



www.rexcontrols.cz/rex

OPC UA server systému REX

Uživatelská příručka

REX Controls s.r.o.

Verze 2.50.1

Plzeň

7.11.2016

Obsah

| | | |
|----------|-----------------------------------|-----------|
| 1 | OPC UA a systém REX | 2 |
| 1.1 | Úvod | 2 |
| 1.2 | Funkce serveru | 3 |
| 2 | Adresní prostor | 5 |
| 2.1 | Metody | 5 |
| 2.2 | Bloky | 6 |
| 2.3 | Parametry | 7 |
| 2.4 | Události a verzování | 7 |
| 3 | Konfigurace | 9 |
| 3.1 | Rex | 9 |
| 3.2 | Application | 10 |
| 3.3 | Security | 12 |
| 3.4 | Auth | 13 |
| 3.5 | Endpoint | 15 |
| 3.6 | Discovery | 16 |
| 3.7 | Options | 17 |
| 4 | Autentifikace a autorizace | 20 |
| 4.1 | RexOpcUaAuth | 20 |
| 4.2 | RexSecurityConfig | 21 |
| 5 | Návod ke spuštění | 25 |
| 5.1 | OPC UA Klienti | 27 |
| 5.1.1 | UaExpert | 29 |
| 5.1.2 | myScada | 41 |
| | Literatura | 47 |

Kapitola 1

OPC UA a systém REX

OPC UA je nová forma komunikace, která je určena hlavně 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.

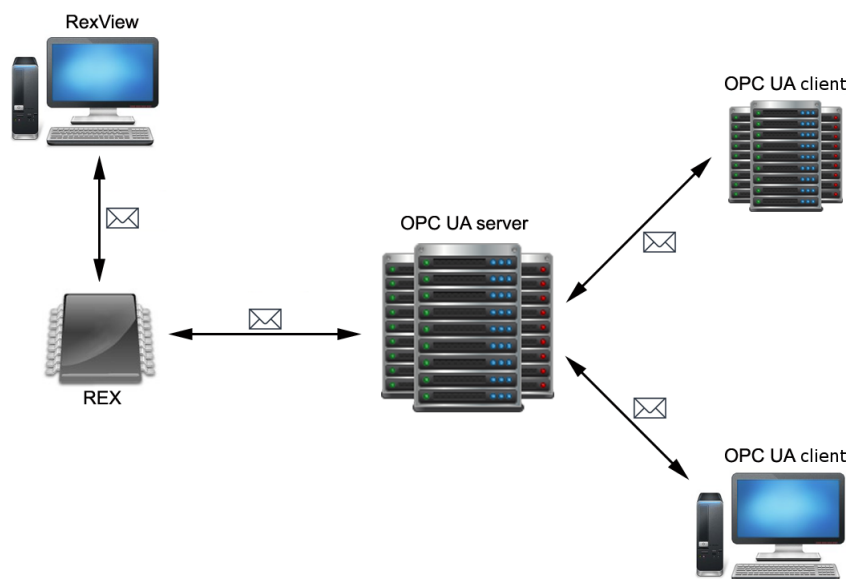
OPC UA není vhodné pro vytváření spojení mezi řídicími jednotkami v reálném čase, ale je použitelné pro téměř reálný čas. Jeho hlavní použití je však v propojení různých aplikací, ve vytváření “Internet of Things” a průmyslové revoluci 4.0.

1.1 Úvod

OPC UA server pro REX je samostatná aplikace, která komunikuje s exekutivou REXu pomocí diagnostického protokolu. Není nutné, aby byl server na stejné výpočetní jednotce jako REX, ale je vhodné co nejvíce zkracovat jejich vzájemnou dobu odezvy. Je výhodnější, aby byl server z pohledu síťového připojení co nejbližší běžící instanci REXu, 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 řídicích jednotek běžících v reálném čase. Propojení serveru a ostatních zúčastněných aplikací je zobrazeno na obrázku [1.1](#).

OPC UA server se vždy připojuje pouze k jedné exekutivě REXu. Na jednom stroji ovšem může běžet více instancí serveru, kde každý musí mít vlastní konfiguraci, aby si vzájemně neblokovaly porty Endpointu.

Server je možné využívat v podobě dema, které běží maximálně jednu hodinu. Pro odemknutí serveru je třeba získat licenci a tu klasicky nainstalovat na cílové zařízení například v programu REX Draw nebo vložit jako hodnotu parametru `REX_LICENCE_KEY` v konfiguračním souboru (viz kapitola [3.1](#)).



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

1.2 Funkce serveru

Běžící OPC UA server je připojen k exekutivě REXu a ve svém Adresním prostoru zobrazuje všechny její bloky i s parametry. Struktura Adresního prostoru je podobná struktuře úkolů v programu RexView. Po připojení k REXu 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. Aplikace by měla být ukončena zavoláním metody “Shutdown” (viz kapitola 2.1).

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 REXu k výměně exekutivy, server smaže a znovu nahraje strukturu bloků a vytvoří událost o změně Adresního prostoru.

Server je nastavitelný pomocí INI konfiguračního souboru, jehož absolutní umístění lze zadat jako parametr při spuštění serveru. Pokud existuje nějaký přednastavený konfigurační soubor, je parametr nepovinný. Konfigurace je popsána v kapitole 3.

RexOpcUa [-c <configFile>]

V OS Windows je možné přednastavit konfigurační soubor v registrech. Pokud je aplikace poté spuštěná bez parametru, použije se přednastavený konfigurační soubor.

Tento registr lze nastavit pomocí příkazové řádky, kde se spustí server s parametrem `-i` a cestou k novému konfiguračnímu souboru. Tento příkaz je popsán níže, *configFile* je nová cesta ke konfiguračnímu souboru:

RexOpcUa -i <configFile>

V Linuxu je přednastavená cesta ke konfiguračnímu souboru pevně daná:

/rex/OpcUa/RexOpcUa.ini

Kapitola 2

Adresní prostor

Adresní prostor serveru obsahuje všechny povinné OPC UA uzly a zároveň i nové typy uzlů, které byly vytvořeny pro REX. Dále obsahuje metody pro manipulaci se serverem a složku ‘Exec’, která obsahuje stromovou strukturu úkolů REXu i se subsystémy a bloky, a to včetně úkolů připojených k ovladačům. Vše kromě obsahu složky ‘Exec’ je vytvořeno při startu serveru. Obsah složky ‘Exec’ je vytvořen při novém připojení k REXu nebo při přehrání exekutivy. Adresní prostor s exekutivou je zobrazen na obrázku 2.1 pomocí OPC UA klienta UaExpert (viz kapitola 5.1.1).

Server používá čtyři vlastní jmenné prostory (Namespace). První Namespace odpovídá URI aplikace (viz tabulka 3.2) a používá se pro chod serveru samotného. Namespace *urn:Rx:TypeDeclaration* se používá pro definici typů, kterými se popisují bloky a parametry exekutivy. Namespace *urn:Rx:Server* obsahuje uzly, které slouží k obsluze serveru, například metody pro správu serveru a složku ‘Exec’. Namespace exekutivy je unikátní pro každou nahranou exekutivu nebo instanci REXu a je popsán v kapitole 3.2. Namespace exekutivy obsahuje všechny uzly úkolů, bloků a parametrů.

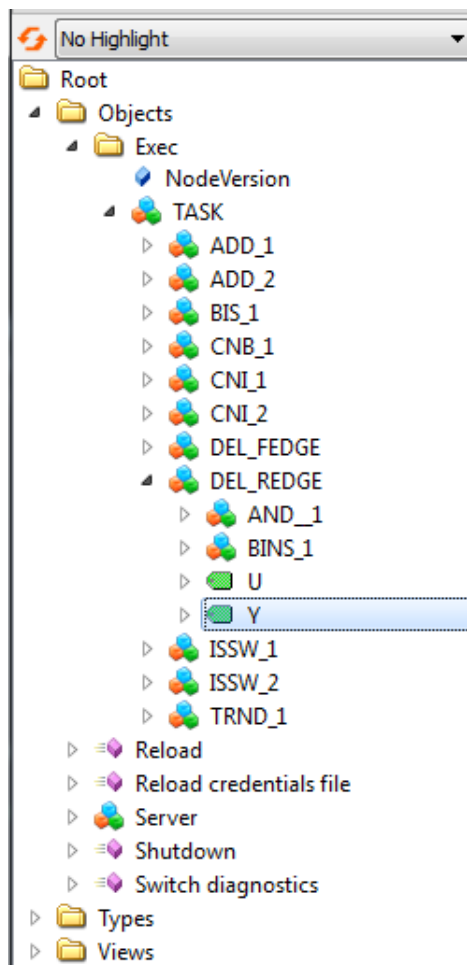
2.1 Metody

Server obsahuje OPC UA metody, pomocí kterých může klient serverem manipulovat. Spouštění těchto metod je povoleno pouze klientům s právy administrátora.

První metodou je metoda ‘Reload’, pomocí které lze serveru explicitně nařídit smazání a opětovné načtení stromové struktury úkolů. Proces je shodný s procesem, který nastane při přehrání exekutivy v REXu.

Pomocí metody ‘Reload credentials file’ lze zajistit, aby server znovu načetl INI soubor s přihlašovacími údaji a aktualizoval tak databázi uživatelů (viz kapitola 4). Pokud server nemá specifikovanou cestu k souboru, vrátí chybový kód `BadNotSupported`.

Metoda ‘Switch diagnostics’ umožňuje vypnout a zapnout tvorbu diagnostických dat. Ty jsou poté přístupny ve standardním místě podle specifikace OPC UA: jsou součástí objektu `Server`. Tuto metodu lze zakázat pomocí parametru `ALLOW_SWITCH_DIAGNOSTICS` v konfiguraci. Samotné generování diagnostiky, které začne již při spuštění serveru, lze nastavit pomocí parametru `ENABLE_DIAGNOSTICS`.



Obrázek 2.1: Adresní prostor zobrazený klientem UaExpert

Metoda ‘Shutdown’ ukončí server. Tato metoda je proti nechtěnému provedení chráněna heslem, které je uloženo v konfiguračním parametru `SHUTDOWN_PASSWORD` (viz tabulka 3.2). Toto heslo musí klient zadat při zavolání metody. Spuštění metody ‘Shutdown’ pomocí klienta je doporučený způsob ukončení serveru.

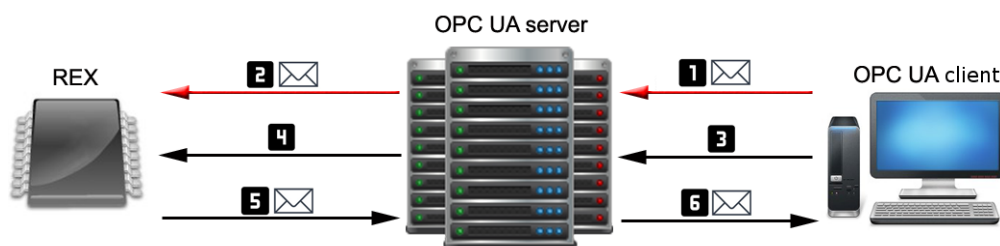
2.2 Bloky

Struktura bloků v Adresním prostoru odpovídá struktuře bloků v exekutivě REXu. Všechny bloky používají Namespace exekutivy (viz kapitola 3.2), jejich `BrowseName` a `DisplayName` odpovídá názvu v REXu (u `BrowseName` je jako předpona uveden typ uzlu) a v popisu je uložen typ uzlu. Bloky obsahují parametry, které jsou shodné s těmi v REXu.

Kromě klasických bloků (BlockType) existuje speciální blok Trend (TrendBlockType). V této verzi není mezi trendem a klasickým blokem žádný další rozdíl. Subsystem (SubsystemType) rozšiřuje klasický blok a kromě vlastních parametrů může obsahovat i další bloky nebo subsystemy. Úkoly (TaskType) je subsystem, který nemůže být navázán na žádný jiný subsystem. Ve stromové struktuře jde o kořenový blok. Všechny úkoly jsou umístěny ve složce 'Exec'.

2.3 Parametry

Při vytváření stromové struktury jsou vytvořeny parametry 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 REXu. 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].

Parametry bloků (IRexVariableType) rozšiřují klasické OPC UA proměnné, jejich součástí je minimální a maximální přípustná hodnota, které jsou zapsané v uzlech Min a Max. Datový typ hodnoty proměnné odpovídá datovému typu proměnné v REXu. BrowseName a DisplayName proměnné odpovídají názvu proměnné v REXu, u BrowseName je jako předpona uveden typ uzlu.

Proměnné se dělí na vstupy, výstupy, parametry a stavy. Pro každou z těchto skupin je vytvořen speciální typ uzlu (ParameterVariableType, StateVariableType, InputVariableType, OutputVariableType) s tím, že do stavů a výstupů nelze zapisovat.

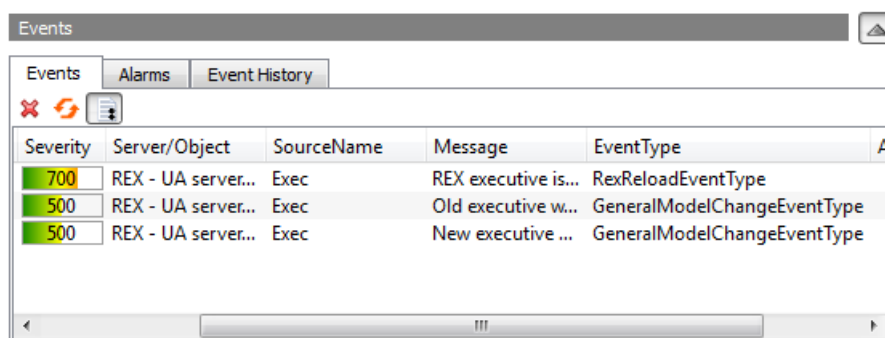
Poznámka: V této verzi OPC UA serveru nejsou synchronizovány pole a trendy.

2.4 Události a verzování

Aby mohly probíhat načítat a mazat bloky nové exekutivy, musí být složka 'Exec' verzovaná. Při každé změně struktury se nastaví aktuální datum jako její verze a je na ní

vyvolána událost ‘GeneralModelChangeEvent’ ve které jsou všechny odebrané a přidané uzly. Událost je při změně exekutivy volána dvakrát, nejprve při odstranění starých a poté při přidání nových bloků. Důvodem je úspora paměti serveru.

Server navíc poskytuje i vlastní události odvozené od třídy ‘DeviceFailureEventType’, které jsou vytvářeny na uzlu ‘Exec’. Událost ‘RexDisconnectEvent’ je vytvořena pokud je ztraceno spojení s REXem. Událost ‘RexReconnectEvent’ je vytvořena při opětovném připojení k REXu. Událost ‘RexReloadEvent’ je vytvořena, pokud je přehrána exekutiva (manuálně či při změně exekutivy v REXu). Zobrazení posloupnosti vytvořených událostí při manuálním přehrávání exekutivy je zobrazeno na obrázku 2.3.



| Severity | Server/Object | SourceName | Message | EventType |
|----------|--------------------|------------|---------------------|-------------------------|
| 700 | REX - UA server... | Exec | REX executive is... | RexReloadEventType |
| 500 | REX - UA server... | Exec | Old executive w... | GeneralModelChangeEvent |
| 500 | REX - UA server... | Exec | New executive ... | GeneralModelChangeEvent |

Obrázek 2.3: Zobrazení událostí serveru klientem UaExpert

Kapitola 3

Konfigurace

Konfigurační INI soubor může obsahovat pouze ASCI znaky, doporučuje se používat kódování UTF-8 a záleží na velikosti písmen. Na konci a začátku řádků a kolem znaménka “=” 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, které nebudou mít nastavenou hodnotu, nebudou brány v potaz.

V konfiguraci mohou být pouze ty sekce, které mají stejný název (velkými písmeny) jako název některé z následujících podkapitol. Pokud je sekce v INI souboru dvakrát, data se jednoduše doplní nebo přepíšu. Nelze však přepisovat prázdnou hodnotou, taková hodnota se ignoruje. Pouze sekce Endpoint může mít podsekce, pro každou podsekcí bude vytvořen jeden Endpoint.

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 středníkem [1;2;text]. Prázdné pole se chová, jako by hodnota nebyla vyplněna. Soubor značí systémovou cestu k souboru a složka cestu ke složce. Pokud není zapnutá relativní cesta pomocí proměnné APPLICATION_USE_RELATIVE_PATH popsané v tabulce 3.2, jsou tyto cesty absolutní. Jinak se použije cesta relativně k umístění konfiguračního souboru.

3.1 Rex

Tato sekce obsahuje parametry, které se týkají spojení serveru s REXem, včetně intervalů pro navazování spojení. Podrobnosti jsou vypsány v tabulce 3.1.

Tabulka 3.1: Nastavení spojení s REXem

| Pole | Hodnota | Původní hodnota | Popis |
|----------------------------|-----------|-----------------|---|
| REX_ADDRESS | IP adresa | – | IP adresa (DNS) běžící exekutivy REXu, ke které se má server připojit. |
| SYNC_INTERVAL | Číslo | 500 | Interval v ms, v kterém má server synchronizovat monitorované položky s REXem. Je vhodné mít tuto hodnotu nižší než minimální interval pro monitorování uzlů. |
| REX_TCP_RECONNECT_INTERVAL | Číslo | 5000 | Interval v ms pro čekání mezi pokusy o znovunavázání spojení. |
| MAX_RECONNECT_BEFORE_STOP | Číslo | 3 | Po kolika pokusech o znovunavázání spojení se server vypne. |
| REX_TCP_IDLE_INTERVAL | Číslo | 30000 | Interval v ms pro obnovu spojení s REXem, aby se neuzavřelo. Toto číslo by mělo být dostatečně menší než 1 minuta. |
| REX_TCP_CONNECT_INTERVAL | Číslo | 30000 | Interval v ms pro čekání mezi pokusy o navázání nového spojení. |

3.2 Application

Tato sekce obsahuje hlavní údaje o serveru, viz tabulka 3.2. Zde se nastavuje i Namespace serveru a exekutivy. Namespace exekutivy odpovídá následujícímu tvaru:

urn:Rex:Exec:<COMPANY_NAME>:<PROJECT_NAME>:<INSTANCE_NAME>

Parametry COMPANY_NAME, PROJECT_NAME a INSTANCE_NAME volte tak, aby jejich kombinace byla unikátní pro každou běžící instanci REXu, aby tak nedocházelo k chybám při použití více OPC UA serverů. Podle specifikace OPC UA by více serverů připojených k jedné instanci REXu mělo mít stejné názvy, servery připojené k různým instancím REXu musí mít různé názvy.

Každá běžící exekutiva má jiný čas spuštění. Pokud je zapnutý parametr NAME_SPACE_WITH_EXEC_START_TIME, používá se čas startu exekutivy v jejím Namespace. Tím se zajistí, že klient rozliší novou exekutivu a nebude používat dříve načtené položky. Pokud se v REXu používá pouze jedna exekutiva a restartuje se, není vhodné tuto funkci používat, protože se v praxi jedná o stejnou exekutivu, tudíž stejná data.

*urn:Rex:Exec:<COMPANY_NAME>:<PROJECT_NAME>:<INSTANCE_NAME>:
<Čas startu exekutivy>*

Pokud má server běžet na stroji, který nemá nainstalovanou licenci REXu, lze použít parametr REX_LICENCE_KEY, který by měl obsahovat licenci REXu tak, jako je například v programu REX View. Pokud tato licence chybí nebo v ní není povolen OPC UA server, server po hodině přestane fungovat (na systému Windows se objeví varovná hláška). Pokud server při startu nezjistí aktivní licenci, vypíše ‘Site code’ stroje, na kterém běží. Každý stroj má svůj vlastní a pro tento ‘Site code’ je tedy poté třeba koupit licenci, aby na daném stroji mohl server běžet v plné verzi.

Tabulka 3.2: Nastavení aplikace

| Pole | Hodnota | Původní hodnota | Popis |
|----------------------------------|--------------|-----------------|--|
| USE_RELATIVE_PATH | Y/N | N | Pokud je zapnuté, všechny soubory a složky se uvažují relativně od konfiguračního souboru, jinak jsou všechny cesty absolutní. |
| 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. |
| REX_LICENCE_KEY | Licence REXu | – | (Volitelné) Licence REXu, v níž je povolen OPC UA server. Pokud není dodána, server běží v demo modu (1 hodinu). |
| COMPANY_NAME | Text | – | Tento text bude částí Namespace exekutivy. |
| PROJECT_NAME | Text | – | Tento text bude částí Namespace exekutivy. |
| INSTANCE_NAME | Text | – | Tento text bude částí Namespace exekutivy. |
| NAMESPACE_WITH_EXEC_START_TIME | Y/N | N | Použije čas startu exekutivy jako část Namespace exekutivy. Každá nahraná exekutiva REXu je z pohledu OPC UA unikátní. |
| SHUTDOWN_PASSWORD | Text | – | (Volitelné) Heslo pro vypnutí serveru metodou ‘Shutdown’. |

3.3 Security

Sekce security obsahuje nastavení uložení a validace klientských certifikátů. Pokud všechny Endpointy mají nastavené zabezpečení komunikace pouze na None a není použito přihlášení pomocí certifikátu (viz kapitola 3.4), je celá tato sekce volitelná. Server využívá

OpenSSL, proxy certifikáty jsou zakázány.

Pro vytvoření certifikátů a adresářů pro klientské certifikáty lze využít aplikaci Rex-SecurityConfig, která je popsána v kapitole 4.2.

Tabulka 3.3: Zabezpečení

| Pole | Hodnota | Původní 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 | – | (Nepoužito) 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 zneplatnění certifikátů. <table><tr><td>N</td><td>Žádná kontrola</td></tr><tr><td>L</td><td>Pouze listy</td></tr><tr><td>S</td><td>Bez sebou–podepsaných</td></tr><tr><td>A</td><td>Všechny</td></tr></table> | N | Žádná kontrola | L | Pouze listy | S | Bez sebou–podepsaných | A | Všechny |
| 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 Auth

Sekce Auth se zabývá možnostmi autorizaci a autentifikaci klientů při připojení k Endpointu, parametry jsou popsány v tabulce 3.4. Úprava přihlašovacích údajů a pravomoci rolí jsou popsány v kapitole 4.1.

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 poskytnout uživatelské jméno a heslo, které se poté validují na serveru. Při použití certifikátu musí klient poskytnout certifikát a klíč, u kterých server ověří důvěryhodnost (certifikát je uložen v seznamu trust, viz tabulka 3.3).

Endpoint vždy obsahuje seznam přihlašovacích politik, které podporuje, v podobě textových identifikátorů. Pokud má Endpoint podporovat některou z politik, musí jí být v této sekci přiřazeno textové ID a v konfiguraci Endpointu musí být stejné ID použito v poli `USER_TOKEN_POLICY_ID`.

Server obsahuje tři anonymní politiky (`ADMIN_USER_TOKEN_POLICY_ID`, `OPERATOR_USER_TOKEN_POLICY_ID`, `GUEST_USER_TOKEN_POLICY_ID`), které se liší tím, jakou roli přiřadí uživateli, který se s ní přihlásí. Je možné, aby Endpoint podporoval více anonymních politik, záleží ovšem poté na klientu, kterou z nich vybere.

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 ty hesla, která byla zakódována s tímto parametrem, při výměně je tedy nutné přegenerovat soubor s přihlašovacími údaji.

Server podporuje tři přihlašovací politiky s certifikátem (`CERT_ADMIN_USER_TOKEN_POLICY_ID`, `CERT_OPERATOR_USER_TOKEN_POLICY_ID`, `CERT_GUEST_USER_TOKEN_POLICY_ID`). Při přihlášení pomocí certifikátu server zjistí, zda daný certifikát vede jako důvěryhodný, a pokud ano, přiřadí uživateli korespondující práva.

Tabulka 3.4: Autorizace a autentifikace

| Pole | Hodnota | Původní hodnota | Popis |
|------------------------------------|-------------|-----------------|--|
| ADMIN_USER_TOKEN_POLICY_ID | ID politiky | – | (Volitelné) ID anonymní přihlašovací politiky, všichni uživatelé budou mít administrátorská oprávnění. |
| OPERATOR_USER_TOKEN_POLICY_ID | ID politiky | – | (Volitelné) ID anonymní přihlašovací politiky, všichni uživatelé budou mít operátorská oprávnění. |
| GUEST_USER_TOKEN_POLICY_ID | ID politiky | – | (Volitelné) ID anonymní přihlašovací politiky, všichni uživatelé budou mít oprávnění hosta. |
| CERT_ADMIN_USER_TOKEN_POLICY_ID | ID politiky | – | (Volitelné) ID politiky přihlašování pomocí certifikátu, všichni uživatelé budou mít administrátorská oprávnění. |
| CERT_OPERATOR_USER_TOKEN_POLICY_ID | ID politiky | – | (Volitelné) ID politiky přihlašování pomocí certifikátu, všichni uživatelé budou mít operátorská oprávnění. |
| CERT_GUEST_USER_TOKEN_POLICY_ID | ID politiky | – | (Volitelné) ID politiky přihlašování pomocí certifikátu, všichni uživatelé budou mít oprávnění hosta. |
| CREDENTIALS_USER_TOKEN_POLICY_ID | ID politiky | – | ID politiky přihlašování pomocí jména a hesla. Volitelné, pokud není zadán parametr CREDENTIALS_INI_PATH. |
| CREDENTIALS_INI_PATH | Soubor | – | Soubor, kde jsou zapsáni uživatelé s heslem a rolí. Volitelné, pokud není zadán parametr CREDENTIALS_USER_TOKEN_POLICY_ID. |
| OPTIONAL_ENCODING_SALT | Text | q1we58 | Hodnota, pomocí které bude zakódováno heslo v souboru s uživateli. Tato hodnota nemusí být příliš velká, 5 - 20 ASCII znaků stačí. |

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 3.5.

Pokud je potřeba využít Endpoint pro služby Discovery, doporučujeme v URL Endpointu nepoužívat localhost, ale veřejnou IP adresu. V opačném případě klient, který bude na Endpointu volat služby Discovery a v získaných datech nezmění adresu stroje, 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 3.5: Nastavení Endpointu

| Pole | Hodnota | Původní hodnota | Popis |
|----------------------|-----------------------------------|-----------------|---|
| URL | URL Endpointu | – | URL Endpointu pro připojení pomocí protokolu opc.tcp. |
| SECURITY_POLICY | [Zabezpečení] (pole) | – | Povolené zabezpečení komunikace - detaily v tabulce 3.6. |
| USER_TOKEN_POLICY_ID | [ID přihlašovací politiky] (pole) | – | Použité uživatelské politiky. ID politik se nastavují pomocí tabulky 3.4. |

Tabulka 3.6: Zabezpečení komunikace

| Zabezpečení | Znak | Podpis | Šifrování | Algoritmus |
|--------------|------|--------|-----------|---------------|
| Žádné | N | Ne | Ne | – |
| Nízké | L | Ano | Ne | Basic128Rsa15 |
| Střední | M | Ano | Ano | Basic128Rsa15 |
| Vysoké | H | Ano | Ne | Basic256 |
| Velmi vysoké | V | Ano | Ano | Basic256 |

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 a všechny adresy 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 3.7.

Tabulka 3.7: Nastavení registrace k průzkumnému serveru

| Pole | Hodnota | Původní hodnota | Popis |
|--------------------------------------|---------------------------|-----------------|---|
| <code>ENDPOINT_URL</code> | [URL Endpointu] (pole) | – | (Volitelné) URL registrovaného Endpointu. Měla by být shodná s URL některého z Endpointů. |
| <code>SERVER_CERTIFICATE_PATH</code> | Soubor | – | Cesta k certifikátu průzkumného serveru. |
| <code>SERVER_URL</code> | URL | – | URL průzkumného serveru, u nějž se bude server registrovat. URL musí začínat <i>opc.tcp://</i> . |
| <code>SECURITY_POLICY</code> | N,L,M,H,V | – | Použité zabezpečení při komunikaci s průzkumným serverem - detaily v tabulce 3.6. Lze použít právě jedno zabezpečení, které průzkumný server podporuje. |
| <code>REFRESH_TIME</code> | 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 3.8 a 3.9.

Tabulka 3.8: Obecné nastavení

| Pole | Hodnota | Původní 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 <i>TooManyPublishRequests</i> . |
| 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 3.9: Obecné nastavení

| Pole | Hodnota | Původní hodnota | Popis |
|--|---------|-----------------|--|
| ENABLE_ DIAGNOSTICS | Y/N | N | Server vytváří diagnostická data. |
| ALLOW_SWITCH_ DIAGNOSTICS | Y/N | N | Povolit volání metody pro zapnutí/vypnutí diagnostiky. |
| MAX_SESSIONS | Číslo | 10 | Maximální počet relací na server, 0 pro neomezeně mnoho. |
| MAX_SESSIONS_ PER_ENDPOINT | Číslo | 10 | Maximální počet relací na Endpoint serveru, 0 pro neomezeně mnoho. |
| MAX_ SUBSCRIPTIONS | Číslo | 20 | Maximální počet odběrů na server, 0 pro neomezeně mnoho. |
| MAX_ SUBSCRIPTIONS_ PER_SESSION | Číslo | 2 | Maximální počet odběrů na jednu relaci, 0 pro neomezeně mnoho. |
| MAX_ SUBSCRIPTION_ LIFETIME | Číslo | 120000 | Maximální životnost odběru v ms. |
| MAX_ MONITORED_ ITEMS | Číslo | 200 | Maximální počet monitorovaných položek na server, 0 pro neomezeně mnoho. |
| MAX_ MONITORED_ ITEMS_PER_ SUBSCRIPTION | Číslo | 25 | Maximální počet monitorovaných položek na odběr, 0 pro neomezeně mnoho. |

Kapitola 4

Autentifikace a autorizace

V OPC UA serveru pro REX se využívají tři role s následujícími oprávněními: Host (Guest) může procházet Adresní prostor a číst data z parametrů bloků REXu. Operátor (Operator) má stejná práva jako host, má ale navíc povoleno zapisovat do parametrů bloků REXu a tím ovlivňovat běžící exekutivu. Administrátor (admin) má práva operátora, a navíc může spouštět metody, které ovlivňují server samotný, viz kapitola 2.1.

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 nebo při spuštění OPC UA metody “ReloadAuth”. Pro manipulaci se souborem je vytvořen samostatný program RexOpcUaAuth nebo lze využít grafické rozhraní RexSecurityConfig.

Oba programy využívají konfigurační soubor serveru, z něhož získají údaje o cestě k souboru s přihlašovacími údaji a přidavném kódování OPTIONAL_ENCODING_SALT. Pokud je hodnota přidavného kódování změněna, je nutné všem uživatelům znovu nastavit hesla nebo celý soubor vygenerovat znovu.

4.1 RexOpcUaAuth

Tento program se spouští z příkazové řádky a lze s jeho pomocí spravovat uživatele. Příkaz by měl být v jedné z následujících forem:

```
RexOpcUaAuth <configFile> -l  
RexOpcUaAuth <configFile> -c <username> <password> <role>  
RexOpcUaAuth <configFile> -p <username> <password>
```

```
RexOpcUaAuth <configFile> -a <username> <role>
RexOpcUaAuth <configFile> -r <username> <new_username>
RexOpcUaAuth <configFile> -d <username>
```

Program umožňuje zobrazit seznam uživatelů a jejich rolí (-l), vytvořit uživatele (-c), změnit mu heslo (-p), roli (-a), přejmenovat ho (-r) nebo ho smazat (-d). Parametr *configFile* je cesta ke konfiguračnímu INI souboru REXu, *username* je uživatelské jméno, *password* je heslo, *new_username* je nové uživatelské jméno při přejmenování. Parametr *role* musí mít hodnotu 'admin', 'operator' nebo 'guest'.

Cestu k INI souboru s přihlašovacími údaji (CREDENTIALS_INI_PATH) a nepovinný parametr OPTIONAL_ENCODING_SALT získá program z konfiguračního souboru.

Poznámka: Pokud soubor poškodíte nebo ztratíte, stačí vytvořit prázdný soubor a dál ho používat klasicky.

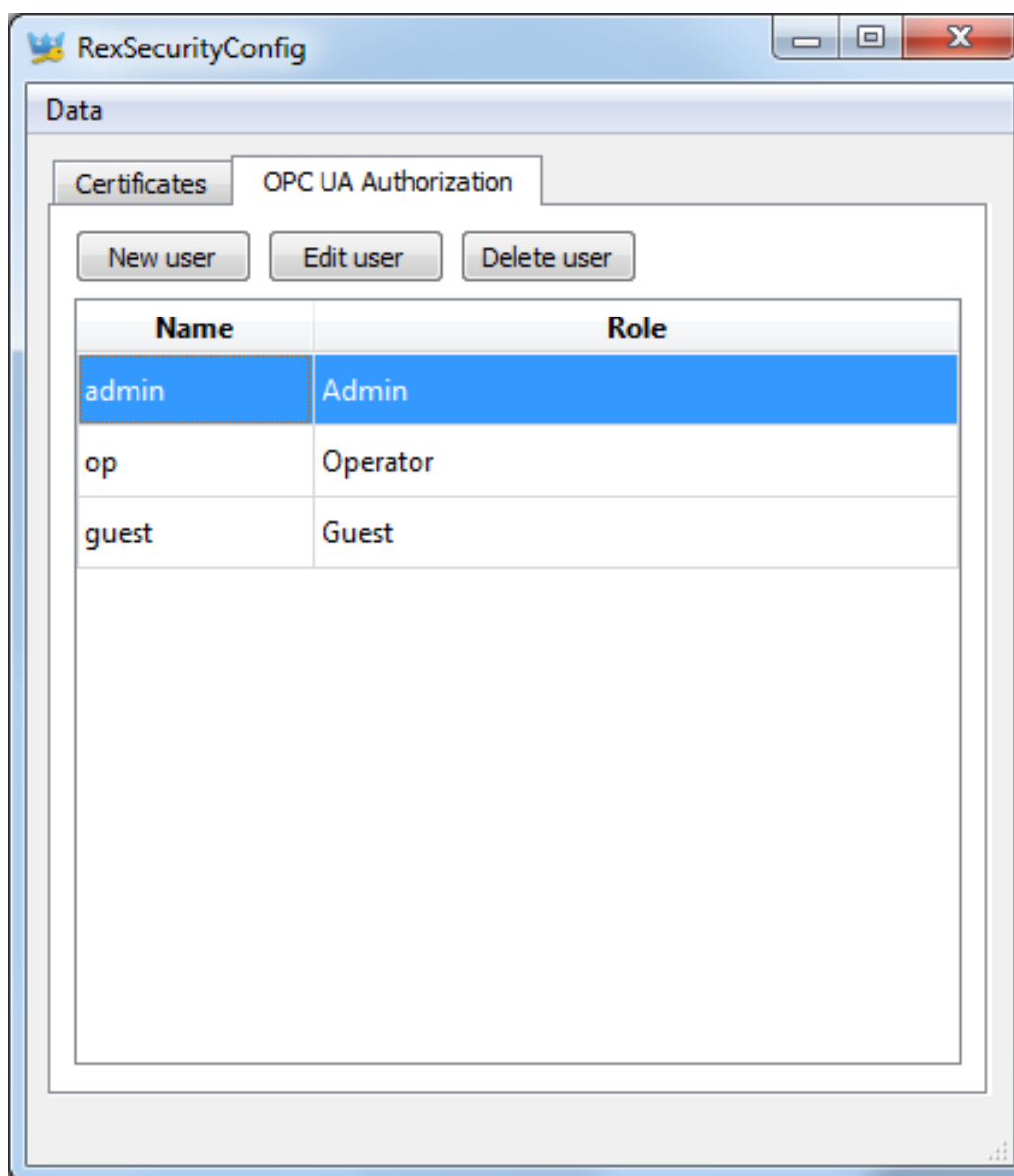
4.2 RexSecurityConfig

Program RexSecurityConfig umožňuje správu uživatelů pomocí grafického rozhraní (viz obrázek 4.1). Používá data z konfiguračního souboru, například cestu k souboru s přihlašovacími údaji a parametr OPTIONAL_ENCODING_SALT. Pokud jsou všechny parametry správně nastaveny, lze upravovat uživatele pomocí tabulky na záložce 'OPC UA Authorization'.

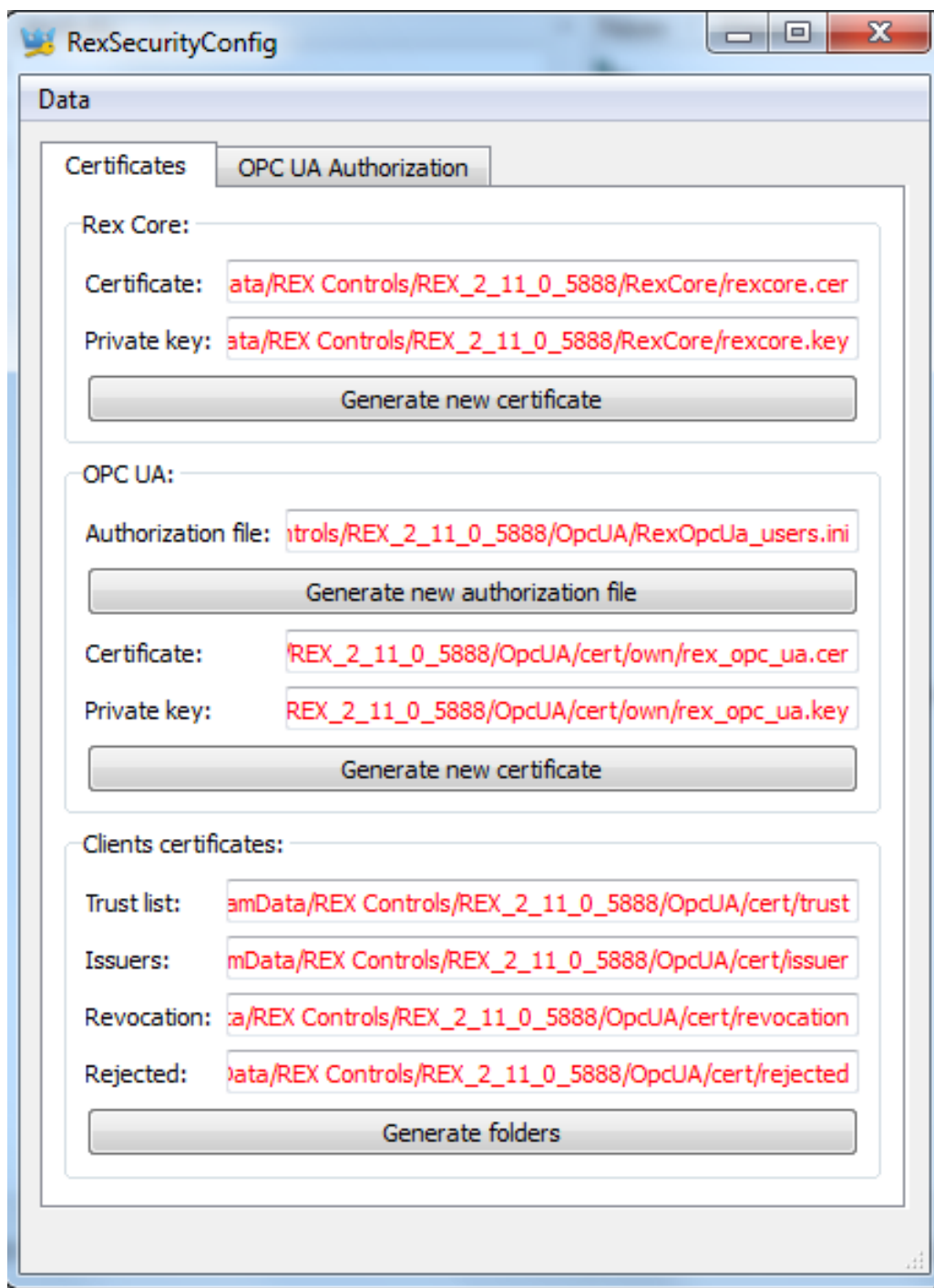
Kromě správy uživatelů umožňuje RexSecurityConfig pomocí jednoduchého dialogu vytvořit aplikační certifikát. Používá k tomu cestu k certifikátu a soukromému klíči. Pokud tyto dva soubory neexistují, zobrazí se na záložce 'Certifikáty' tlačítko Generovat (viz obrázek 4.2). Při generování je důležité vytvořit správnou Application URI, která souhlasí s parametrem APPLICATION_URI v konfiguraci a heslo k soukromému klíči, které musí souhlasit s parametrem APPLICATION_PRIVATE_KEY_PASSWORD. Aby byl certifikát použitelný i na jiných serverech, doporučujeme nespecifikovat (nevyplňovat) doménová jména a IP adresy.

Při práci s programem RexSecurityConfig je třeba dát pozor, aby program pracoval se správnou instancí REXu a správným konfiguračním souborem. To lze zjistit v Nastavení (viz obrázek 4.3).

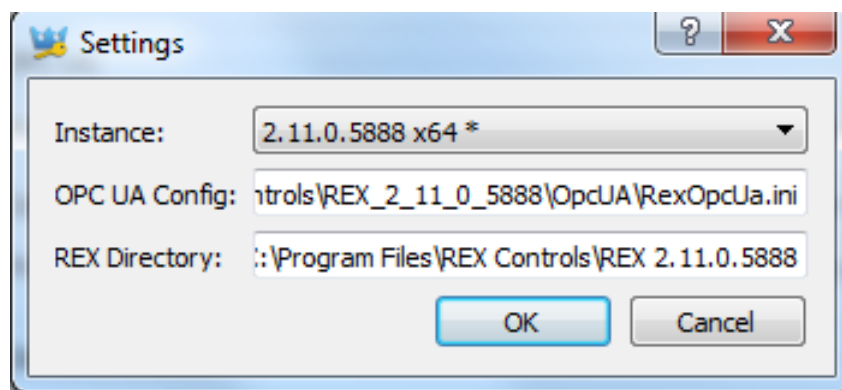
Poznámka: Další informace o programu RexSecurityConfig jsou obsaženy v dokumentaci tohoto programu.



Obrázek 4.1: Správa uživatelů v RexSecurityConfig



Obrázek 4.2: Záložka se základními informacemi



Obrázek 4.3: Nastavení programu RexSecurityConfig

Kapitola 5

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 REXu, ke kterému je server připojen. Ostatní záležitosti už musí vyřídit klient.

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

1. **Instalace REXu** spolu s OPC UA serverem (pokud není nainstalován)
2. Nalezení umístění přednastaveného konfiguračního souboru (pokud není vhodné při každém startu specifikovat cestu ke konfiguračnímu souboru, RexSecurityConfig - Nastavení)
3. **Změna konfiguračního souboru** - výběr z možností
 - (a) Zkopírování připraveného souboru ze složky příkladů pro REX
 - (b) Změna současného konfiguračního souboru
4. **Tvorba certifikátu** (pokud neexistuje) - výběr z možností
 - (a) Pomocí RexSecurityConfig (viz kapitola [4.2](#))
 - (b) Pomocí OpenSSL
5. Změna přihlašovacích údajů uživatelů (pokud je vyžadováno)
6. **Nastavení klientských certifikátů** (pokud Endpointy podporují zabezpečenou komunikaci nebo přihlášení pomocí certifikátů)
 - (a) Vytvoření složek pro klientské certifikáty (RexSecurityConfig)
 - (b) **Zkopírování** souborů **certifikátů klientů**, kteří se chtějí přihlašovat pomocí zabezpečeného kanálu, **do složky trust** (CERTIFICATE_TRUST_LIST_PATH)

7. 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 REX
 - 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)

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

- MINIMAL - Minimální konfigurace pro nezabezpečený Endpoint a REX na localhostu
- ENDPOINTS - Konfigurace se dvěma Endpointy
- LICENCE - Konfigurace s explicitně zadaným licenčním klíčem
- SECURITY - Konfigurace s Endpointem, který podporuje zabezpečenou komunikaci
- PASS_AUTH - Konfigurace s Endpointem, který podporuje přihlášení pomocí přihlašovacích údajů
- CERT_AUTH - Konfigurace s Endpointem, který podporuje přihlášení pomocí certifikátů
- DISCOVERY - Konfigurace s připojením informací o serveru na Discovery server
- FULL - Konfigurace se všemi možnými parametry

Pro úplně první úpravu konfiguračního souboru je doporučeno změnit, tak aby vyhovovali plánovanému použití serveru, parametry REX_ADDRESS, COMPANY_NAME, PROJECT_NAME a INSTANCE_NAME. Případně APPLICATION_PRIVATE_KEY_PASSWORD a OPTIONAL_ENCODING_SALT. Další parametry je možné měnit postupně a získávat tím informace o možnostech serveru.

5.1 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 UaExpert od firmy Unified Automation GmbH a software myScada.

U obou klientů bude navíc vysvětleno, jak má vypadat nastavení pro anonymní připojení (obrázek 5.1) a pro připojení pomocí přihlašovacích údajů (obrázek 5.2 a 5.3).

```
58 ;policies for anonymous access with default privileges
59 ADMIN_USER_TOKEN_POLICY_ID=0
60 OPERATOR_USER_TOKEN_POLICY_ID=1
61 GUEST_USER_TOKEN_POLICY_ID=2
62
63 [ENDPOINT:1]
64 ;N=none, L=low, M=medium, H=high, V=very high - could u
65 SECURITY_POLICY=[N;V]
66 ;policy id has to be identical to id of predefined user
67 USER_TOKEN_POLICY_ID=[Certificate;UsernamePassword;2]
68 URL=opc.tcp://localhost:4885/REX
69
70 [ENDPOINT:2]
71 ;N=none, L=low, M=medium, H=high, V=very high - could u
72 SECURITY_POLICY=[N;L;M;H;V]
73 USER_TOKEN_POLICY_ID=[0]
74 ;additional endpoint url is optional
75 URL=opc.tcp://localhost:4888/None/None
76
```

Obrázek 5.1: 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 5.2: Nastavení přihlašování pomocí přihlašovacích údajů

```
70 [ENDPOINT:2]
71 ;N=none, L=low, M=medium, H=high, V=very high
72 SECURITY_POLICY=[N;L;M;H;V]
73 USER_TOKEN_POLICY_ID=[UsernamePassword]
74 ;additional endpoint url is optional
75 URL=opc.tcp://localhost:4888/None/None
76
```

Obrázek 5.3: Nastavení uživatelské politiky na Endpoint

5.1.1 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 autentifikace, 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 5.4) 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.

Připojení k serveru lze provést pomocí tlačítka plus. Otevře se konfigurace připojení. V záložce ‘Advanced’ (obrázek 5.5) je možné nastavit Endpoint, zabezpečení a přihlašovací politiku (‘Session name’ nemá na chod vliv). Pokud je vybrána přihlašovací politika pomocí přihlašovacích údajů, je třeba zadat jméno a heslo uživatele (viz obrázek 5.6). Při přihlašování pomocí certifikátu je třeba zadat certifikát a soukromý klíč.

Funkční připojení je znázorněno zapojenou zástrčkou (viz obrázek 5.7). 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.

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 5.8) a kliknutím na ‘Add’. Monitorované položky je třeba najít ve stromu Adresního prostoru a přetáhnout do prostoru dokumentu (viz obrázek 5.9). 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 5.10).

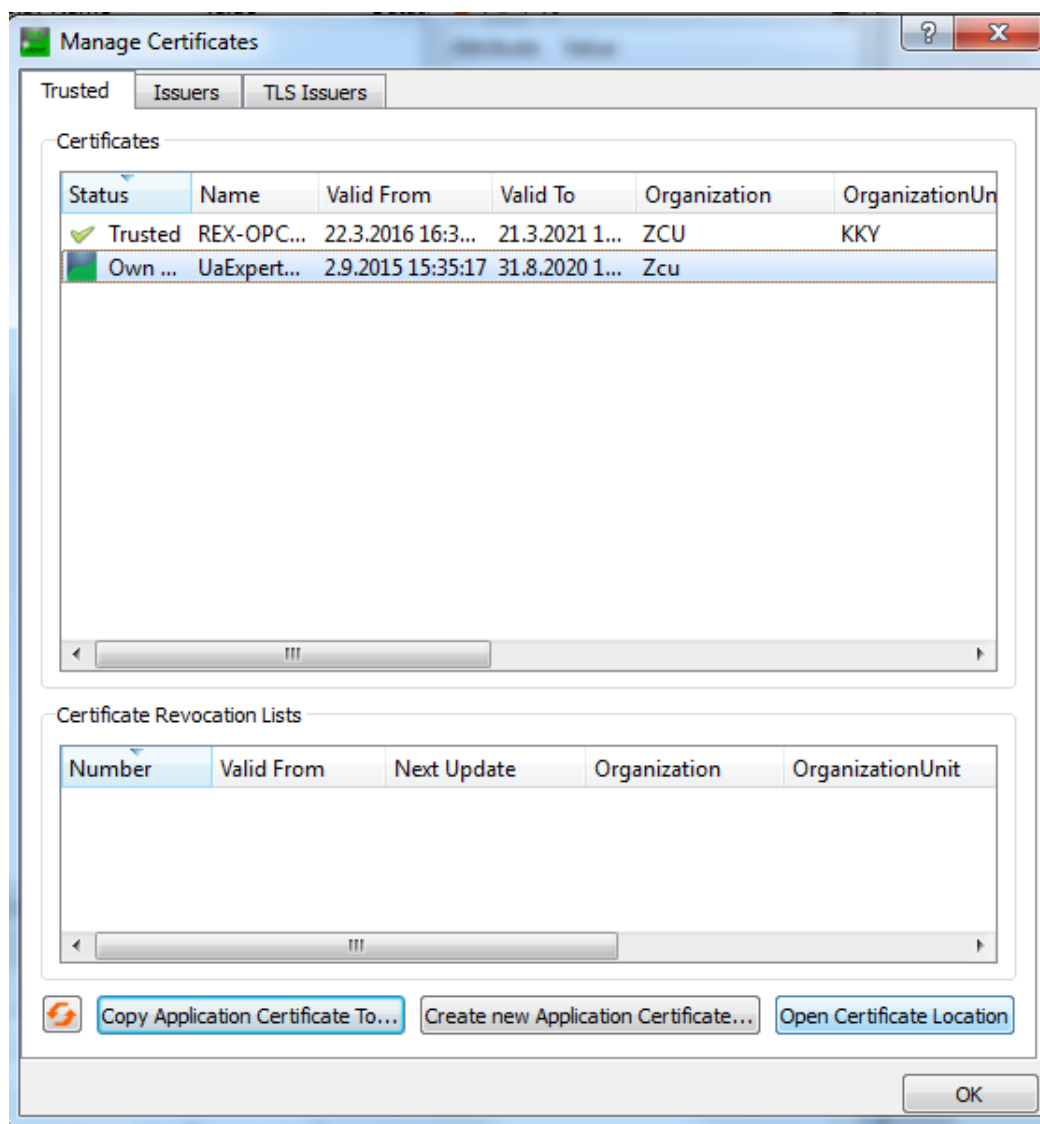
Pro monitorování je třeba přidat dokument ‘Event View’ (obrázek 5.11) a do něj přidat monitorované uzly přetáhnutím ze stromu Adresního prostoru do prostoru ‘Configuration’ (viz obrázek 5.12). Všechny události uzlu poté ukazují v poli ‘Events’ (obrázek 5.13). Při označení události se v poli ‘Details’ objeví detaily události. U OPC UA serveru pro REX je vhodné monitorovat složku ‘Exec’, popřípadě objekt ‘Server’, který je notifikován složkou ‘Exec’ (zobrazuje i její události).

UaExpert umí i jednoduché čtení, kdy se při kliknutí na uzel ve stromě Adresního

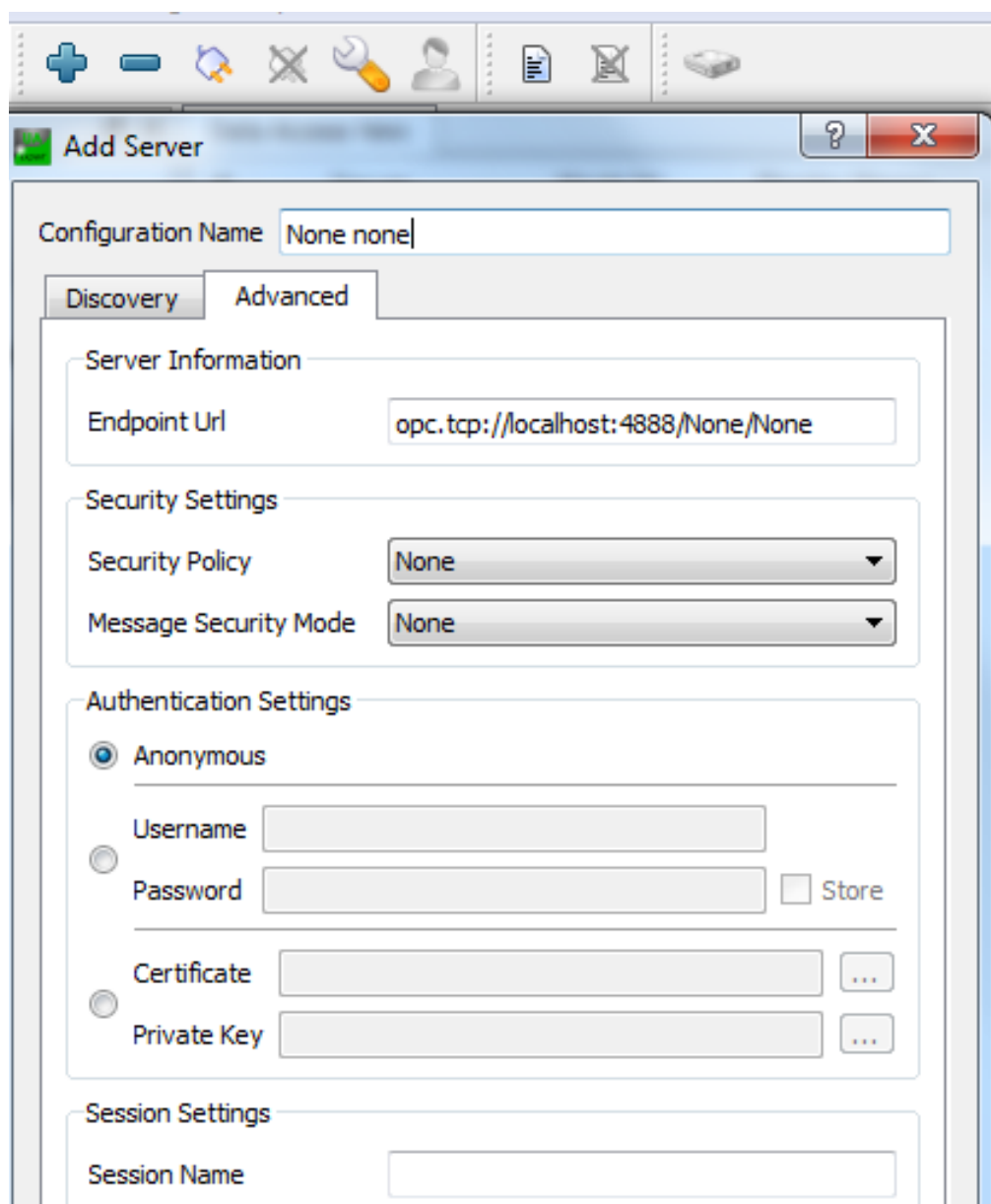
prostoru zobrazí v pravé části informace o uzlu. U hodnot proměnných se zobrazí i jejich hodnota (viz obrázek 5.14). Zápis do proměnné lze provést při dvojkliku na hodnotu value (viz obrázek 5.15).

UaExpert implementuje i Discovery služby, pomocí nichž zobrazuje všechny dostupné Endpointy registrovaných serverů (obrázek 5.16). Uživateli pak stačí pouze rozbalit seznam příslušného serveru, vybrat jednu z možností, nastavit přihlašovací politiku a připojit 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 REX, který podporuje Discovery služby a poskytuje informace o svých Endpointech.

Pokud nastávají problémy při připojení, zápisu, čtení nebo při čemkoliv jiném, je dobré zkontrolovat logy aplikace, které jsou zobrazeny v dolním panelu (viz obrázek 5.17). Podle nahlášené chyby lze často snadno dohledat zdroj problému.



Obrázek 5.4: UaExpert: Uložení certifikátu do důvěryhodných



Obrázek 5.5: UaExpert: Připojení k serveru anonymně

Add Server

Configuration Name: None none

Discovery **Advanced**

Server Information

Endpoint Url: opc.tcp://localhost:4888/None/None

Security Settings

Security Policy: None

Message Security Mode: None

Authentication Settings

☐ Anonymous

☒ Username: admin

Password: ••••• ☒ Store

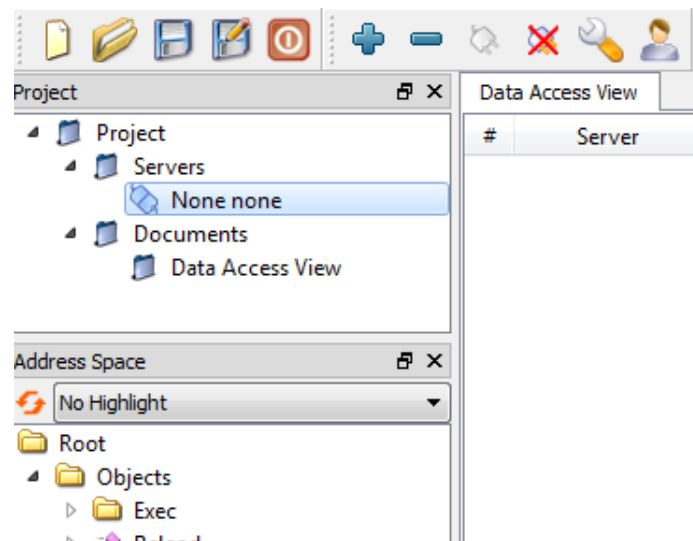
☐ Certificate: ...

☐ Private Key: ...

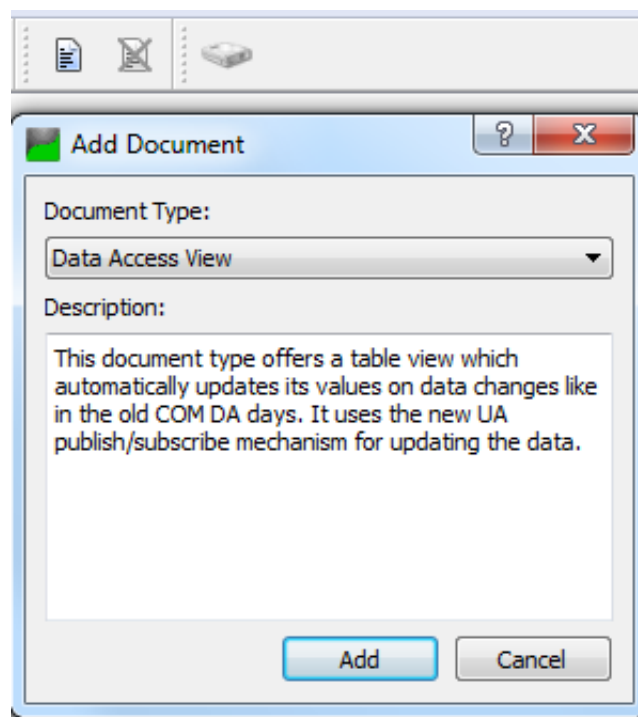
Session Settings

Session Name:

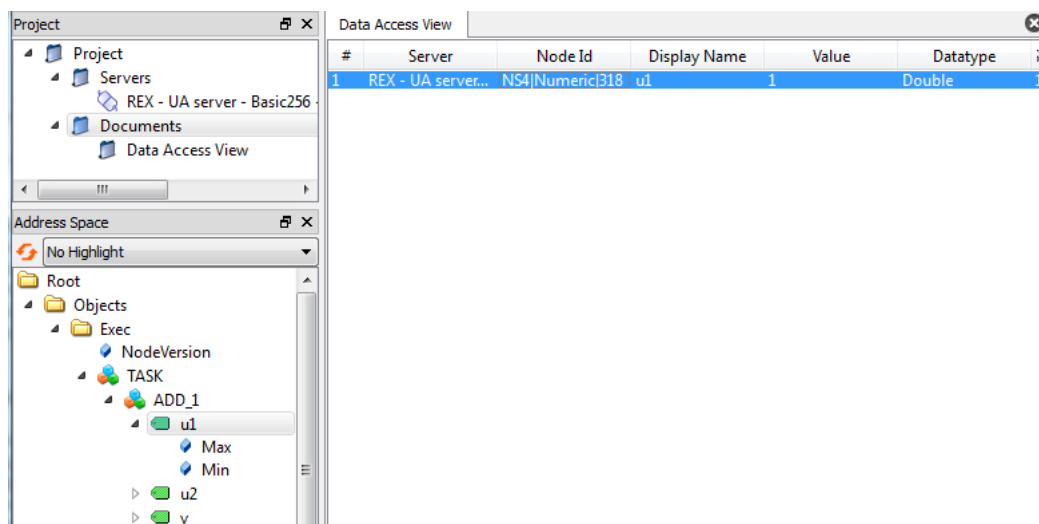
Obrázek 5.6: UaExpert: Připojení k serveru s přihlašovacími údaji



Obrázek 5.7: UaExpert: Připojený k serveru



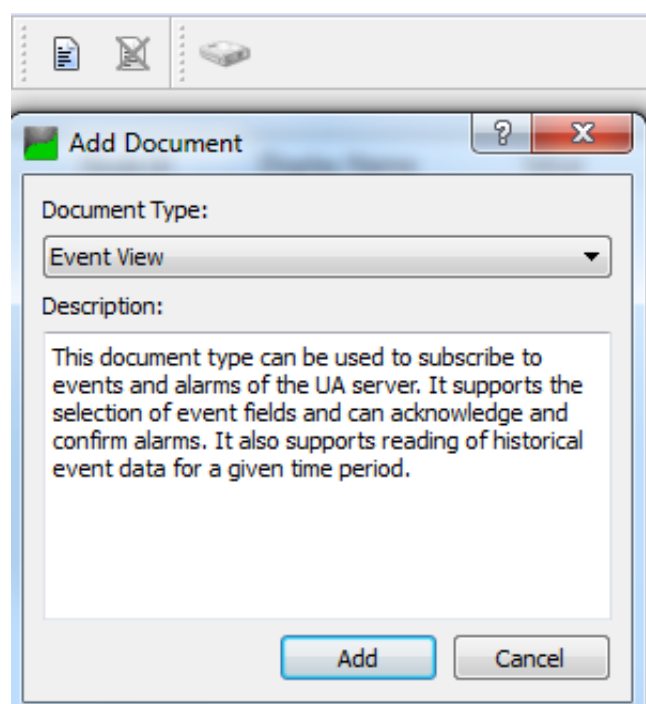
Obrázek 5.8: UaExpert: Přidání monitorování dat



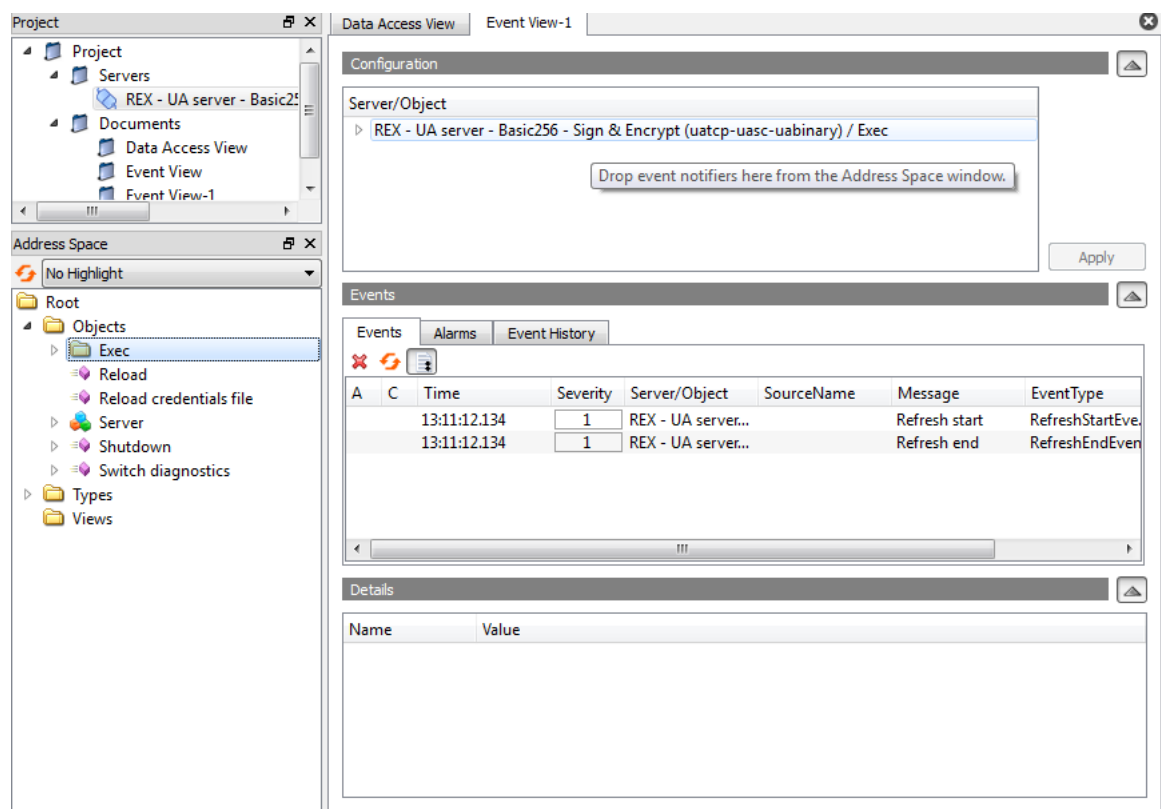
Obrázek 5.9: UaExpert: Monitorování proměnné u1

| Data Access View | | | | | | |
|------------------|--------------------|-----------------|--------------|-------|----------|---|
| # | Server | Node Id | Display Name | Value | Datatype | |
| 1 | REX - UA server... | NS4 Numeric 318 | u1 | 0 | Double | 1 |

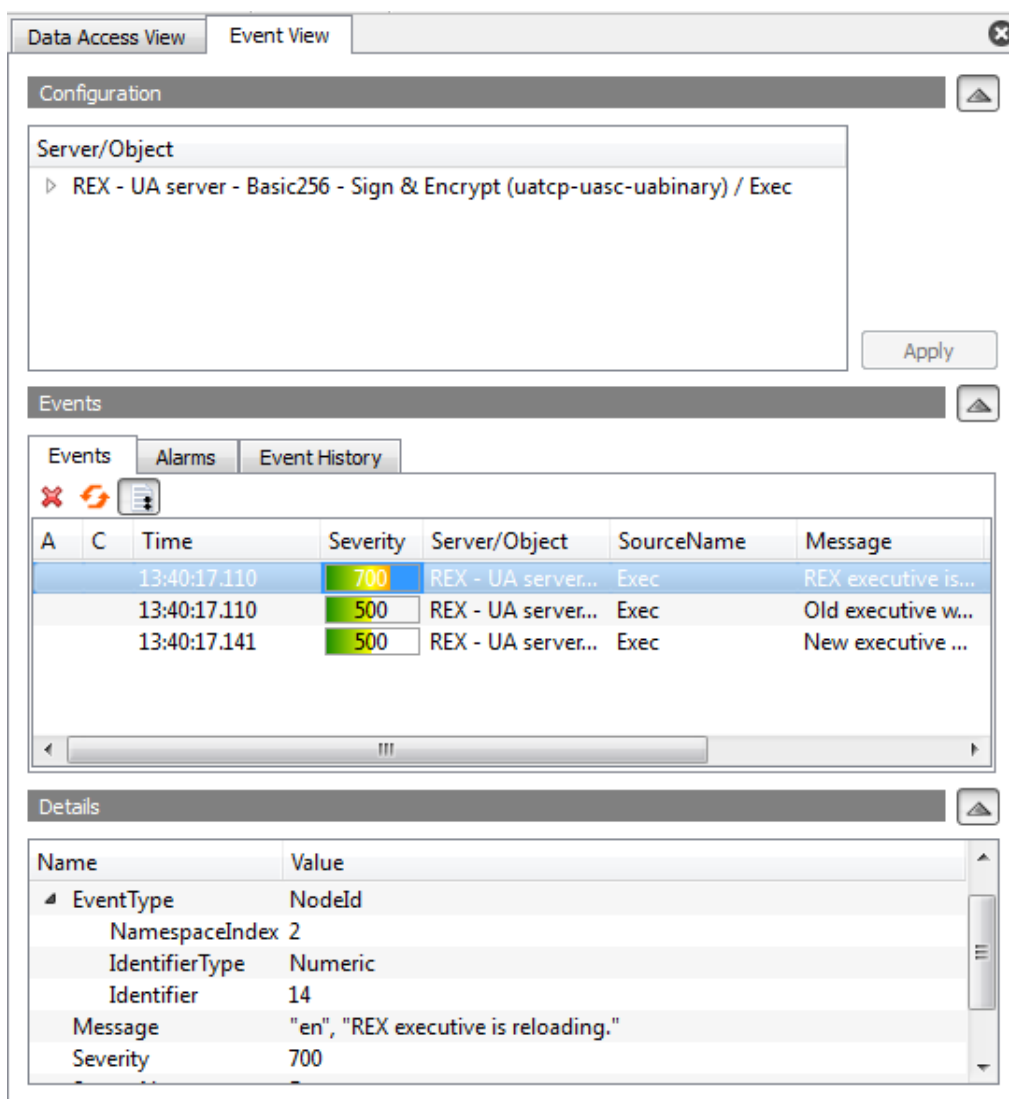
Obrázek 5.10: UaExpert: Zápis do proměnné u1



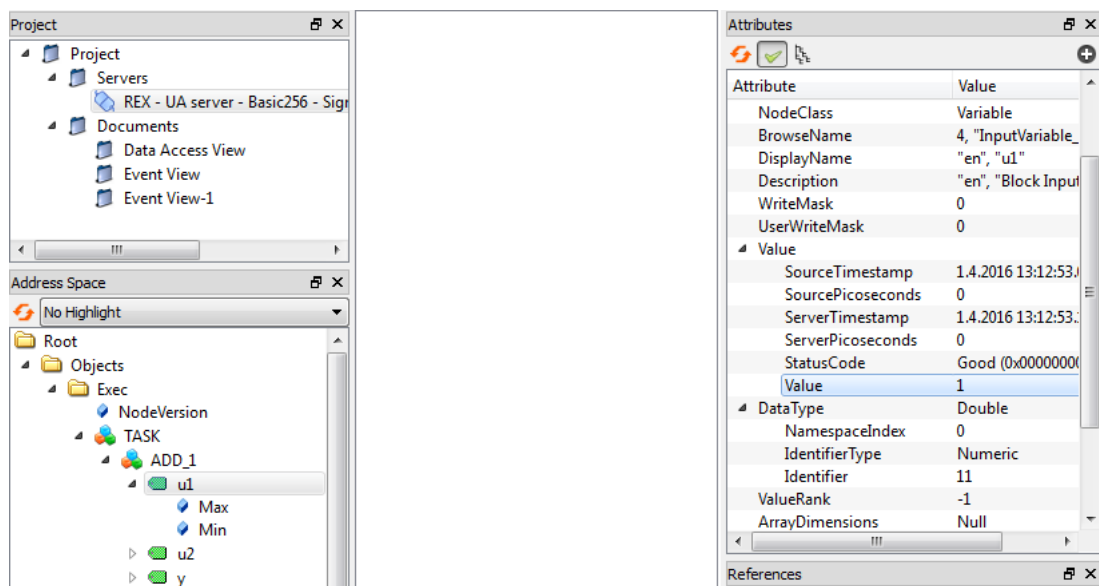
Obrázek 5.11: UaExpert: Přidání monitorování událostí



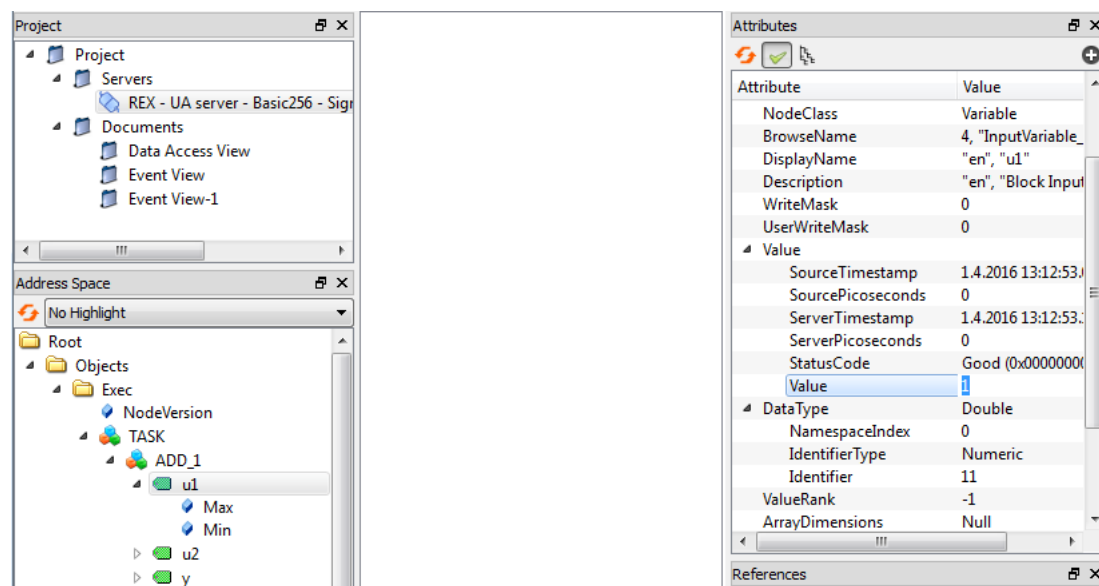
Obrázek 5.12: UaExpert: Monitorování událostí složky Exec



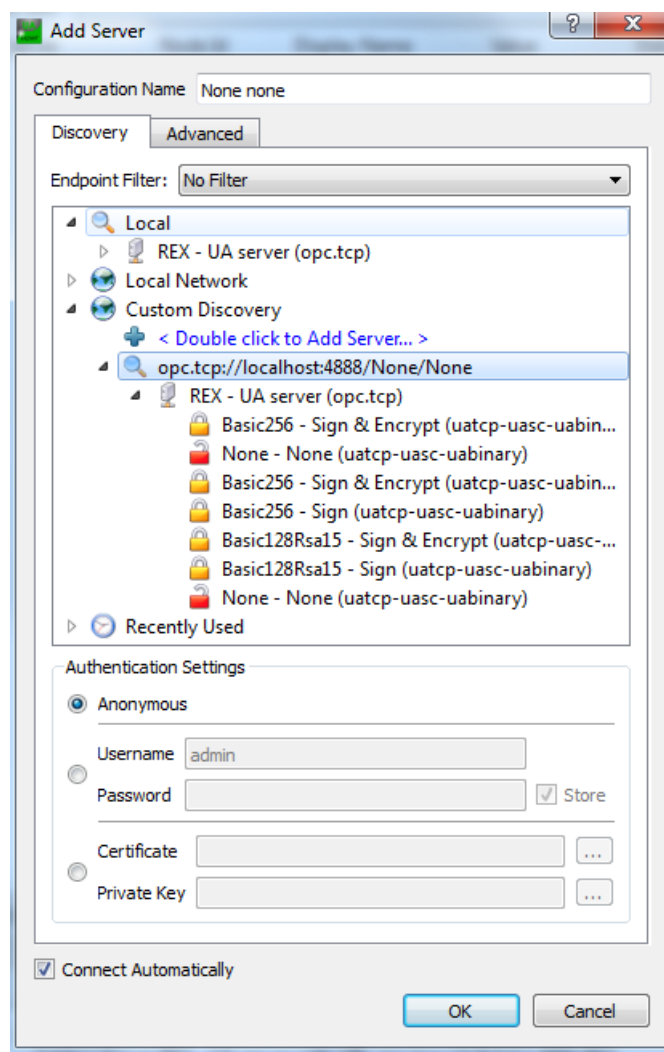
Obrázek 5.13: UaExpert: Zobrazení událostí





Obrázek 5.14: UaExpert: Čtení proměnné u1



Obrázek 5.15: UaExpert: Zápis do proměnné u1



Obrázek 5.16: UaExpert: Použití Discovery služeb

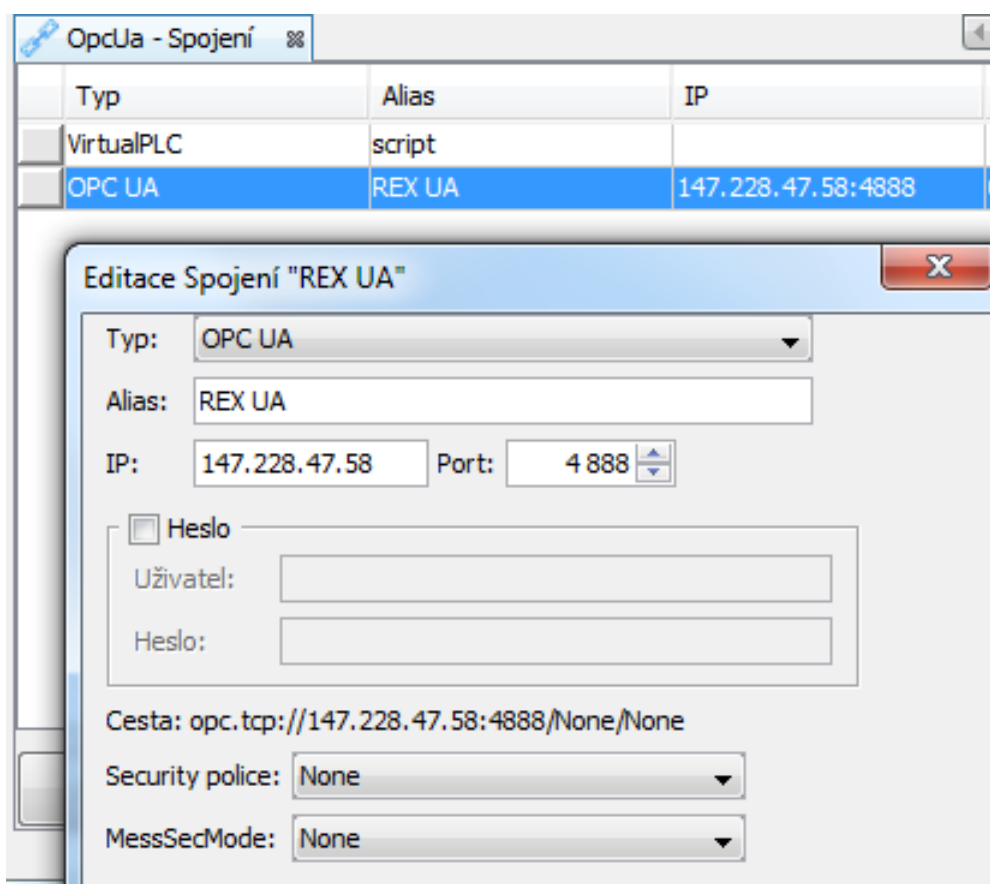
| Log | | | |
|---|-----------|--------------------|---|
|   | | | |
| Timestamp | Source | Server | Message |
| 1.4.2016 13:20:49.... | DA Plugin | REX - UA server... | Write to node 'NS4 Numeric 318' succeeded [ret = Good] |
| 1.4.2016 13:18:56.... | DA Plugin | REX - UA server... | Item [NS4 Numeric 318] succeeded : RevisedSamplingInter |
| 1.4.2016 13:18:56.... | DA Plugin | REX - UA server... | CreateMonitoredItems succeeded [ret = Good] |
| 1.4.2016 13:18:56.... | DA Plugin | REX - UA server... | Item [NS4 Numeric 318]: SamplingInterval=250, QueueSize |
| 1.4.2016 13:18:56.... | DA Plugin | REX - UA server... | Created subscription for ServerId 0 |

Obrázek 5.17: UaExpert: Logování akcí

5.1.2 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.

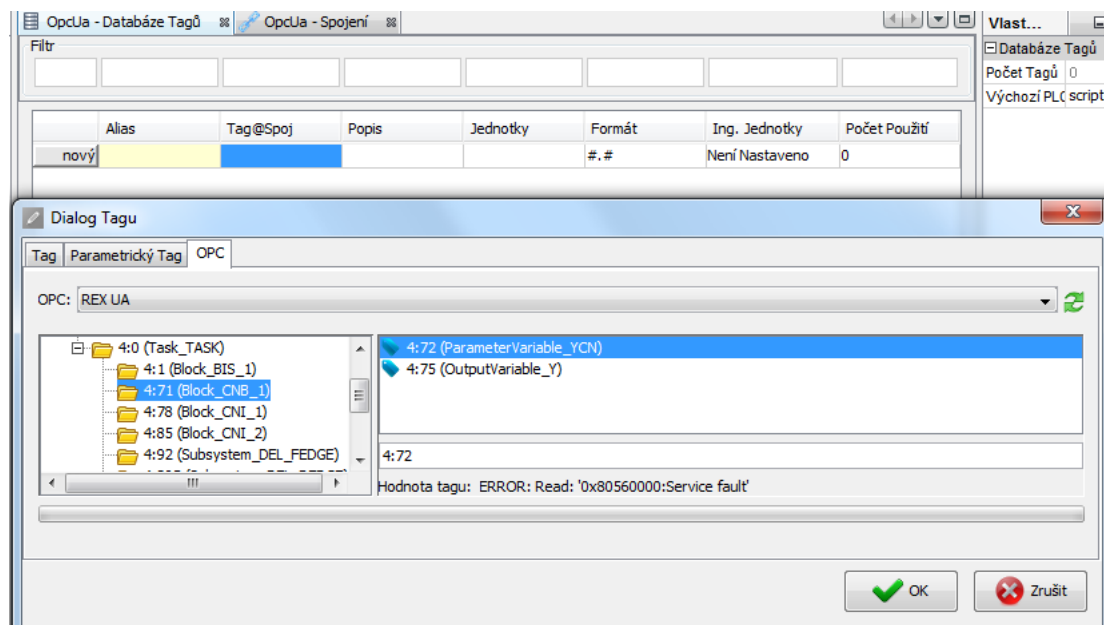
Pro použití OPC UA v myScada je třeba v programu myPROJECT designer vytvořit projekt, otevřít záložku spojení, přidat nové spojení a zvolit OPC UA. Otevře se dialog, v němž je možné nastavit spojení s OPC UA serverem (obrázek 5.18).



Obrázek 5.18: 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 5.19). Hodnotu tohoto tagu lze na závěr využít v zobrazení aplikace (obrázek 5.20). Hotový projekt lze poté nahrát na zařízení a sledovat jeho chod pomocí programu myView.

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 5.21). Pokud se přeruší spojení nebo serveru dojdou zdroje, označí



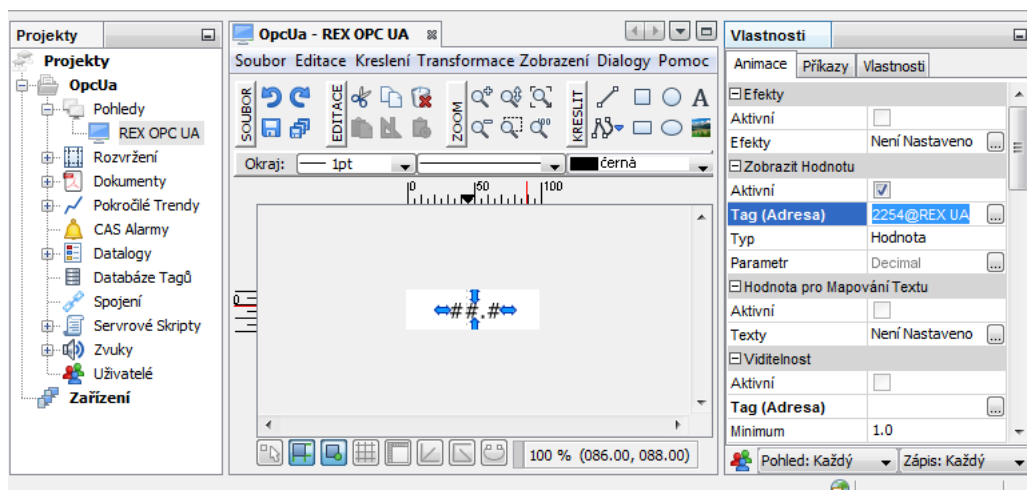
Obrázek 5.19: mySCADA: Vytvoření tagu z uzlu OPC UA serveru

zařízení hodnotu v zobrazení červeně (viz obrázek 5.22).

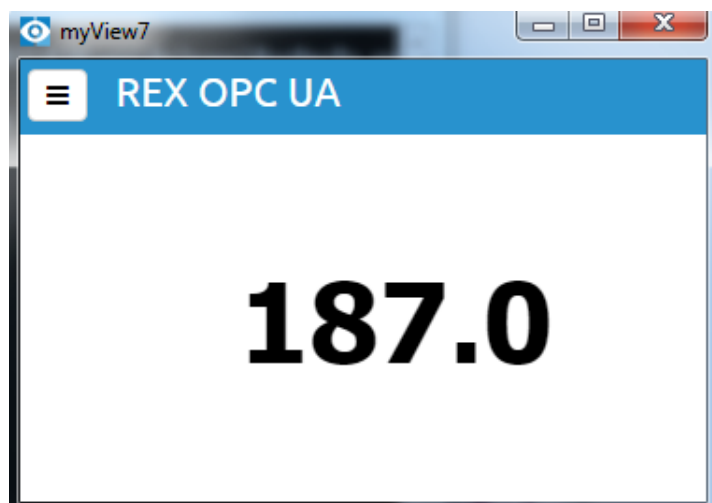
Jelikož se myScada neřídí přesně specifikací OPC UA, je nutné upravit server tak, aby se k němu mohl klient připojit. Pro nezabezpečené přihlášení musí mít server Endpoint s koncovou URI '/None/None'. Pro anonymní přihlášení musí být ID přihlašovací politiky '0' (viz [Endpoint:2] a ADMIN_USER_TOKEN_POLICY_ID na obrázku 5.1).

Pro nastavení připojení pomocí přihlašovacích údajů je třeba nastavit ID patřičné politiky na 'UserNameIdentityToken' (obrázek 5.23 a 5.24) a zadat uživatelské jméno a heslo v dialogu pro úpravu spojení (viz obrázek 5.25).

Při práci s myScada je doporučeno **nemířit na OPC UA server pomocí localhostu, 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 5.26).



Obrázek 5.20: mySCADA: Použití tagu v zobrazení



Obrázek 5.21: mySCADA: Zobrazení dat v zařízení



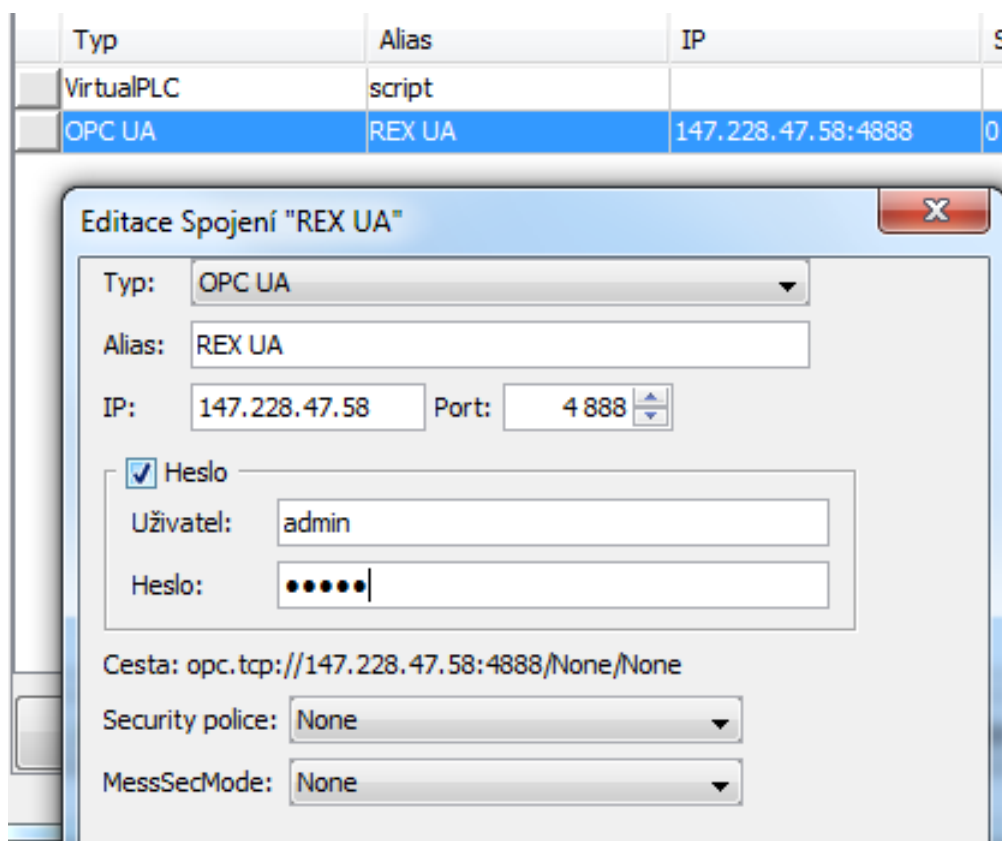
Obrázek 5.22: mySCADA: Zařízení bez dat

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

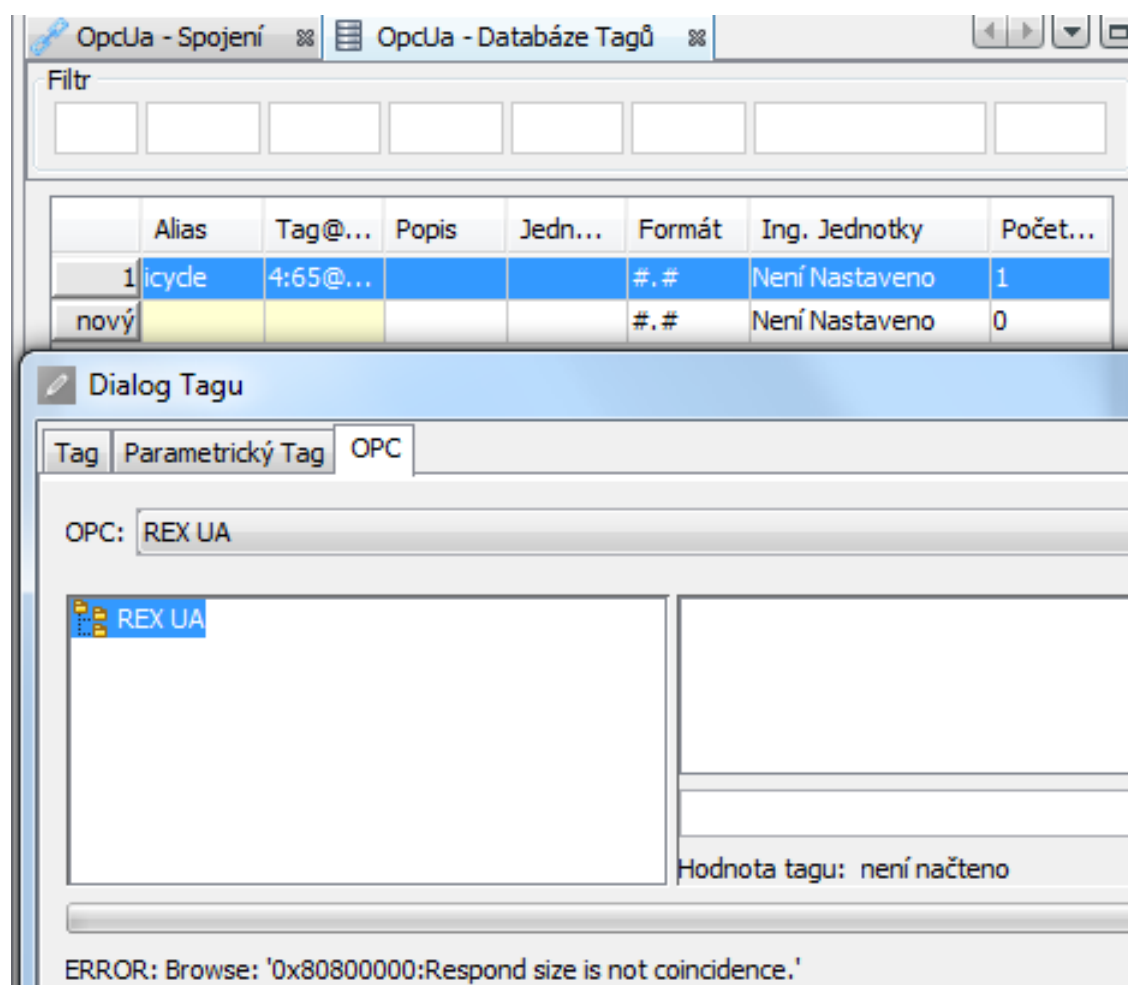
Obrázek 5.23: mySCADA: Nastavení přihlašování pomocí přihlašovacích údajů

```
[ENDPOINT:2]
;N=none, L=low, M=medium, H=high, V=very high
SECURITY_POLICY=[N;L;M;H;V]
USER_TOKEN_POLICY_ID=[UserNameIdentityToken]
;additional endpoint url is optional
URL=opc.tcp://localhost:4888/None/None
```

Obrázek 5.24: mySCADA: Nastavení uživatelské politiky na Endpoint



Obrázek 5.25: mySCADA: Přihlášení pomocí přihlašovacích údajů



Obrázek 5.26: mySCADA: Chyba připojení

Literatura