu i se subsystémy a bloky, a to včetně úkolů připojených k ovladačům. Vše kromě obsahu složek 'Exec' je vytvořeno při startu serveru. Obsah složky 'Exec' je vytvořen při novém připojení k REXYGENu nebo při přehrání exekutivy. Adresní prostor serveru připojeného k jedné exekutivě je zobrazen na obrázku 2.1 pomocí OPC UA klienta UaExpert (viz kapitola 6.3.1).



Obrázek 2.1: Adresní prostor zobrazený klientem UaExpert

Server používá několik vlastních jmenných prostorů (Namespace). První Namespace odpovídá URI aplikace (viz tabulka 3.3) a používá se pro chod serveru samotného. Namespace *urn:Rex:TypeDeclaration* se používá pro definici typů, kterými se popisují bloky a proměnné exekutivy. Namespace *urn:Rex:Server* obsahuje uzly, které slouží k obsluze serveru, například metody pro správu serveru. Namespace exekutivy je unikátní pro každou nahranou exekutivu nebo instanci REXYGENu a je popsán v kapitole 3.2. Namespace exekutivy obsahuje všechny uzly úkolů, bloků a parametrů dané exekutivy.

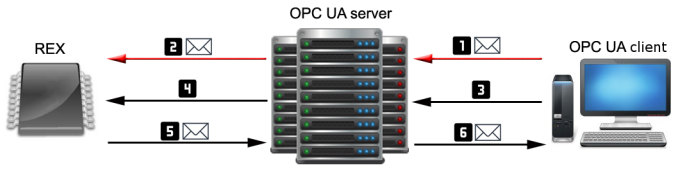
2.1 Bloky

Struktura bloků ve složce 'Exec' odpovídá struktuře bloků v exekutivě REXYGENu. Všechny bloky používají Namespace exekutivy (viz kapitola 3.2), jejich BrowseName a DisplayName odpovídá názvu v REXYGENu (u BrowseName je jako předpona uveden typ uzlu) a v popisu je uložen typ uzlu. Bloky obsahují proměnné, které jsou shodné s těmi v REXYGENu.

Základními typy uzlů jsou klasický blok (BlockType), subsystém (SubsystemType) a úkol (TaskType). Ve stromové struktuře je úkol vždy kořenový blok a je umístěn ve složce 'Exec'.

2.2 Proměnné

Při vytváření stromové struktury jsou vytvořeny proměnné všech bloků spolu se svým datovým typem a povoleným rozmezím hodnot, které je uloženo v uzlech Min a Max. Jejich hodnoty jsou jediná data, která se synchronizují s exekutivou REXYGENu. Při zápisu a čtení se hodnoty synchronizují ihned. Je-li však hodnota čteného parametru v serveru dostatečně nová, server ji vrátí bez synchronizace. Pokud je parametr monitorován, synchronizuje se hodnota opakovaně, a to s nastaveným intervalem SYNC_INTERVAL (viz kapitola 3.1). Proces synchronizace je zobrazen na obrázku 2.2.



Obrázek 2.2: Při zápisu do serveru [1] se hodnota uloží a propíše do exekutivy [2]. Při čtení [3] server zkontroluje stáří hodnoty. Pokud klient požaduje novější hodnotu, pak si ji server vyžádá od exekutivy [4] a uloží [5]. Na závěr je hodnota poslána klientu [6].

Proměnné bloků (RexDataItemType) rozšiřují klasické OPC UA proměnné pro reprezentaci hodnot zařízení (DataItemType), jejichž součástí jsou vlastnosti Definition a ValuePrecision. Datový typ hodnoty proměnné odpovídá datovému typu proměnné

v REXYGENu. BrowseName a DisplayName proměnné odpovídají názvu proměnné v REXYGENu, u BrowseName je jako předpona uveden typ uzlu.

Proměnné se dělí na skalární vstupy, výstupy, parametry a stavy a na pole stavů a parametrů. Jednotlivé typy proměnných jsou rozeznatelné pouze pomocí atributu Description a vlastnosti Definition. Pole a skaláry lze také rozlišit pomocí atributu ValueRank. Pole mohou být reprezentovány jako vektor nebo matice. Pro interní použití se pole dělí na krátká, která jsou synchronizována jednorázově, a dlouhá, která musí být synchronizována po částech.

2.3 Události a verzování

Aby mohlo probíhat načítání bloků exekutivy, musí být složka 'Exec' verzovaná. Při každé změně struktury se nastaví její verze na aktuální a vyvolá se událost 'GeneralModelChangeEvent' ve které jsou uvedeny všechny odebrané a přidané uzly.

Kapitola 3

Konfigurace

Konfigurační INI soubor může obsahovat pouze ASCII znaky, doporučuje se používat kódování UTF-8 a záleží na velikosti písmen. Na konci a začátku řádků a kolem znamínka "=" nesmí být žádné přidané mezery. Komentáře začínají středníkem. Sekce jsou označeny názvem v hranatých závorkách a podsekce se tvoří dvojtečkou v názvu *[SEKCE: PODSEKCE]*. Parametry bez nastavené hodnoty nejsou brány v potaz.

V konfiguraci mohou být pouze ty sekce, které mají stejný název (velkými písmeny) jako některé z následujících podkapitol. Pokud je sekce v INI souboru dvakrát, data se jednoduše doplní. Sekce `User Token Policy (UTP)`, `Endpoint` a `Target` můžou mít podsekce, pro každou podsekci bude vytvořen jeden Endpoint či připojení k exekutivě.

V následujících kapitolách jsou popsány jednotlivé nastavitelné parametry. Parametry s přednastavenou původní hodnotou jsou vždy volitelné. Většina hodnot parametrů je ve formě textu. Číslo značí, že hodnota parametru musí být přirozené číslo. Y/N znamená, že parametr je přepínač, kde hodnota Y, YES, ON znamená povolení a N, NO, OFF vypnutí. Pole je značeno hranatými závorkami a jednotlivé hodnoty jsou odděleny čárkou [1,2,text]. Prázdné pole se chová, jako by hodnota nebyla vyplněna. Soubor značí systémovou cestu k souboru a složka cestu ke složce. Cesta je buď zadána absolutně, nebo relativně k umístění konfiguračního souboru.

3.1 Target

Tato sekce obsahuje možnosti, které ovlivňují komunikační spojení vytvořené s cílovým zařízením. Podrobnosti jsou vypsány v tabulce 3.1. Pro každou sekci `TARGET` bude vytvořena jedna složka `Exec` s názvem, který je určen názvem podsekce `TARGET:Exec1`.

Tabulka 3.1: Nastavení spojení s cílovým zařízením

Parametr	Typ	Výchozí hodnota	Popis
CONNECTION	Target URL	–	URL cílového zařízení. URL by mělo mít tvar `rex://[username[:password]@]host[:port]` nebo `rexs://[username[:password]@]host[:port]`.
SYNC_INTERVAL	Číslo	500	Doba synchronizace dat v milisekundách.
TCP_IDLE_INTERVAL	Číslo	30000	Interval oznámení o nečinnosti pro cílové zařízení v milisekundách. Tato hodnota by měla být menší než 60 000 (60 sekund).
USERNAME	Text	–	(Volitelné) Uživatelské jméno pro připojení k cílovému zařízení. Tento parametr má prioritu před uživatelským jménem v URL.
PASSWORD	Text	–	(Volitelné) Heslo pro připojení k REXYGENu. Tento parametr má prioritu před heslem v URL.
CERTIFICATE_PATH	Text	–	(Volitelné) Cesta ke klientskému certifikátu cílového zařízení. Používá se pro zabezpečený protokol `rexs`.

Tabulka 3.2: Možnosti cílového zařízení

Parametr	Typ	Výchozí hodnota	Popis
IGNORE_INPUTS	Y/N	N	Ignorovat vstupy bloků.
IGNORE_OUTPUTS	Y/N	N	Ignorovat výstupy bloků.
IGNORE_PARAMETERS	Y/N	N	Ignorovat parametry bloků.
IGNORE_STATES	Y/N	N	Ignorovat stavy bloků.
IGNORE_PARAMETER_ARRAYS	Y/N	N	Ignorovat pole parametrů bloků.
IGNORE_STATE_ARRAYS	Y/N	N	Ignorovat pole stavů bloků.
IGNORE_LARGE_ARRAYS	Y/N	N	Ignorovat velká pole bloků.
MAX_ARRAY_SIZE	Číslo	65536	Maximální povolená velikost polí [B].
SMALL_ARRAY_SIZE	Číslo	1024	Maximální velikost malých polí [B].
LARGE_ARRAY_READ_BLOCK_SIZE	Číslo	1024	Velikost bloku pro postupné čtení [B].
LARGE_ARRAY_WRITE_BLOCK_SIZE	Číslo	1024	Velikost bloku pro postupný zápis [B].
COMMUNICATION_DIAGNOSTICS	Y/N	N	Spustit diagnostiku komunikace s REXYGENem. Objekt CommunicationDiagnostics bude vytvořen ve složce `Exec`.
COMMUNICATION_DIAGNOSTICS_WINDOW_WIDTH	Číslo	10	Délka intervalu [s] pro výpočet plovoucího průměru využitý v diagnostice.
ADD_WHITE_LIST	Text/Pole	–	Přidání položky do seznamu částí exekutivy, které mají být zrcadleny. Seznam je reprezentován stromem. Každý záznam odpovídá textovému identifikátoru `NodeID`, který má být zobrazen. Může být zadán jednotlivě nebo jako seznam pravidel.
ADD_BLACK_LIST	Text/Pole	–	Přidání položky do seznamu částí exekutivy, které mají být ignorovány. Seznam je reprezentován stromem. Každý záznam odpovídá textovému identifikátoru `NodeID`, který má být ignorován. Může být zadán jednotlivě nebo jako seznam pravidel.

Použití parametrů ADD_WHITE_LIST a ADD_BLACK_LIST

Při použití `ADD_WHITE_LIST` a `ADD_BLACK_LIST` platí pravidla i pro dceřiné uzly (bloky, proměnné) cílového uzlu, pokud není specifikováno jinak. Pravidla lze zadávat ve formě pole nebo jednotlivých textových řetězců. Dříve přidaná pravidla se nezahazují, ale rozšiřují.

Příklad zadání pravidel:

```
ADD_WHITE_LIST=[$.task1,$.task2]
ADD_WHITE_LIST=[$.task3]
ADD_WHITE_LIST=$.task4
```

Tento zápis zavádí čtyři pravidla, která se kontrolují současně.

Pravidla s kvantifikátory

- Pokud název pravidla neobsahuje dvojtečku, platí pro uzel i všechny jeho dceřiné uzly (vstupy, výstupy, subsystémy, bloky atd.).
- Pravidla s dvojtečkou za názvem uzlu platí pouze pro vstupy, výstupy, parametry, stavy a pole tohoto uzlu (ne pro jeho dceřiné uzly).
- Za dvojtečku lze přidat další kvantifikátory pro filtrování:
  - `I` – Vstupy uzlu
  - `O` – Výstupy uzlu
  - `P` – Parametry uzlu
  - `S` – Stavy uzlu
  - `A` – Pole uzlu

Příklad rovnocenných zápisů: Následující dva zápisy mají stejný význam:

```
ADD_WHITE_LIST=$.task4:
ADD_WHITE_LIST=$.task4:IOPSA
```

Chování black listu

Pokud je uzel zařazen do black listu, nebude v OPC UA zobrazen ani on, ani jeho dceřiné uzly. Výjimku tvoří situace, kdy některý z jeho dceřiných uzlů (nebo jejich vstupy, výstupy, parametry, stavy, pole) je ve white listu. V takovém případě se rodičovské uzly zobrazí „holé“, aby struktura v OPC UA zůstala konzistentní.

Poznámka: Pokud uzel ve white listu na cílovém zařízení neexistuje, jeho rodičovské uzly se stejně zobrazí, protože pravidla vycházejí z konfigurace, ne z reálné struktury exekutivy.

Stromová struktura pravidel

Pravidla `ADD_WHITE_LIST` a `ADD_BLACK_LIST` tvoří stromovou strukturu. Název každého pravidla je rozdělen podle teček (kvantifikátory uvozené dvojtečkou se v tento moment neuvažují). Pokud se názvy na daných úrovních liší, vytvoří se nová větev. Také je možné použít místo jedné úrovně hvězdičku *, která funguje jako fallback na dané úrovni.
Vizualizace pravidel:
Pro názvy pravidel:

```
$.task3
$.task3.sub2
$.task4.sub2
$.task4.sub3:
$.task4.*
$.*.*.block1
```

Se vytváří následující strom pravidel, kde podtržené jsou uzly stromu, které mají nějaká pravidla:

```
$
+— task3
|    +— sub2
+— task4
|    +— sub2
|    +— sub3
|    +— *
+— *
     +— *
          +— block1
```

Vyhledávání v pravidlech

Při prohledávání pravidel stromu platí následující pravidla:

1. Na každé úrovni se hledají konkrétní pravidla.
2. Pokud nejsou nalezena, použijí se fallback pravidla označená hvězdičkou (*).
3. Pokud není nalezeno žádné pravidlo, algoritmus se vrátí o úroveň výše (doleva).

Příklad:

- Pro $.task3 se použijí pravidla $.task3 - existují a jsou nejvíce vpravo.
- Pro $.task3.sub1 se použijí pravidla $.task3 - existují a jsou nejvíce vpravo.
- Pro $.task3.sub2 se použijí pravidla $.task3.sub2 - existují a jsou nejvíce vpravo.

- Pro $.task4 neexistují žádná pravidla - $.task4 pravidla nemá, $.* pravidla nemá a $ pravidla také nemá.
- Pro $.task4.sub1 se použijí pravidla $.task4.* - $.task4.sub1 pravidla nemá, ale pro $.task4.* existují a jsou nejvíce vpravo.
- Pro $.task4.sub2 se použijí pravidla $.task4.sub2 - existují a jsou nejvíce vpravo.
- Pro $.task4.sub2.block1 se použijí pravidla $.task4.sub2 - existují a jsou nejvíce vpravo.
- Pro $.task4.sub3 se použijí pravidla $.task4.sub3 - existují a jsou nejvíce vpravo.
- Pro $.task5 neexistují žádná pravidla - $.task5 ani $.* pravidla nemají a $ pravidla také nemá.
- Pro $.task5.sub1 neexistují žádná pravidla - $.task5.sub1, $.task5, $.*.sub1 ani $.*.* pravidla nemají a $ pravidla také nemá.
- Pro $.task5.sub1.block1 se použijí pravidla $.*.*.block1 - $.task5, $.*.sub1 ani $.*.* pravidla nemají, ale pro $.*.*.block1 existují a jsou nejvíce vpravo.

Poznámka: Pro $.task3.sub1.block1 neplatí pravidla pro $.*.*.block1, protože se pro něj jako první najdou pravidla $.task3.
Pozor: Výjimky z pravidel vyžadují explicitní definici. Například z pravidla $.*.sub1.* lze udělat výjimku pomocí $.*.sub1.*.variable1. Z pravidla $.task4.sub1.* lze udělat výjimku $.task4.sub1.*.variable1.

3.2 Aplikace

Tato sekce obsahuje hlavní údaje o serveru, viz tabulka 3.3. Zde se nastavuje i Namespace serveru a Namespace exekutivy. Namespace serveru je určeno parametrem APPLICATION_URI. Namespace exekutivy odpovídá následujícímu tvaru:

`urn:Rex:Exec:<COMPANY_URI_NAME>:<PROJECT_URI_NAME>:<INSTANCE_URI_NAME>:<TARGET_NAME>`

Parametry COMPANY_URI_NAME, PROJECT_URI_NAME a INSTANCE_URI_NAME volte tak, aby jejich kombinace byla unikátní pro každou běžící instanci REXYGENu, aby tak nedocházelo k chybám při použití více OPC UA serverů. Podle specifikace OPC UA by více serverů připojených k jedné instanci REXYGENu mělo mít stejné názvy, servery připojené k různým instancím REXYGENu musí mít různé názvy. Parametr TARGET_NAME odpovídá názvu podsekce TARGET (viz 3.1).

Tabulka 3.3: Nastavení aplikace

Parametr	Typ	Výchozí hodnota	Popis
APPLICATION_ CERTIFICATE_ PATH	Soubor	–	Certifikát serveru ve formě DER.
APPLICATION_ PRIVATE_KEY_ PATH	Soubor	–	Soukromý klíč certifikátu serveru ve formě PEM.
APPLICATION_ PRIVATE_KEY_ PASSWORD	Text	–	(Volitelné) Heslo ke klíči certifikátu serveru.
APPLICATION_ URI	URI serveru	–	Tato položka by měla být shodná s URI v certifikátu serveru a zároveň bude použita jako Namespace serveru.
COMPANY_URI_ NAME	Text	–	Tento text bude částí Namespace exekutivy.
PROJECT_URI_ NAME	Text	–	Tento text bude částí Namespace exekutivy.
INSTANCE_URI_ NAME	Text	–	Tento text bude částí Namespace exekutivy.

3.3 Security

Sekce security obsahuje nastavení validace a umístění klientských certifikátů. Pokud všechny Endpointy mají nastavené zabezpečení komunikace pouze na None a není použito přihlášení pomocí certifikátu, je celá tato sekce volitelná. Server využívá OpenSSL, proxy certifikáty jsou zakázány.

Pro vytvoření certifikátů a adresářů pro klientské certifikáty lze využít aplikaci RexOpcUaConfig, která je popsána v kapitole 5.1.

Tabulka 3.4: Zabezpečení

Parametr	Typ	Výchozí hodnota	Popis
CERTIFICATE_ TRUST_LIST_ PATH	Složka	–	Důvěryhodné certifikáty - certifikáty, které jsou zde uložené, a certifikáty, které jsou jimi podepsané, jsou povoleny.
CERTIFICATE_ REJECTED_LIST_ PATH	Složka	–	(Volitelné) Odmítnuté certifikáty - zde se shromažďují všechny certifikáty, které byly serverem odmítnuty. Pokud není zadáno, odmítnuté certifikáty se nebudou ukládat.
CERTIFICATE_ REVOCATION_ LIST_PATH	Složka	–	(Volitelné) Odvolané (zneplatněné) certifikáty, které byly vyřazeny.
CERTIFICATE_ ISSUER_LIST_ PATH	Složka	–	(Volitelné) Certifikační autority - certifikáty potřebné k ověření certifikačního řetězce, které ale nejsou automaticky důvěryhodné.
CERTIFICATE_ REVOCATION_ CHECK_OPTION	N/L/S/A	N	Kontrola zneplatněních certifikátů. N: Žádná kontrola L: Pouze listy S: Bez sebou–podepsaných A: Všechny
CHECK_SELF_ SIGNATURE	Y/N	N	Kontrola podpisu sebou–podepsaných certifikátů.
CHECK_ CERTIFICATE_ URL	Y/N	N	Kontrola URL certifikátu vůči URI aplikace.

3.4 User Token Policy

Sekce User Token Policy (UTP) definuje možnostmi autorizace a autentizace klientů při připojení k Endpointu, parametry jsou popsány v tabulce 3.5. Úprava přihlašovacích údajů a rolí jsou popsány v kapitole 4 a 5.2.

Klient při vytváření připojení zadává způsob ověření identity (UserTokenPolicy). Při anonymním přihlášení nejsou po klientovi požadovány žádné další informace. Při použití přihlašovacích údajů (credentials) musí klient poskytnou uživatelské jméno a heslo, které se poté validují na serveru. Při použití certifikátu musí klient poskytnout certifikát, u kterého server ověří důvěryhodnost. Validaci certifikátů pro účely autentizace lze nastavit stejným způsobem jako v kapitole 3.3.

V této sekci jsou definovány jednotlivé přihlašovací politiky. Název sekce je zároveň

názvem politiky. Pokud má Endpoint podporovat některou z politik, musí její název zmínit ve svém seznamu USER_TOKEN_POLICY.

Při použití politiky s přihlašovacími údaji musí být definován INI soubor, který obsahuje jméno, zakódované heslo a roli uživatele. Server poté zjistí, zda klient poskytl správnou kombinaci jména a hesla a pokud ano, povolí mu připojení a přiřadí mu korespondující roli. Heslo je kódováno pomocí různých mechanismů. Aby mohl uživatel mechanismus ovlivnit a jeho heslo nemohl rozluštit někdo jiný, lze nastavit parametr OPTIONAL_ENCODING_SALT. Server přečte pouze ta hesla, která byla zakódována s tímto parametrem, při výměně je tedy nutné přegenerovat soubor s přihlašovacími údaji.

Tabulka 3.5: User Token Policy (UTP)

Parametr	Typ	Výchozí hodnota	Popis
USER_TOKEN_POLICY_TYPE	Anonymous, Certificate, Username	–	Typ přihlašovací politiky.
AUTH_ROLE	Supervisor, Operator, Observer, AuthorizedUser, Anonymous	–	(Anonymous, Certificate) Přiřazené uživatelské oprávnění.
CREDENTIALS_INI_PATH	Soubor	–	(Username) Soubor, kde jsou zapsáni uživatelé s heslem a rolí.
OPTIONAL_ENCODING_SALT	Text	q1we58	(Username) Hodnota, pomocí které bude zakódováno heslo v souboru s uživateli. Tato hodnota nemusí být příliš velká, 5 - 20 ASCI znaků stačí.
CERTIFICATE_TRUST_LIST_PATH	Složka	–	(Certificate) Složka s důvěryhodnými certifikáty.
...	...	–	(Certificate) Lze nastavit i další parametry kontroly certifikátů. Možnosti jsou stejné jako v tabulce 3.4

3.5 Endpoint

Sekce Endpoint obsahuje nastavení OPC UA Endpointů, ke kterým bude možné se připojit. V této sekci lze vytvářet podsekce, kde každá podsekce vytvoří nový Endpoint a musí tedy obsahovat všechny povinné parametry. Všechny parametry jsou popsány v

tabulce 3.6.

Pokud je potřeba využít Endpoint pro služby Discovery, doporučujeme v URL Endpointu nepoužívat localhost, ale veřejnou IP adresu. V opačném případě bude možná (v závislosti na implementaci klienta a discovery serveru) klient volat adresu na svém vlastním localhostu, nikoliv na serveru. URL adresa by měla mít následující tvar:

opc.tcp://<IP adresa | DNS>:<port>[/<konečná část URL>]

Tabulka 3.6: Nastavení Endpointu

Parametr	Typ	Výchozí hodnota	Popis
URL	URL Endpointu	–	URL Endpointu pro připojení pomocí protokolu opc.tcp.
SECURITY_ POLICY	[Zabezpečení] (pole)	–	Povolené zabezpečení komunikace - detaily v tabulce 3.7.
USER_TOKEN_ POLICY (UTP)	[Přihlašovací politiky] (pole)	–	Povolené metody autentizace (UTP). UTP se definují pomocí tabulky 3.5.
OPEN_WHEN_ ALL_BROWSED	Y/N	N	Otevře Endpoint až po načtení všech targetů a zavře ho při každé změně. Mód pro špatně implementované OPC UA klienty ignorující zprávy o změně adresního prostoru.

Tabulka 3.7: Zabezpečení komunikace a míra bezpečnosti (červená nejnižší (zastaralá), oranžová střední, zelená vysoká, modrá nejvyšší

Zabezpečení	Podpis	Šifrování	Algoritmus
None	Ne	Ne	–
Sign_Basic128Rsa15	Ano	Ne	Basic128Rsa15
SignEncrypt_Basic128Rsa15	Ano	Ano	Basic128Rsa15
Sign_Basic256	Ano	Ne	Basic256
SignEncrypt_Basic256	Ano	Ano	Basic256
Sign_Basic256Sha256	Ano	Ne	Basic256Sha256
SignEncrypt_Basic256Sha256	Ano	Ano	Basic256Sha256
Sign_Aes128Sha256RsaOaep	Ano	Ne	Aes128Sha256RsaOaep
SignEncrypt_ Aes128Sha256RsaOaep	Ano	Ano	Aes128Sha256RsaOaep
Sign_Aes256Sha256RsaPss	Ano	Ne	Aes256Sha256RsaPss
SignEncrypt_ Aes256Sha256RsaPss	Ano	Ano	Aes256Sha256RsaPss

3.6 Discovery

Tato sekce se zabývá registrací k průzkumnému (Discovery) serveru. Celá tato sekce je nepovinná. Parametr ENDPOINT_URL může obsahovat více průzkumných bodů, které budou zaregistrovány, není to však doporučováno, jedna adresa by měla stačit. Klient při dotazu na tento Endpoint zjistí adresy všech Endpointů na serveru. Parametr ENDPOINT_URL by měl být shodný s parametrem URL některého z Endpointů, nicméně tato shoda není kontrolována.

Aby byla registrace úspěšná, je nutné použít správnou URL průzkumného serveru, správné zabezpečení a cestu k jeho certifikátu v parametru SERVER_CERTIFICATE_PATH. Průzkumný server naopak musí důvěřovat aplikačnímu certifikátu serveru. Nastavení registrace je popsáno v tabulce 3.8.

Tabulka 3.8: Nastavení registrace k průzkumnému serveru

Parametr	Typ	Výchozí hodnota	Popis
ENDPOINT_URL	[URL Endpointu] (pole)	–	(Volitelné) URL registrovaného Endpointu. Měla by být shodná s URL některého z Endpointů.
SERVER_ CERTIFICATE_ PATH	Soubor	–	Cesta k certifikátu průzkumného serveru.
SERVER_URL	URL	–	URL průzkumného serveru, u nějž se bude server registrovat. URL musí začínat *opc.tcp://*.
SECURITY_ POLICY	Zabezpečení	–	Použité zabezpečení při komunikaci s průzkumným serverem - detaily v tabulce 3.7. Lze použít právě jedno zabezpečení, které průzkumný server podporuje.
REFRESH_TIME	Číslo	30000	Interval obnovy registrace v ms.

3.7 Options

V kategorii Options se nacházejí zbylé parametry, kterými lze ovlivňovat běh a bezpečnost serveru. Tyto parametry nastavujte pouze se znalostí specifikace OPC UA. Všechny parametry v této sekci jsou nepovinné, viz tabulka 3.9 a 3.10.

Tabulka 3.9: Obecné nastavení

Parametr	Typ	Výchozí hodnota	Popis
MIN_SAMPLING_INTERVAL	Číslo	600	Minimální interval pro vzorkování uzlů.
MAX_SAMPLING_INTERVAL	Číslo	10000	Maximální interval pro vzorkování uzlů.
MIN_PUBLISHING_INTERVAL	Číslo	500	Minimální interval pro publikování.
MAX_PUBLISHING_INTERVAL	Číslo	600000	Maximální interval pro publikování.
MIN_SESSION_TIMEOUT	Číslo	1000	Minimální životnost spojení v ms.
MAX_SESSION_TIMEOUT	Číslo	600000	Maximální životnost spojení v ms.
MAX_PIPED_PUBLISH_REQUEST	Číslo	5	Maximální počet uskladněných požadavků k publikování. Server na další požadavky vrací chybový kód *TooManyPublishRequests*.
MAX_NODES_TO_ANALYZE_PER_QUERY_REQUEST	Číslo	100	Maximální počet analyzovaných uzlů dotazovacími službami.
MAX_DATA_CHANGE_MONITORING_QUEUE_SIZE	Číslo	1000	Maximální velikost fronty pro položky monitorované na změnu dat.
MAX_EVENT_MONITORING_QUEUE_SIZE	Číslo	1000	Maximální velikost fronty pro položky monitorované na události.
MAX_DATA_SETS_TO_RETURN	Číslo	0	Maximální počet datových kolekcí v odpovědi na dotazovací služby.
ENABLE_AUDIT_EVENTS	Y/N	N	Server vytváří události při vytvoření relace, aktivování relace, volání služby pro zrušení a pokud je vytvořena relace, ale nesouhlasí URL v certifikátu.

Tabulka 3.10: Obecné nastavení

Parametr	Typ	Výchozí hodnota	Popis
ENABLE_ DIAGNOSTICS	Y/N	N	Server vytváří standarní diagnostická data.
ALLOW_SWITCH_ DIAGNOSTICS	Y/N	N	Povolit zapnutí/vypnutí standardní diagnostiky serveru.
MIN_ DIAGNOSTICS_ UPDATE_ INTERVAL	Číslo	100	Minimální interval pro upravení diagnostiky.
MAX_ DIAGNOSTICS_ UPDATE_ INTERVAL	Číslo	86400000	Maximální interval pro upravení diagnostiky.
MAX_SESSIONS	Číslo	0	Maximální počet relací na server, 0 pro neomezeně mnoho.
MAX_SESSIONS_ PER_ENDPOINT	Číslo	0	Maximální počet relací na Endpoint serveru, 0 pro neomezeně mnoho.
MAX_ SUBSCRIPTIONS	Číslo	0	Maximální počet odběrů na server, 0 pro neomezeně mnoho.
MAX_ SUBSCRIPTIONS_ PER_SESSION	Číslo	0	Maximální počet odběrů na jednu relaci, 0 pro neomezeně mnoho.
MAX_ SUBSCRIPTION_ LIFETIME	Číslo	120000	Maximální životnost odběru v ms.
MAX_ MONITORED_ ITEMS	Číslo	0	Maximální počet monitorovaných položek na server, 0 pro neomezeně mnoho.
MAX_ MONITORED_ ITEMS_PER_ SUBSCRIPTION	Číslo	0	Maximální počet monitorovaných položek na odběr, 0 pro neomezeně mnoho.

Kapitola 4

Autentifikace a autorizace

V OPC UA serveru pro REXYGEN se využívá pět rolí s následujícími oprávněními: AuthorizedUser může procházet Adresní prostor. Observer může zároveň číst data z proměnných bloků REXYGENu. Operátor má stejná práva jako Observer, má ale navíc povoleno zapisovat do proměnných bloků REXYGENu a tím ovlivňovat běžící exekutivu. Supervisor má práva operátora, a navíc může pracovat s diagnostikou připojení a spouštět metody. V tabulce 4.1 jsou uživatelská práva zobrazena graficky.

Tabulka 4.1: Uživatelská práva

Práva	Supervisor	Operator	Observer	AuthorizedUser	Anonymous
Procházení	X	X	X	X	
Čtení hodnot	X	X	X		
Zápis hodnot	X	X			
Čtení práv	X				
Diagnostika komunikace	X				
Volání metod	X				

Server určuje roli uživatele na základě přihlašovací politiky, kterou klient použije při navázání spojení. Pokud klient použije některou z anonymních politik nebo politik s certifikátem, server mu automaticky přiřadí roli, která je na politiku navázaná. Druhou možností je přihlášení pomocí jména a hesla, kde server nastaví uživateli roli, kterou má přiřazenou. Klient může využít pouze ty přihlašovací politiky, které podporuje daný Endpoint, k němuž se snaží připojit. Zabezpečení serveru lze tedy zajistit správným nastavením přihlašovacích politik jednotlivých Endpointů, viz kapitola 3.5.

Pokud server využívá přihlášení pomocí přihlašovacích údajů, musí mu být nastavena cesta k INI souboru, který tyto přihlašovací údaje obsahuje. Údaje se načtou ze souboru při startu serveru. Pro manipulaci se souborem lze využít grafické rozhraní RexOpcUaConfig.

Oba programy využívají konfigurační soubor serveru, z něhož získají údaje o cestě k souboru s přihlašovacími údaji a přídavném kódování OPTIONAL_ENCODING_

SALT. Pokud je hodnota přídavného kódování změněna, je nutné všem uživatelům znovu nastavit hesla nebo celý soubor vygenerovat znovu.

4.1 INI soubor s přihlašovacími údaji

INI soubor s přihlašovacími údaji obsahuje informace o uživatelích, jejich heslech a rolích. Tento soubor obsahuje pět sekcí odpovídajících jednotlivým rolím OPC UA serveru: SUPERVISOR, OPERATOR, OBSERVER, AUTHORIZED_USER, ANONYMOUS. Tyto sekce obsahují páry uživatelů se zakódovanými hesly. Hesla jsou zakódována pomocí SHA1 řetězce: `<password><username><OPTIONAL_ENCODING_SALT>`. Příklad INI souboru s přihlašovacími údaji je zobrazen níže.

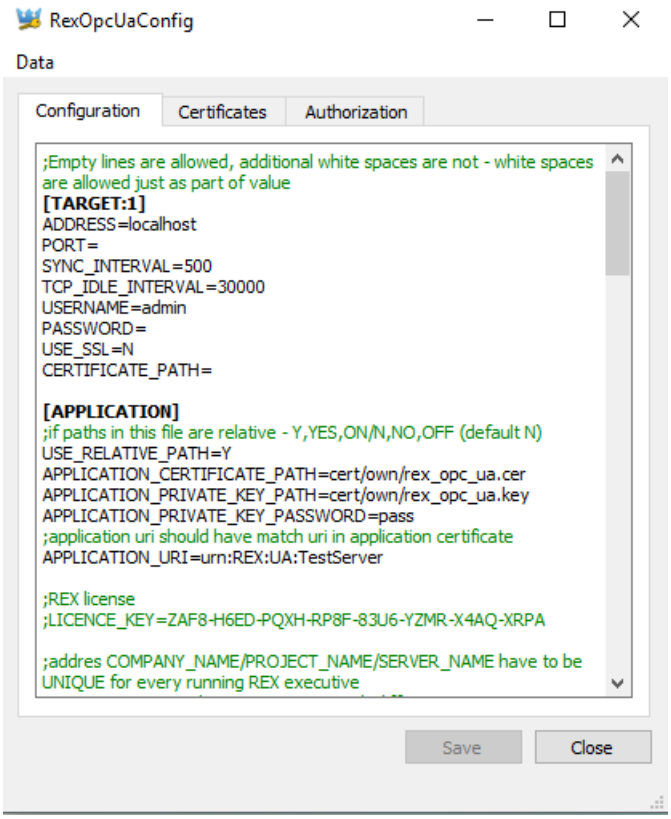
```
[SUPERVISOR]
supervisor=718DA2408623AD7786E2E79AA700E8A8FBC49221

[OPERATOR]
operator=844BD4CBFF1FEF80251306E0E359243CC267DB2B
```

Kapitola 5

RexOpcUaConfig

RexOpcUaConfig je grafické rozhraní určené pro nastavení OPC UA serveru pro REXYGEN. Umožňuje úpravu INI souboru, vytvoření certifikátů, správu uživatelů a použití některého z příkladů.

Na záložce 'Configuration' (obrázek 5.1) je zobrazen obsah přednastaveného konfiguračního INI souboru. Tento soubor lze pomocí vestavěného editoru upravovat, ukládat a znovu načítat. Při každém uložení nebo načtení souboru zpracuje program aktuální konfiguraci. Pokud narazí na závažný problém, zapíše nalezenou chybu do záložky 'Errors'. Tento dialog nekontroluje všechny nastavitelné parametry. Kontroluje pouze ty, které potřebuje ke své funkci.

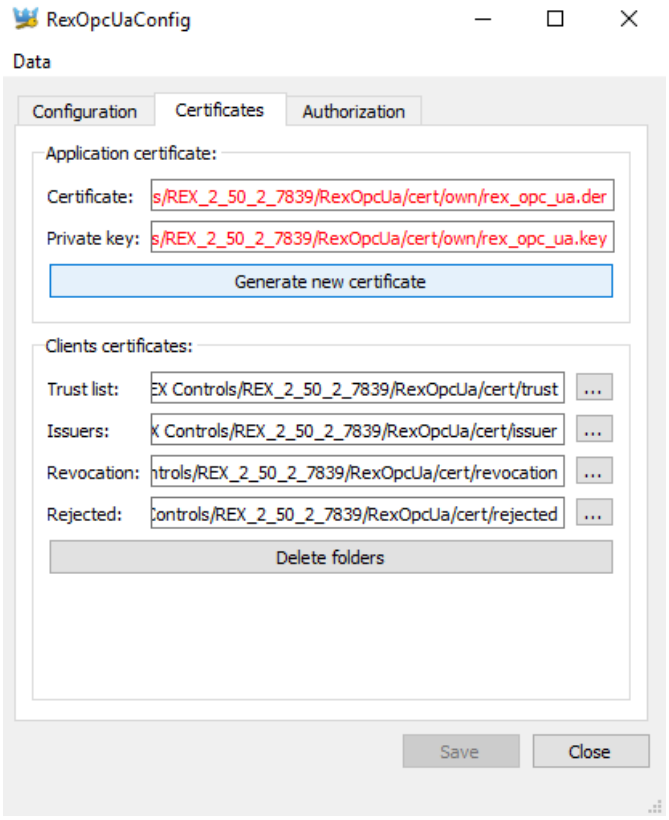


Obrázek 5.1: RexOpcUaConfig s vestavěným INI editorem

5.1 Certifikáty

Na záložce ‘Certificates’ (obrázek 5.2) je možné spravovat certifikát aplikace a klientské certifikáty. Všechny cesty se čtou z konfiguračního INI souboru. Pokud soubor neexistuje, obarví se text na červeno.

Klientské certifikáty jsou uloženy v různých složkách. RexOpcUaConfig umožňuje vytvořit, otevřít a smazat tyto složky. Pro povolení klientského certifikátu zkopírujte certifikát do složky ‘Trust list’. Certifikáty klientů, kteří se pokusili připojit a byli odmítnuti kvůli certifikátu, jsou uloženy ve složce ‘Rejected’.

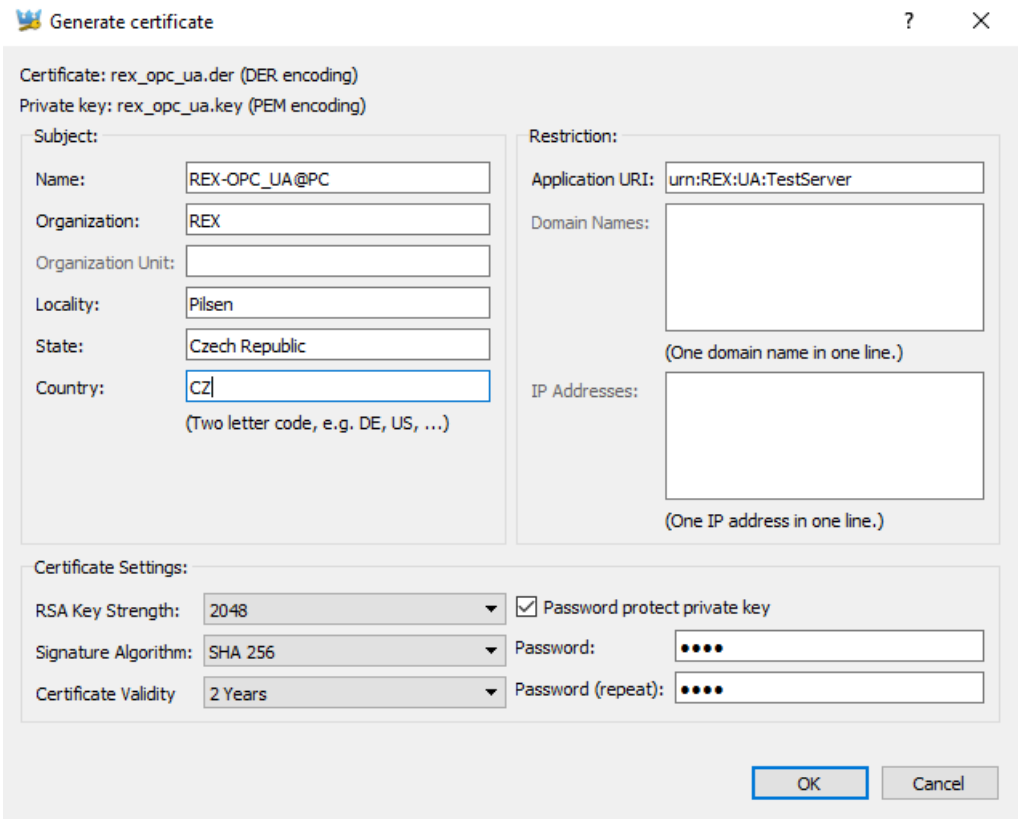


Obrázek 5.2: Správa certifikátů

Při vytváření aplikačního certifikátu se otevře samostatný dialog ‘Generate certificate’. Pole ‘Password’ a ‘Application URI’ jsou předvyplněna hodnotami parametrů APPLICATION_PRIVATE_KEY_PASSWORD a APPLICATION_URI z konfiguračního INI souboru.

Pole ve skupině ‘Subject’ je možné zvolit podle vlastního uvážení. Pole ve skupině ‘Restriction’ umožňují omezení použití certifikátu na určitou IP adresu nebo doménu. Hodnota pole ‘Application URI’ musí souhlasit s hodnotou parametru APPLICATION_URI v konfiguračním INI souboru. Parametry ve skupině ‘Certificate Settings’ ovlivňují dobu trvanlivosti certifikátu a sílu soukromého klíče.

Umístění generovaného certifikátu a klíče je určeno v konfiguračním INI souboru a je vidět na záložce ‘Certificates’ (obrázek 5.2). Soubor klíče (.key, .pem) je uložen ve formátu PEM. Formát certifikátu je odvozen z koncovky jeho souboru. Pro soubor s koncovkou .pem je certifikát uložen ve formátu PEM pro jiné koncovky souboru (.der, .cer, .crt, .cert) je vygenerován certifikát ve formátu DER.



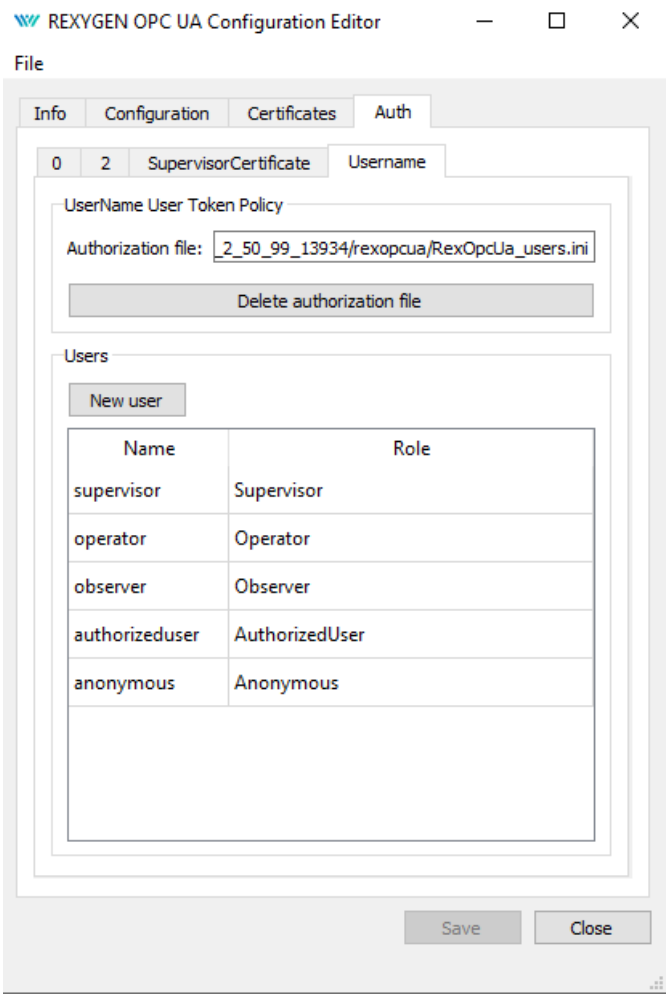
Obrázek 5.3: Dialog pro vytváření aplikačního certifikátu

5.2 Autentizace

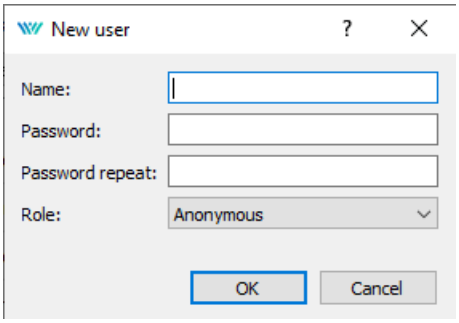
Na záložce 'Authorization' (obrázek 5.4) lze nastavit uživatelské přihlašovací údaje. Tato stránka se objeví pouze, pokud je v konfiguračním INI souboru nastaven parametr CREDENTIALS_INI_PATH. Veškeré údaje o uživatelích jsou poté ukládány do INI souboru na této cestě.

Pro bezpečné použití tohoto způsobu autentizace je doporučeno smazat soubor s přihlašovacími údaji, nastavit parametr OPTIONAL_ENCODING_SALT na vlastní hodnotu (viz tabulka 3.5) a poté celý soubor vytvořit znovu.

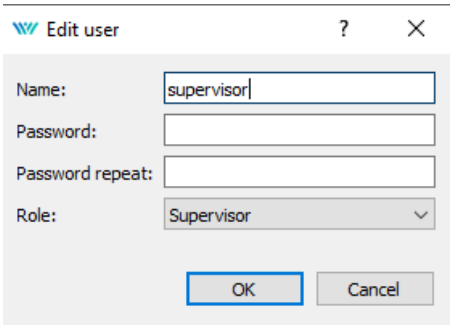
Uživatele lze spravovat pomocí jednoduchého grafického rozhraní. Uživatele lze přidávat (obrázek 5.5), upravovat (obrázek 5.6) a mazat. Klient se pak přihlásí se svým jménem a heslem (program UaExpert na obrázku 6.9).



Obrázek 5.4: Správa uživatelů



Obrázek 5.5: Dialog pro vytvoření nového uživatele

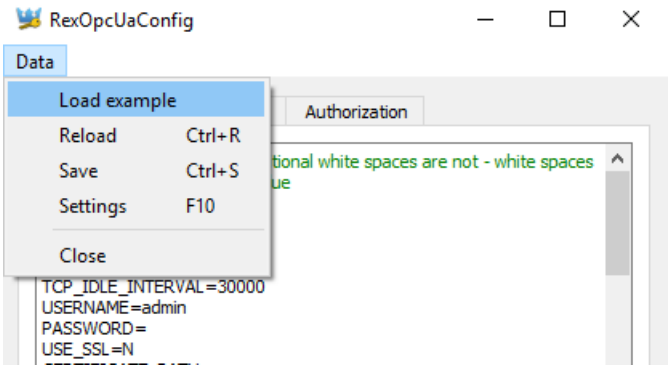


Obrázek 5.6: Dialog pro úpravu uživatele

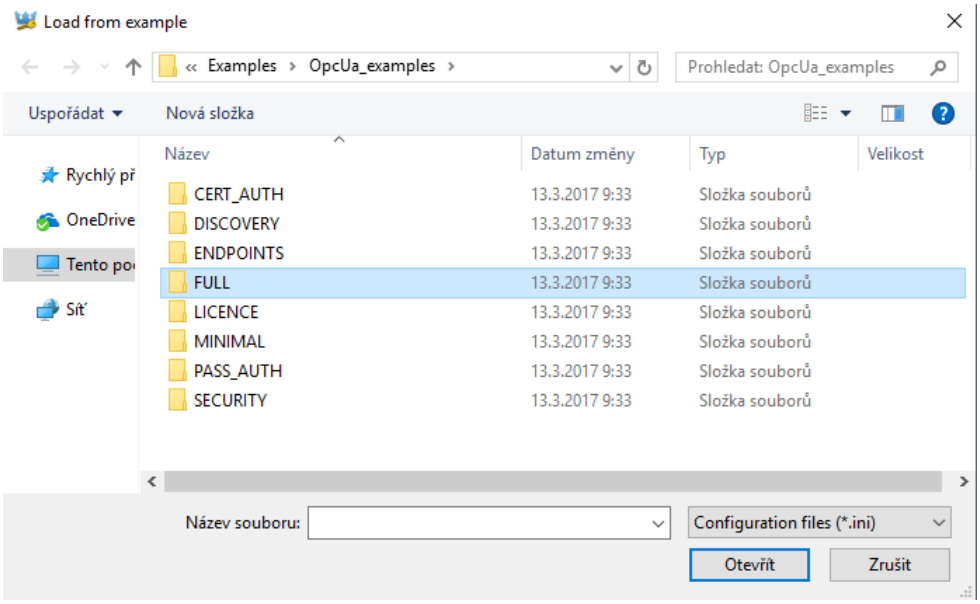
5.3 Využití příkladů

Program RexOpcUaConfig umožňuje pro snadnější použití využít některý příklad konfigurace (kapitola 6.1) a použít ho jako základ vlastní konfigurace (obrázky 5.7, 5.8 a 5.9). Obsah konfiguračního INI souboru se přepíše obsahem vybraného příkladu.

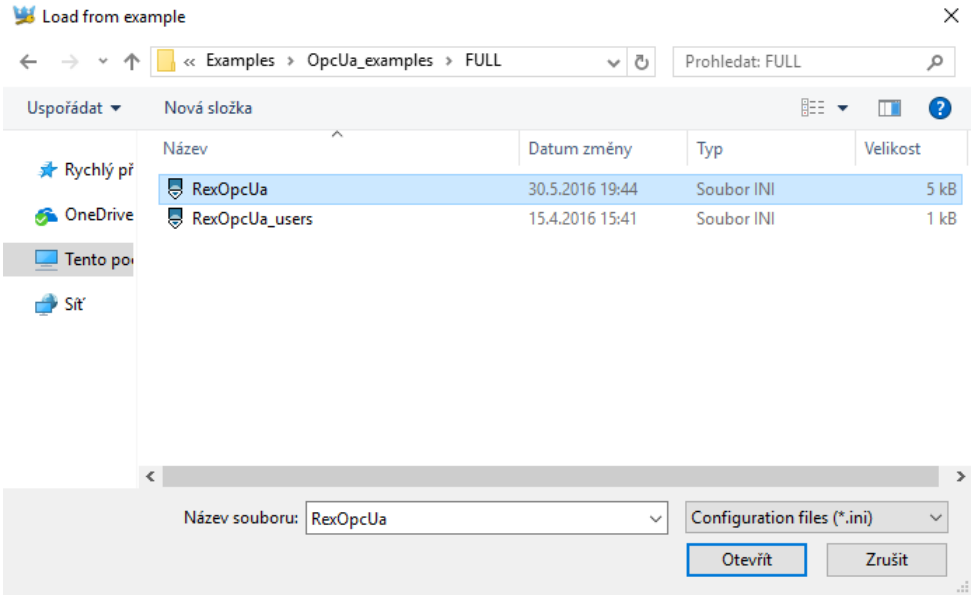
Soubory s přihlašovacími údaji (RexOpcUa_users.ini) se nekopírují. Pro správnou funkci serveru je nutné nastavit správně parametr OPTIONAL_ENCODING_SALT a vygenerovat nový soubor s přihlašovacími údaji. Pokud byl starý soubor vygenerován už s novým parametrem OPTIONAL_ENCODING_SALT, lze použít ten.



Obrázek 5.7: Příkaz na načtení příkladu



Obrázek 5.8: Seznam příkladů



Obrázek 5.9: Vybraná konfigurace

Kapitola 6

Návod ke spuštění

Základem pro spuštění serveru je správně nastavený konfigurační INI soubor a aplikační certifikát a soukromý klíč. Veškeré další důležité činnosti jsou navázány na nastavení konfigurace, správu klientských certifikátů a zajištění správné činnosti REXYGENu, ke kterému je server připojen. Ostatní záležitosti už musí vyřídit klient.

Pro jednoduchý start serveru jsou zde uvedeny jednotlivé kroky potřebné pro připravení serveru ke spuštění, poté lze server jednoduše spustit a pokud je vše nastaveno v pořádku, server bude po spuštění fungovat.

1. **Instalace** REXYGENu spolu s OPC UA serverem (pokud není nainstalován)
2. **Změna konfiguračního souboru** - výběr z možností
   (a) Zkopírování připraveného souboru ze složky příkladů pro REXYGEN (viz kapitolu 6.1)
   (b) Změna současného konfiguračního souboru
3. **Tvorba certifikátu** (pokud neexistuje) - výběr z možností
   (a) Pomocí RexOpcUaConfig (viz kapitolu 5.1)
   (b) Pomocí OpenSSL
   (c) Pomocí skriptu */etc/rexcore/rexopcua.d/10-cert.sh*
   - `-i <IP_ADDRESS>` – Externí IP adresa OPC UA serveru
   - `-d <DNS>` – Externí DNS OPC UA serveru
   - `-f` – Explicitní přegenování certifikátu
   - `-k` – Explicitní přegenerování soukromého klíče
4. Změna přihlašovacích údajů uživatelů (pokud je vyžadováno)
5. **Nastavení klientských certifikátů** (pokud Endpointy podporují zabezpečenou komunikaci nebo přihlášení pomocí certifikátů)
   (a) Vytvoření složek pro klientské certifikáty (RexOpcUaConfig, viz 5.1)

(b) **Zkopírování** souborů **certifikátů klientů**, kteří se chtějí přihlašovat pomocí zabezpečeného kanálu, **do složky trust** (CERTIFICATE_TRUST_LIST_PATH)

6. Nastavit službu **Discovery** (pokud je to vyžadováno)
   (a) Nalézt informace o Discovery serveru
   (b) Zkopírovat certifikát serveru do trust složky Discovery serveru
   (c) Zkopírovat certifikát Discovery serveru do složky pro certifikáty serveru (doporučené)
   (d) Nastavit sekci DISCOVERY v konfiguračním souboru serveru pro REXYGEN
      i. SERVER_URL - URL Endpointu Discovery serveru
      ii. SECURITY_POLICY - Způsob zabezpečení připojení k Discovery serveru (ten ho musí podporovat)
      iii. SERVER_CERTIFICATE_PATH - Cesta k certifikátu Discovery serveru (doporučeně ve složce pro certifikáty serveru)
      iv. ENDPOINT_URL - Seznam Endpointů (stačí jeden), které bude mít Discovery server v databázi (jeden z Endpointů serveru)

6.1 Příklady

Pro jednodušší nastavení konfiguračních souborů byly vytvořeny předpřipravené příklady, které lze použít jako základ pro nové nastavení.

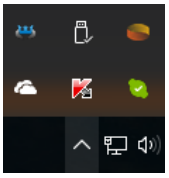
- `Minimal` - Minimální konfigurace pro nezabezpečený Endpoint a REXYGEN na localhostu
- `Secured_communication` - Konfigurace s Endpointem, který podporuje zabezpečenou komunikaci
- `Username_Authentication` - Konfigurace s Endpointem, který podporuje přihlášení pomocí přihlašovacích údajů
- `Certificate_Authentication` - Konfigurace s Endpointem, který podporuje přihlášení pomocí certifikátů
- `Multi_Authentication` - Konfigurace s Endpointem, který podporuje přihlášení pomocí více politik najednou
- `Endpoints` - Konfigurace se dvěma Endpointy
- `Discovery` - Konfigurace s připojením informací o serveru na Discovery server
- `Full` - Konfigurace se všemi možnými parametry

Pro první použití je doporučeno použít některý z těchto příkladů a nastavit parametry ADDRESS, COMPANY_URI_NAME, PROJECT_URI_NAME a INSTANCE_URI_NAME. Případně APPLICATION_PRIVATE_KEY_PASSWORD a OPTIONAL_ENCODING_SALT. Další parametry je možné měnit postupně a získávat tím informace o možnostech serveru.

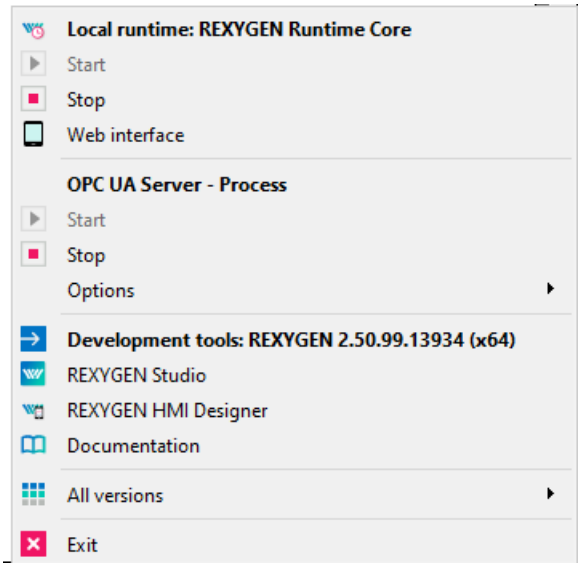
6.2 Služba OPC UA

OPC UA server pro systém REXYGEN je možno (a doporučeno) spouštět jako systémovou službu.

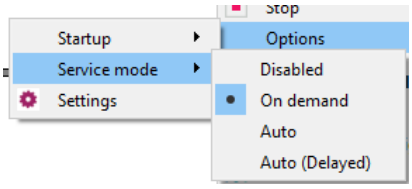
V OS Windows je možné tuto službu spravovat pomocí monitorovací aplikace (viz 6.1, 6.2 a 6.3). Pomocí této aplikace je možné spouštět, zastavovat službu, nastavit její automatické spouštění a otevřít dialog RexOpcUaConfig pro nastavení serveru (kapitola 5).



Obrázek 6.1: Monitorovací služba REXYGENu



Obrázek 6.2: Umístění OPC UA služby v nabídce monitorovací služby



Obrázek 6.3: Obsluha a nastavení OPC UA služby

V OS Linux je možné službu spustit pomocí system.d.

systemctl start rexopcua

Této službě lze nastavit cestu ke konfiguračnímu INI souboru parametrem CFGFILE v souboru:

/etc/rexcore/rexopcua.conf

6.3 OPC UA Klienti

Pro vyzkoušení OPC UA serveru je možné použít některého z veřejně dostupných klientů. I přes přesnou specifikaci se každý klient chová trochu jinak a ne vždy využívá všech možností, které mu server nabízí. V tomto návodu jsme použili klienta UaEpert od firmy Unified Automation GmbH a software myScada.

U obou klientů bude navíc vysvětleno, jak má vypadat nastavení pro anonymní připojení (obrázek 6.4) a pro připojení pomocí přihlašovacích údajů (obrázek 6.5 a 6.6).

```
[AUTH]
;file with usernames and passwords and user token id for username/password login (optional - binded to
CREDENTIALS_INI_PATH=RexOpcUa_users.ini
CREDENTIALS_USER_TOKEN_POLICY_ID=UsernamePassword
OPTIONAL_ENCODING_SALT=q1we58
;policies for anonymous access with default privileges
ADMIN_USER_TOKEN_POLICY_ID=0
OPERATOR_USER_TOKEN_POLICY_ID=1
GUEST_USER_TOKEN_POLICY_ID=2
;policies for access with certificate
CERT_ADMIN_USER_TOKEN_POLICY_ID=AdminCertificate
CERT_OPERATOR_USER_TOKEN_POLICY_ID=OperatorCertificate
CERT_GUEST_USER_TOKEN_POLICY_ID=GuestCertificate

[ENDPOINT:1]
SECURITY_POLICY=[None,SignEncrypt_Basic256]
;policy id has to be identical to id of predefined user token policies
USER_TOKEN_POLICY_ID=[AdminCertificate,UsernamePassword,2]
URL=opc.tcp://localhost:4885/REX

[ENDPOINT:2]
SECURITY_POLICY=[None,Sign_Basic128Rsa15,SignEncrypt_Basic128Rsa15,Sign_Basic256,SignEncrypt_Basic256]
USER_TOKEN_POLICY_ID=[0]
;additional endpoint url is optional
URL=opc.tcp://localhost:4888/None/None
```

Obrázek 6.4: Nastavení Endpointu bez zabezpečení

```
[AUTH]
;file with usernames and passwords and user token
CREDENTIALS_INI_PATH=RexOpcUa_users.ini
CREDENTIALS_USER_TOKEN_POLICY_ID=UsernamePassword
```

Obrázek 6.5: Nastavení přihlašování pomocí přihlašovacích údajů

```
[ENDPOINT:2]
SECURITY_POLICY=[None,Sign_Basic128Rsa15,SignEncrypt_Basic128Rsa15,Sign_Basic256,SignEncrypt_Basic256]
USER_TOKEN_POLICY_ID=[UsernamePassword]
;additional endpoint url is optional
URL=opc.tcp://localhost:4888/None/None
```

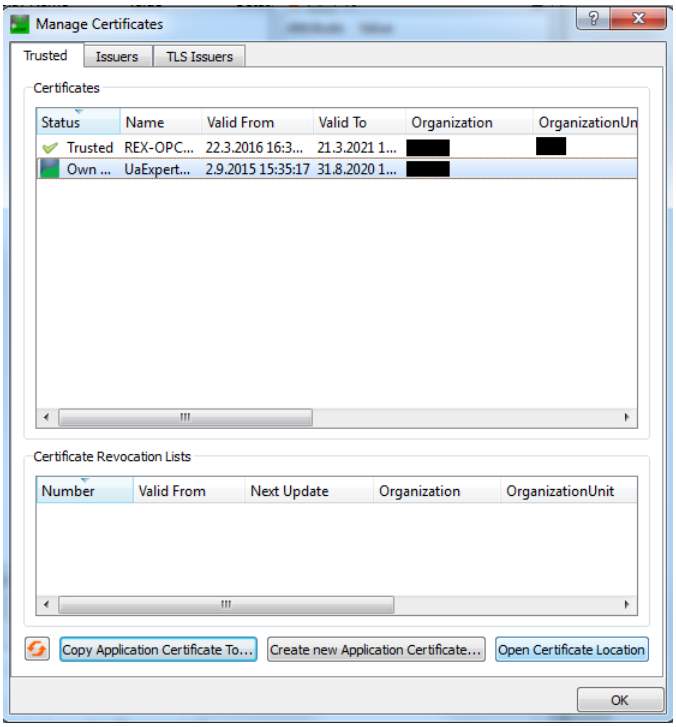
Obrázek 6.6: Nastavení uživatelské politiky na Endpoint

6.3.1 UaExpert

UaExpert je obecný plně funkční OPC UA klient, který je používán pro testování vyvíjených OPC UA serverů, pro zobrazení dat nebo pro použití pokročilých funkcí ze specifikace OPC UA. Tohoto klienta používá široká veřejnost jako standardizovanou aplikaci.

UaExpert umí tři druhy autentizace, zabezpečené přihlášení, Discovery služby, čtení, zápis a monitorování uzlů, zobrazení uzlů a jejich referencí pomocí stromové struktury, monitorování událostí, spouštění metod, nastavení a mnoho dalšího.

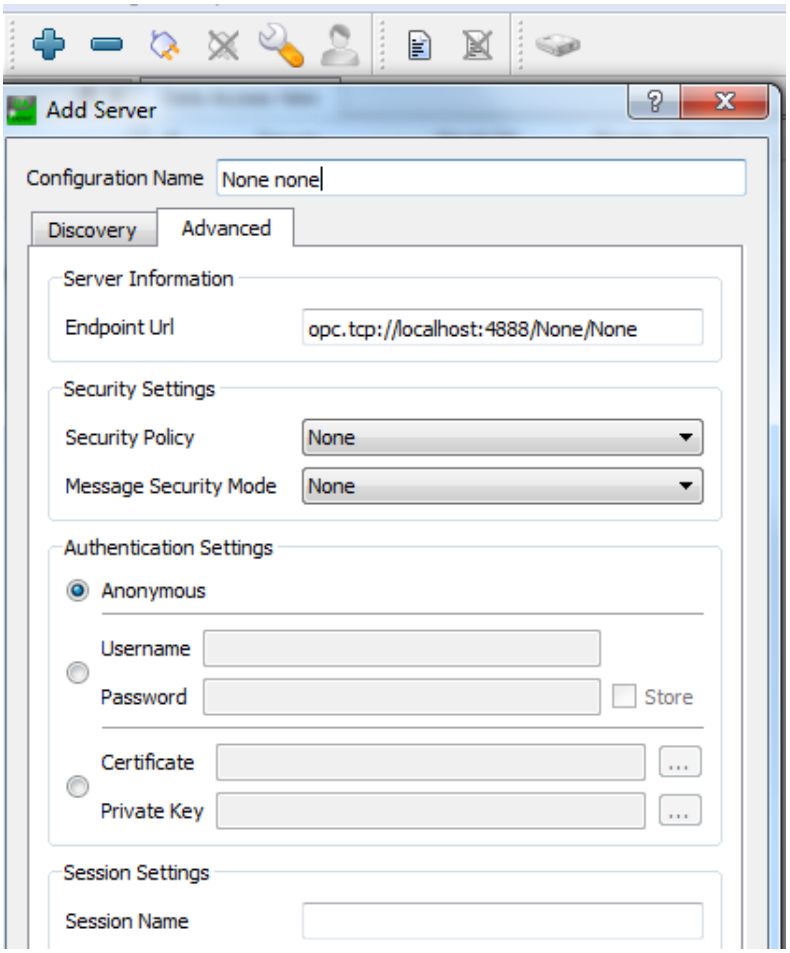
Při prvním spuštění program vyzve ke vygenerování aplikačního certifikátu. Pokud má klient komunikovat se serverem pomocí zabezpečeného připojení, musí být tento certifikát zkopírován do trust složky serveru, například pomocí 'Settings' > 'Manage Certificates' > 'Copy Application Certificate To...' (viz obrázek 6.7) a poté zvolte složku trust OPC UA serveru (tento postup samozřejmě nefunguje, pokud je server na jiném stroji než klient, pak je nutné certifikát zkopírovat ručně). Zkopírovat certifikát serveru do trust složky klienta není nutné. Pokud klient narazí na neznámý certifikát serveru, zeptá se zda mu má věřit. V dialogu pak navíc existuje tlačítko, pomocí něhož je možné zkopírovat certifikát do trust složky klienta a tím zajistit, že příště bude klient serveru věřit.



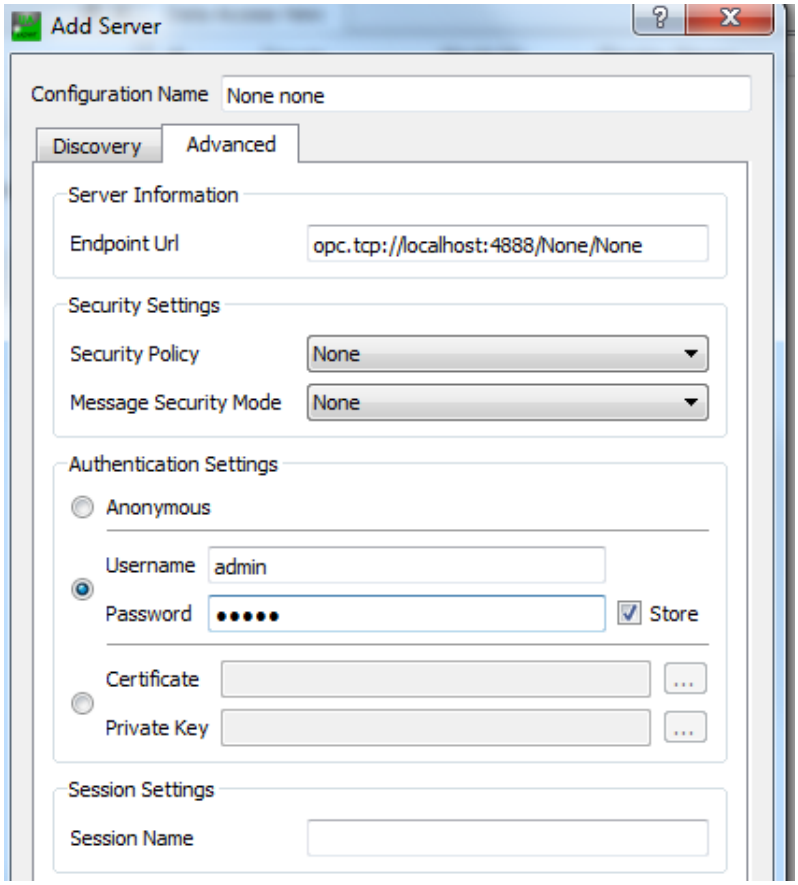
Obrázek 6.7: UaExpert: Uložení certifikátu do důvěryhodných

Připojení k serveru lze provést pomocí tlačítka plus. Otevře se konfigurace připojení. V záložce 'Advanced' (obrázek 6.8) je možné nastavit Endpoint, zabezpečení a přihlašo-

vací politiku ('Session name' nemá na chod vliv). Pokud je vybrána přihlašovací politika pomocí přihlašovacích údajů, je třeba zadat jméno a heslo uživatele (viz obrázek 6.9). Při přihlašování pomocí certifikátu je třeba zadat certifikát a soukromý klíč.

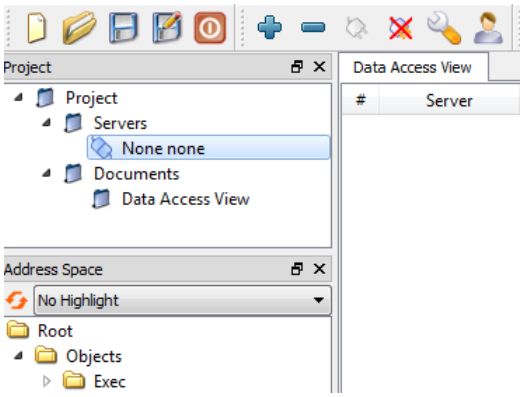


Obrázek 6.8: UaExpert: Připojení k serveru anonymně



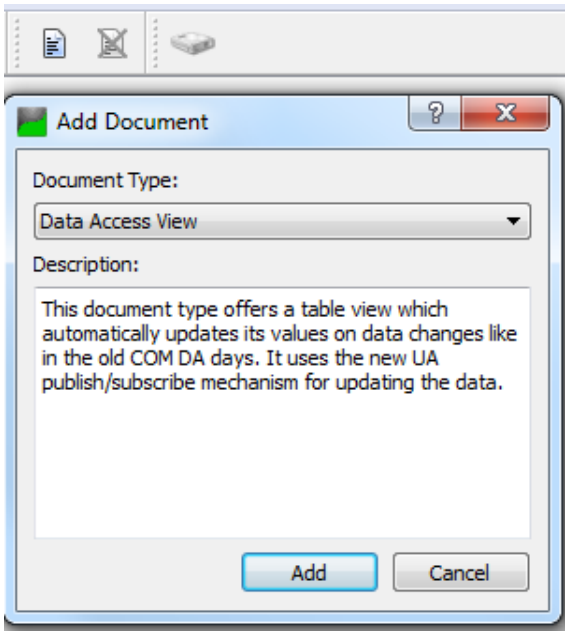
Obrázek 6.9: UaExpert: Připojení k serveru s přihlašovacími údaji

Funkční připojení je znázorněno zapojenou zástrčkou (viz obrázek 6.10). Připojení lze rozpojit (ikonka s přeškrtnutou zástrčkou) a znovu spojit (ikonka se zástrčkou). Změnu zabezpečení připojení a jinou konfiguraci (ikonka s klíčem) lze provést pouze s rozpojeným připojením. Změnu přihlašovací politiky je možné provést za běhu (pomocí ikonky uživatele). Klient může obsluhovat více připojení naráz. Konfiguraci klienta (připojení, monitorované položky apod.) lze uložit a při příštím použití jednoduše nahrát.

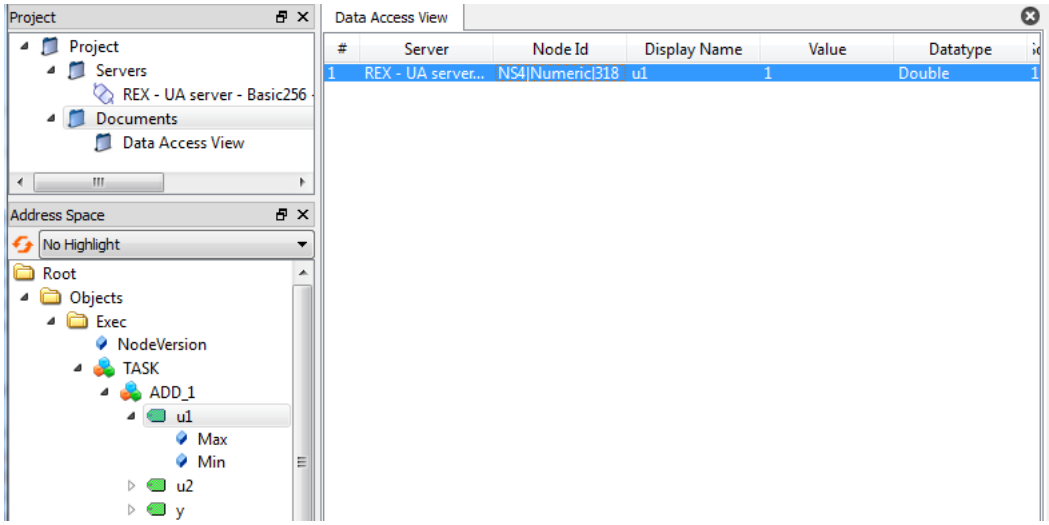


Obrázek 6.10: UaExpert: Připojený k serveru

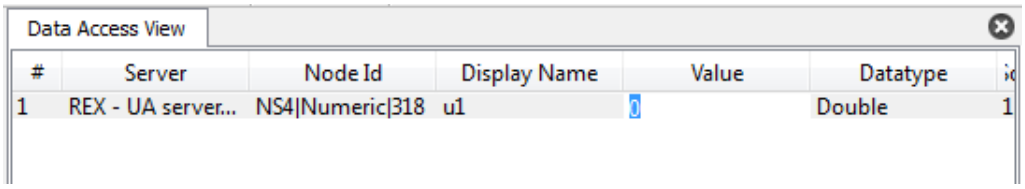
Pro monitorování hodnot uzlů je třeba přidat dokument 'Data Access View' (často je už přítomen) kliknutím na ikonku dokumentu, vybrání položky 'Data Access View' (viz obrázek 6.11) a kliknutím na 'Add'. Monitorované položky je třeba najít ve stromu Adresního prostoru a přetáhnout do prostoru dokumentu (viz obrázek 6.12). Položka se přidá do monitorovacího seznamu a jsou zde vidět pravidelné aktualizace hodnoty (pokud se monitorovaná hodnota mění). Položku lze poté kdykoliv smazat. Do položky v monitorovaném seznamu lze zapsat dvojklikem na hodnotu uzlu a zadáním nové hodnoty (viz obrázek 6.13).



Obrázek 6.11: UaExpert: Přidání monitorování dat

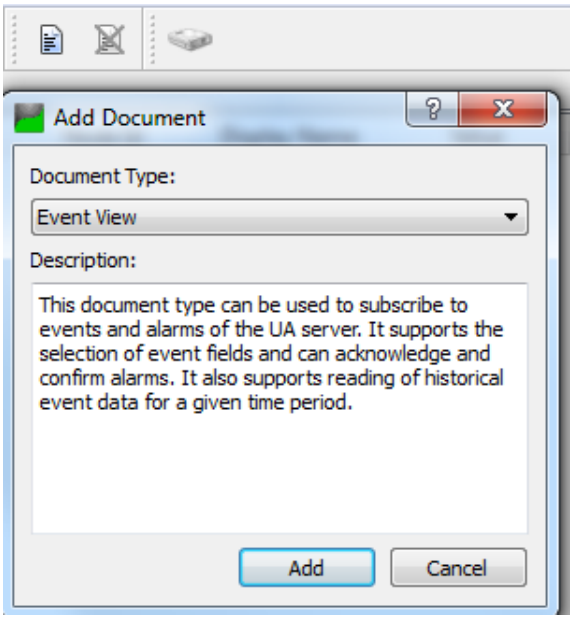


Obrázek 6.12: UaExpert: Monitorování proměnné u1

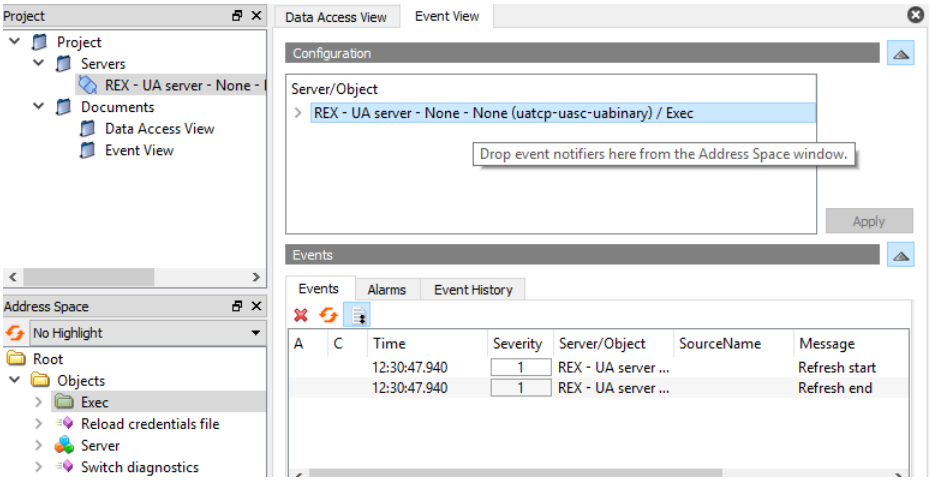


Obrázek 6.13: UaExpert: Zápis do proměnné u1

Pro monitorování událostí je třeba přidat dokument 'Event View' (obrázek 6.14) a do něj přidat monitorované uzly přetáhnutím ze stromu Adresního prostoru do prostoru 'Configuration' (viz obrázek 6.15). Všechny události uzlu poté ukazují v poli 'Events'. Při označení události se v poli 'Details' objeví detaily události. U OPC UA serveru pro REXYGEN je vhodné monitorovat složku 'Exec', popřípadě objekt 'Server', který je notifikován složkou 'Exec' (zobrazuje i její události).

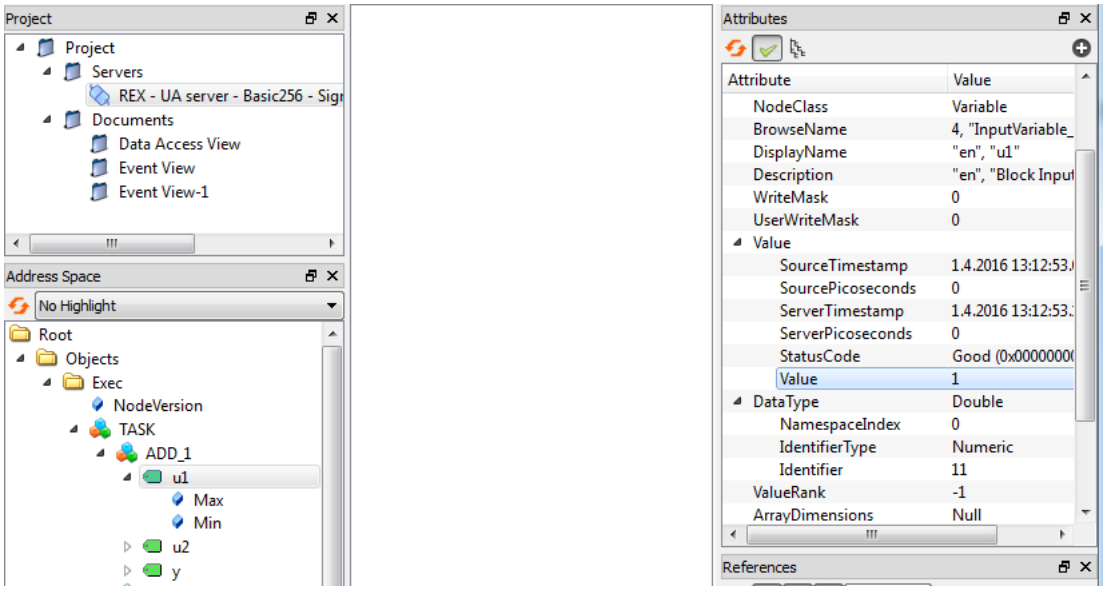


Obrázek 6.14: UaExpert: Přidání monitorování událostí

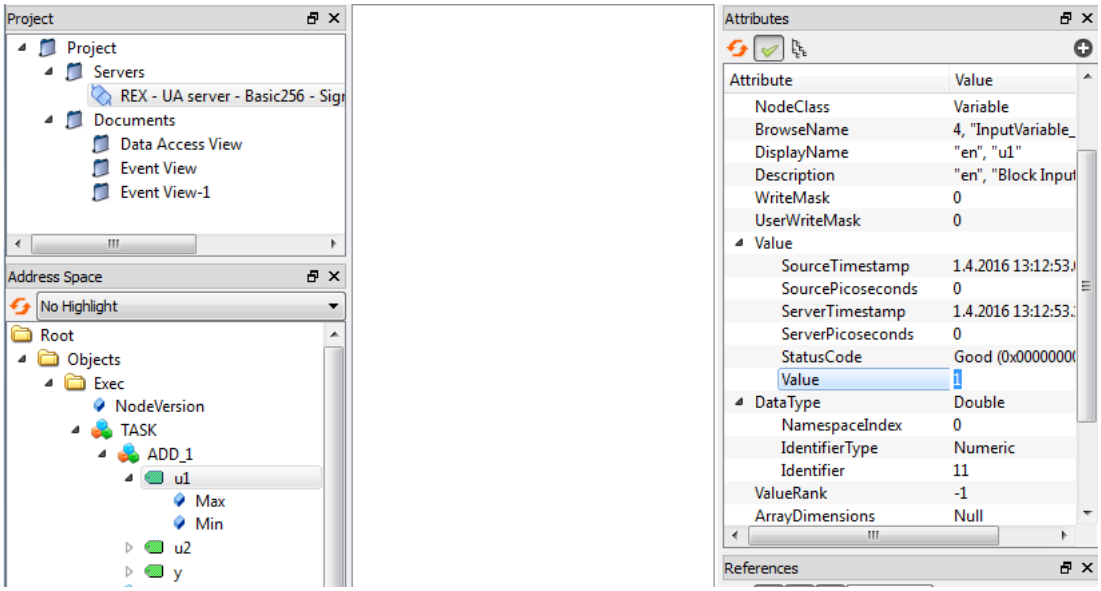


Obrázek 6.15: UaExpert: Monitorování událostí složky Exec

UaExpert umí i jednoduché čtení, kdy se při kliknutí na uzel ve stromě Adresního prostoru zobrazí v pravé části informace o uzlu. U hodnot proměnných se zobrazí i jejich hodnota (viz obrázek 6.16). Zápis do proměnné lze provést při dvojkliku na hodnotu value (viz obrázek 6.17).



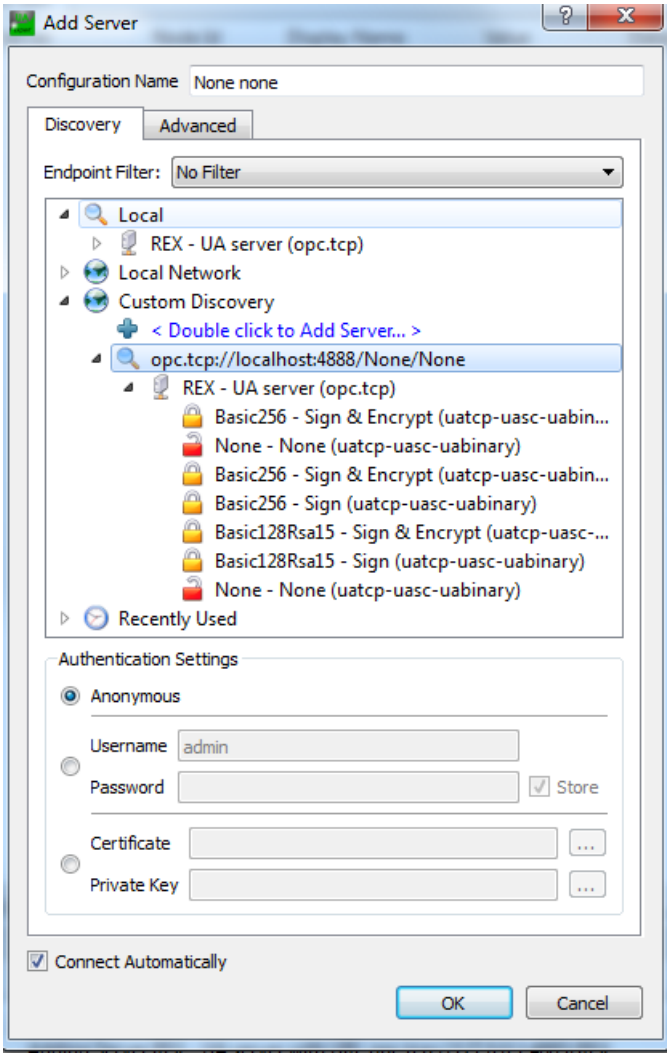
Obrázek 6.16: UaExpert: Čtení proměnné u1



Obrázek 6.17: UaExpert: Zápis do proměnné u1

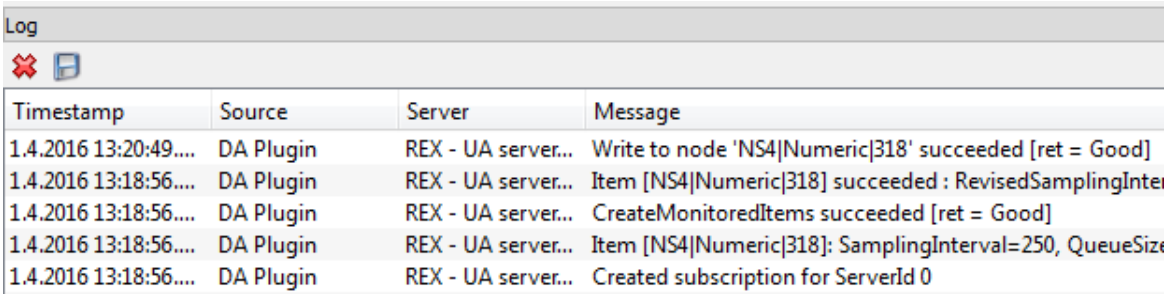
UaExpert implementuje i Discovery služby, pomocí nichž zobrazuje všechny dostupné Endpointy registrovaných serverů (obrázek 6.18). Uživateli pak stačí pouze rozbalit seznam příslušného serveru, vybrat jednu z možností, nastavit přihlašovací politiku a připo-

jit se. Klient vždy kontroluje LDS (Local Discovery Server - volně dostupný program), kde jsou zobrazeny všechny servery, které jsou zde zaregistrovány (viz kapitola 3.6). Druhou možností je přidání a prozkoumání vlastního Discovery serveru, například přímo OPC UA serveru pro REXYGEN, který podporuje Discovery služby a poskytuje informace o svých Endpointech.



Obrázek 6.18: UaExpert: Použití Discovery služeb

Pokud nastávají problémy při připojení, zápisu, čtení nebo při čemkoliv jiném, je dobré zkontrolovat logy aplikace, které jsou zobrazeny v dolním panelu (viz obrázek 6.19). Podle nahlášené chyby lze často snadno dohledat zdroj problému.

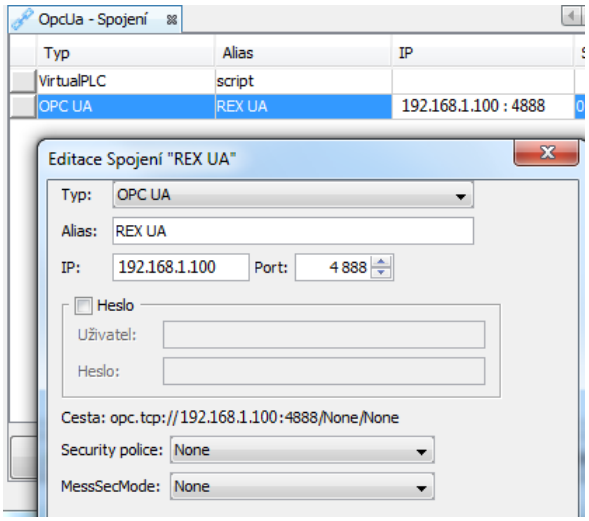


Obrázek 6.19: UaExpert: Logování akcí

6.3.2 myScada

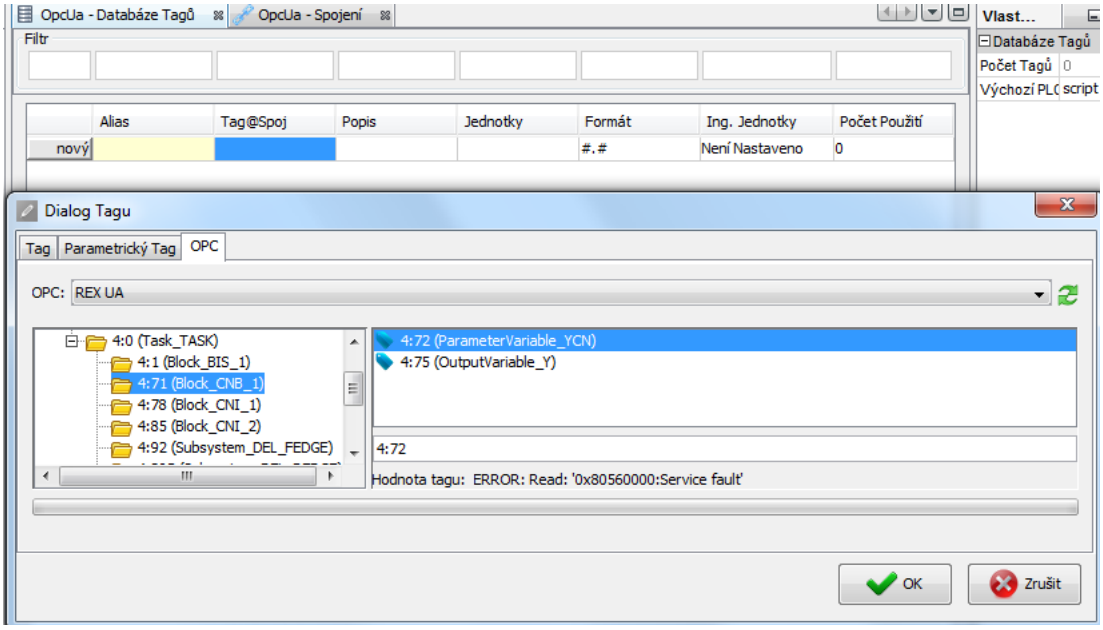
Program myScada umožňuje komunikovat pomocí OPC UA, vytvořit tagy, které jsou propojené s hodnotami uzlu serveru, a tyto tagy zobrazovat. Na rozdíl od programu UaExpert, myScada nevyužívá všechny možnosti specifikace OPC UA.

Pro použití OPC UA v myScada je třeba v programu myPROJECT designer vytvořit projekt, otevřít záložku spojení, přidat nové spojení a zvolit OPC UA. Otevře se dialog, v němž je možné nastavit spojení s OPC UA serverem (obrázek 6.20).

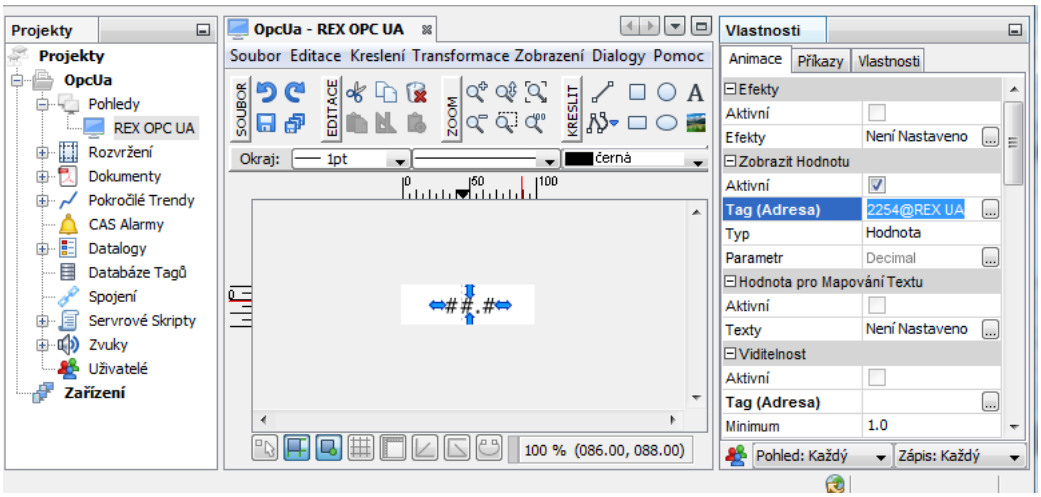


Obrázek 6.20: mySCADA: Anonymní přihlášení

V další fázi je třeba vytvořit v projektu tag, který bude směřovat na hodnotu některého uzlu na serveru (záložka OPC v dialogu pro vytváření tagů, obrázek 6.21). Hodnotu tohoto tagu lze na závěr využít v zobrazení aplikace (obrázek 6.22). Hotový projekt lze poté nahrát na zařízení a sledovat jeho chod pomocí programu myView.



Obrázek 6.21: mySCADA: Vytvoření tagu z uzlu OPC UA serveru



Obrázek 6.22: mySCADA: Použití tagu v zobrazení

Program myView slouží k zobrazení dat na zařízení. Pokud je použit v zobrazení tag propojený s OPC UA, bude se jeho hodnota pravidelně měnit podle hodnoty v OPC UA serveru (obrázek 6.23). Pokud se přeruší spojení nebo serveru dojdou zdroje, označí zařízení hodnotu v zobrazení červeně (viz obrázek 6.24).

Obrázek 6.23: mySCADA: Zobrazení dat v zařízení

Obrázek 6.24: mySCADA: Zařízení bez dat

Jelikož se myScada neřídí přesně specifikací OPC UA, je nutné upravit server tak, aby se k němu mohl klient připojit. Pro nezabezpečené přihlášení musí mít server Endpoint s koncovou URI '/None/None'. Pro anonymní přihlášení musí být ID přihlašovací politiky '0' (viz [Endpoint:2] a ADMIN_USER_TOKEN_POLICY_ID na obrázku 6.4).

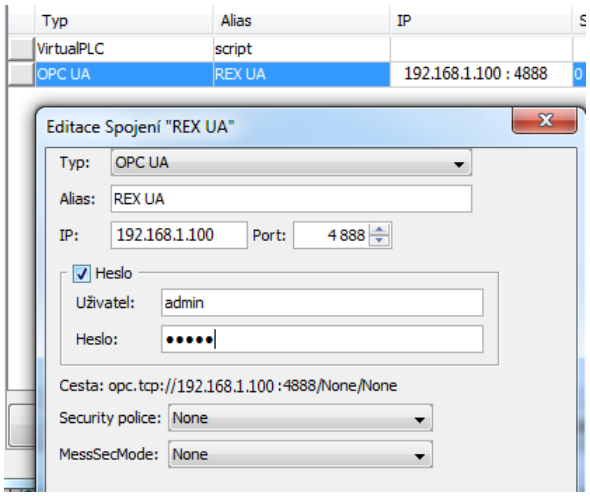
Pro nastavení připojení pomocí přihlašovacích údajů je třeba nastavit ID patřičné politiky na 'UserNameIdentityToken' (obrázek 6.25 a 6.26) a zadat uživatelské jméno a heslo v dialogu pro úpravu spojení (viz obrázek 6.27).

```
[AUTH]
;file with usernames and passwords and user token id for
CREDENTIALS_INI_PATH=RexOpcUa_users.ini
CREDENTIALS_USER_TOKEN_POLICY_ID=UserNameIdentityToken
```

Obrázek 6.25: mySCADA: Nastavení přihlašování pomocí přihlašovacích údajů

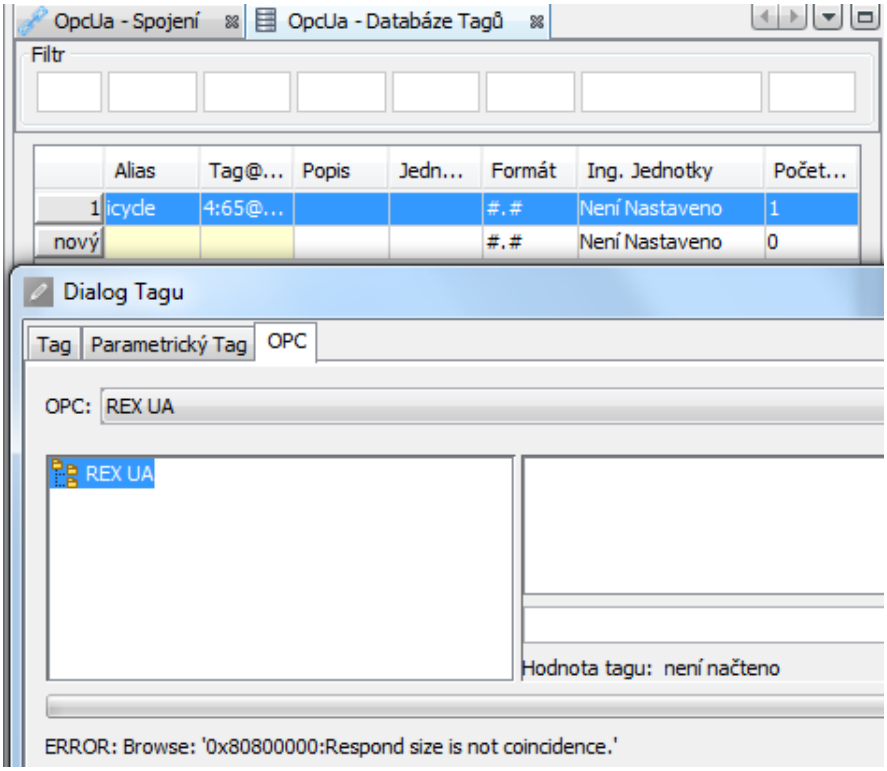
```
[ENDPOINT:2]
SECURITY_POLICY=[None,Sign_Basic128Rsa15,SignEncrypt_Basic128Rsa15,Sign_Basic256,SignEncrypt_Basic256]
USER_TOKEN_POLICY_ID=[UserNameIdentityToken]
;additional endpoint url is optional
URL=opc.tcp://localhost:4888/None/None
```

Obrázek 6.26: mySCADA: Nastavení uživatelské politiky na Endpoint



Obrázek 6.27: mySCADA: Přihlášení pomocí přihlašovacích údajů

Při práci s myScada je doporučeno **nemířit na** OPC UA server pomocí **localhost**u, **nepoužívat zabezpečenou komunikaci** a **neomezovat zdroje**. Případně zdrojů poskytnout dostatek a nastavit parametr MAX_SESSION_TIMEOUT dostatečně krátký, jinak může dojít k vyčerpání zdrojů a server začne zobrazovat chyby. Chyby připojení se během návrhu nejlépe zjistí při vytváření tagů (viz obrázek **??**).



Obrázek 6.28: mySCADA: Chyba připojení

Literatura