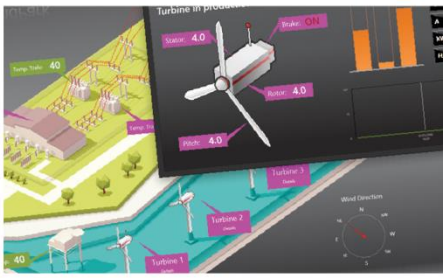
INDUSTRIAL AUTOMATION SOFTWARE
progea

Movicon.next™
Automation Platform.next generation



Movicon NExT

4.2 OPC UA

Ver.3.4.268

Sommario

OPC UA - IEC62541 1

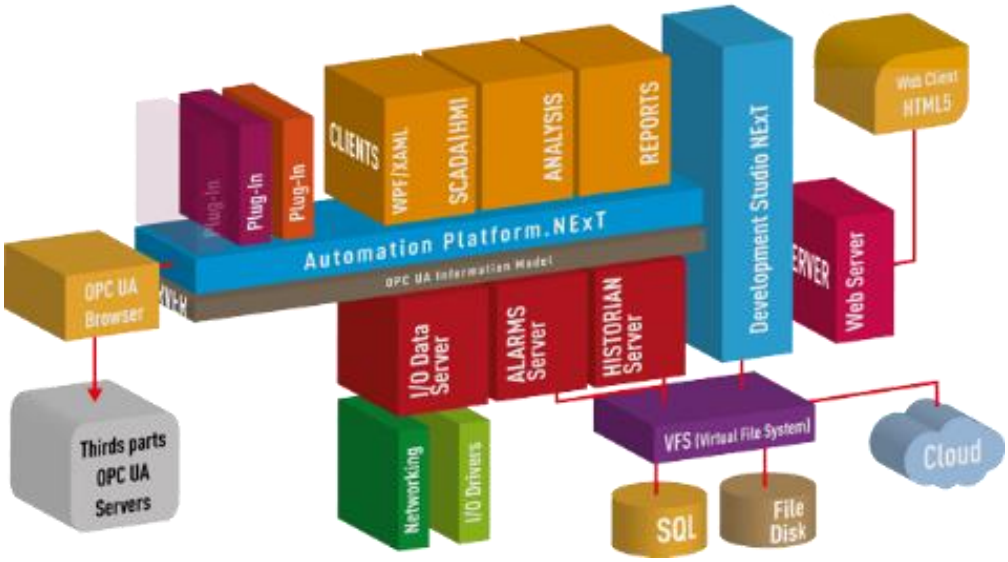
OPC UA Information Model in Platform.NExT 1
OPC UA (Unified Architecture) IEC 62541 1
Indipendenza dal Trasporto Dati 3
OPC UA Sicurezza 4
OPC UA Server 8
OPC UA Server 8
OPC UA Client 12
OPC UA Client 12
Connessione lato Client, tramite OPC UA Browser 14
Connessione lato Server, tramite Driver OPC UA Client 17

1. OPC UA - IEC62541

1.1. OPC UA Information Model in Platform.NExT

Il framework della piattaforma Automation Platform.NExT basa il proprio modello di informazione dati sulla tecnologia OPC UA, sia differenziando l'infrastruttura in Client e Server, sia sviluppandone appieno le parti funzionali:

- l'I/O Data Server svolge la funziona di Data Access (DA),
- l'Alarm Server assolve il ruolo di Alarm and Condition (AC)
- l'Historian Server gestisce la parte di Historical Access (HA)



Questa tecnologia esclusiva di Progea, totalmente trasparente per l'utente che usa il framework, consente di disporre di un grande vantaggio: quello di utilizzare una piattaforma aperta, espandibile, modulare, conforme agli standards e che consente la piena e totale integrazione con i modelli di dati, semplici o complessi, provenienti da altri livelli d'automazione, come ad esempio i sistemi di controllo IEC 61131-3 PLCopen o i moderni sistemi MES/ERP.

1.2. OPC UA (Unified Architecture) IEC 62541

L'OPC Unified Architecture, rilasciata nel 2008, è una piattaforma indipendente basata su un'architettura service-oriented che integra tutte le funzionalità previste dalla specifica OPC Classic in un'unico framework estensibile. Questo approccio su più livelli permette di realizzare gli obiettivi delle specifiche originali quali:

- Functional equivalence: tutte le specifiche COM OPC Classic sono mappate su UA.

- Platform independence: da un microcontroller incorporato ad un'infrastruttura basata su cloud.
- Secure: crittografia, autenticazione e auditing.
- Extensible: possibilità di aggiungere nuove funzionalità senza influire sulle applicazioni esistenti
- Comprehensive information modeling: per la definizione di informazioni complesse.

Functional Equivalence

Basandosi sul successo di OPC Classic, la tecnologia OPC UA è stata progettata per migliorare e superare le capacità delle specifiche OPC Classic. OPC UA è funzionalmente equivalente
a OPC Classic, ma capace di molto di più:

- Discovery: trova la disponibilità di OPC Server su PC e / o reti locali.
- Address Space: tutti i dati sono rappresentati gerarchicamente (ad esempio file e cartelle) permettendo delle strutture semplici e complesse da utilizzare con OPC Client.
- On-demand: leggere e scrivere dati / informazioni in base alle autorizzazioni di accesso.
- Subscriptions: monitora dati / informazioni e segnala con un'eccezione quando i valori cambiano in base ai criteri di un cliente.
- Events: notifica informazioni importanti in base ai criteri del cliente.
- Metods: i client possono eseguire programmi, ecc. In base ai metodi definiti sul server.

Piattaforma indipendente

Data la vasta gamma di piattaforme hardware e sistemi operativi disponibili, la platform independence risulta essenziale. La tecnologia OPC UA funziona quindi sulle seguenti e su tanto altro:

- Piattaforme hardware: hardware PC tradizionale, server basati su cloud, PLC, microcontrollori (ARM ecc.)
- Sistemi operativi: Microsoft Windows, Apple OSX, Android o qualsiasi distribuzione di Linux, ecc.

OPC UA fornisce l'infrastruttura necessaria per l'interoperabilità in tutta l'azienda, da macchina a macchina, da macchina a impresa e per tutto ciò che sta nel mezzo.

Sicurezza

Una delle considerazioni più importanti nella scelta di una tecnologia è la sicurezza. OPC UA è svincolata dalle limitazioni legate ai firewall per quanto riguarda le regole di sicurezza, fonrnendo una serie di controlli così caratterizzati:

- Transport: vengono definiti numerosi protocolli che offrono opzioni come l'ultraveloce OPC-binary transport o il più universalmente compatibile SOAP-HTTPS, per esempio
- Session Encryption: i messaggi vengono trasmessi in modo sicuro con crittografia a 128 o 256 bit livelli.
- Message Signing: i messaggi vengono ricevuti esattamente come sono stati inviati.
- Sequenced Packets: l'esposizione agli attacchi di ripetizione dei messaggi viene eliminata con il sequenziamento
- Autentication: ogni client e server UA vengono identificati tramite i cert+ificati OpenSSL fornendo il controllo su cui le applicazioni e i sistemi saranno autorizzati a connettersi l'un l'altro.

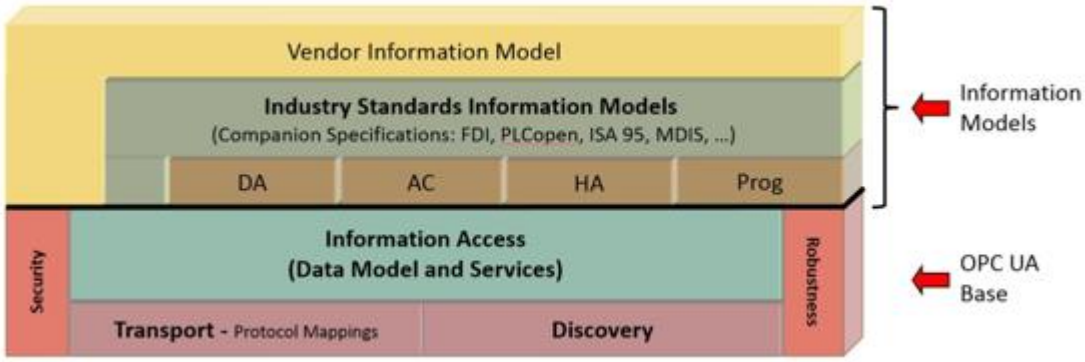
- User Control: le applicazioni possono richiedere agli utenti di autenticarsi (credenziali di accesso, certificato, ecc.) e può ulteriormente limitare e migliorare le proprie capacità con l'accesso diritti e spazio degli indirizzi "viste"
- Auditing: le attività dell'utente e / o del sistema vengono registrate fornendo una traccia di controllo degli accessi.

Estensibile

L'architettura multistrato di OPC UA offre una struttura "a prova di futuro". Innovative tecnologie e metodologie come nuovi protocolli di trasporto, algoritmi di sicurezza, standard di codifica, o servizi applicativi possono essere incorporati in OPC UA mantenendo comunque la retrocompatibilità per i prodotti esistenti. Questo significa che i prodotti UA costruiti oggi lavoreranno anche con i prodotti di domani.

Modellazione Informazioni

La struttura di modellazione delle informazioni di OPC UA trasforma i dati in informazioni. ..... I tipi di dati e le strutture verranno definite poi nei profili.
In questo esempio, le specifiche OPC Classic esistenti sono state modellate in profili UA che possono anche essere estesi da altre organizzazioni:



OPC UA Base Services Architecture

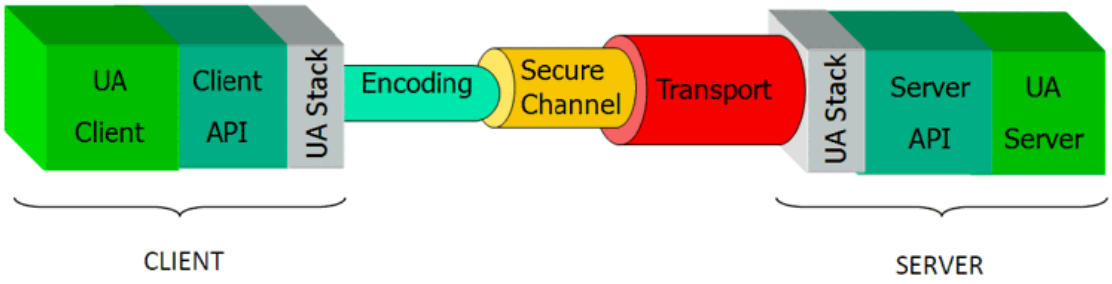
More info:
https://opcfoundation.org/about/opc-technologies/opc-ua/

1.3. Indipendenza dal Trasporto Dati

Una delle componenti fondamentali nel modello dei dati OPC UA è il trasporto. Il Trasporto corrisponde al mezzo, all'infrastruttura che i dati utilizzeranno per essere connessi tra Client e Server. In Platform.NExT, i dati sono indipendenti dal trasporto che si desidera utilizzare, o che occorre utilizzare per le proprie connessioni verso sistemi di terze parti su OPC UA.
Infatti, i Trasporti possono essere di tipo diverso, a secondo se si desidera privilegiare le prestazioni o la sicurezza.

I trasporti proposti dal Server Dati sono descritti nell'apposito capitolo dell'I/O Data Server di Platform.NExT.

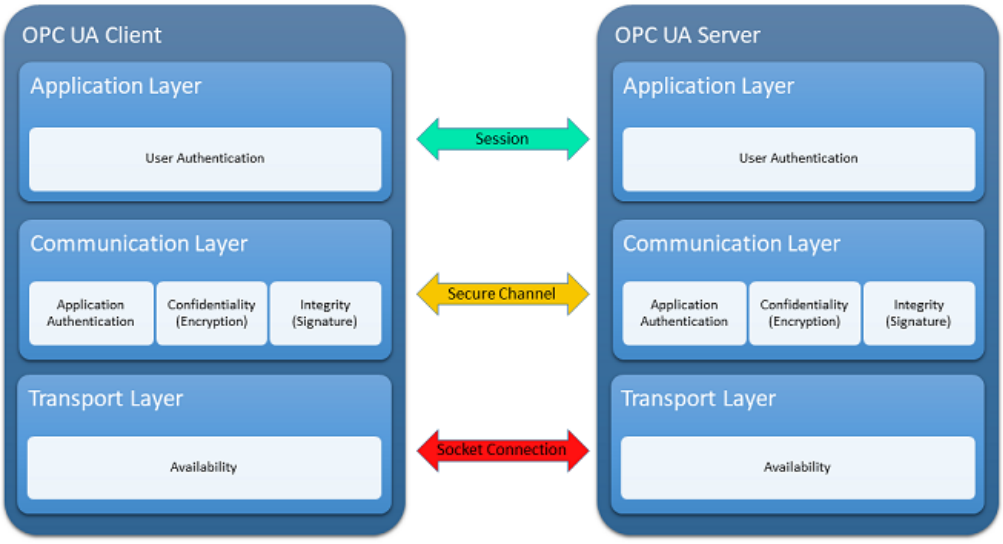
Per mettere in comunicazione un Server ed un Client OPC UA, è sempre necessario accertarsi che il trasporto utilizzato sia idoneo e coerente.



In OPC UA, pur essendo utilizzata un'architettura client-server, è tipico che un'applicazione rivesta entrambi i ruoli, ciò perché spesso nei dispositivi fisici è integrato anche il lato server (comunicazione device to device). Una tipica applicazione OPC UA è composta da tre strati software descritti nella figura indicata sopra.

1.4. OPC UA Sicurezza

L'architettura di sicurezza OPC UA è una soluzione che consente l'implementazione delle funzioni di sicurezza in varie parti ed è strutturato in diversi livelli: Application Layer, Communication Layer e Transport Layer.



L'Application Layer viene usato per la trasmissione di informazioni tra Client e Server che hanno stabilito una OPC-UA Session su un Secure Channel (che si trova al Communication Layer). Il Secure Channel rende sicuro lo scambio dei dati di una sessione. Il Transport Layer invece è il livello responsabile della trasmissione e ricezione dei dati.

La sicurezza in OPC UA dunque è realizzata tramite la definizione di meccanismi di sicurezza che consentono l'Autenticazione dell'Applicazione nel Secure Channel e l'Autenticazione dell'Utente nella Sessione.

Secure Channel definition

Il Secure Channel, dove avviene l'Autenticazione dell'Applicazione, rende sicuro lo scambio dei dati in due differenti modi:

- **Sign**: assicurando l'integrità dei dati mediante firma digitale. Il meccanismo di sicurezza previsto per la firma digitale prevede che:
    - Ciascuna parte (Client e Server) possiede una coppia di chiavi (Public/Private Key)
    - Ciascuna parte (Client e Server) rende nota la propria public key
    - Ciascun messaggio scambiato tra le parti viene "firmato" utilizzato la chiave privata del mittente, e "verificato" dal destinatario con la chiave pubblica del mittente.

- **Encrypt**: assicurando la confidenzialità dei dati tramite cifratura ovvero adottando la crittofrafia asimmetrica. Il meccanismo di sicurezza previsto per la crittografia asimmetrica prevede che:
    - Ciascuna parte (Client e Server) possiede una coppia di chiavi (Public/Private Key)
    - Ciascuna parte (Client e Server) rende nota la propria public key
    - Cifratura: Ciascun messaggio scambiato tra le parti viene "cifrato" utilizzando la chiave pubblica della controparte, e "decifrato" con la propria chiave privata.

La definizione di un Secure Channel avviene poi applicando un Security Profile ed una Security Policy resi disponibili dal Server. I Security Profile (denominati anche Security Mode) sono di fatto tre profili di sicurezza predefiniti quali:

- None: nessuna sicurezza, canale non sicuro.
- Sign: sicurezza sull'integrità dei dati tramite firma digitale (Application Certificate X.509). Ai dati scambiati viene applicata la firma digitale del mittente. Il destinatario può verificare che i dati provengano effettivamente dal mittente che si aspetta.
- Sign & Encrypt: sicurezza sia sull'integrità che sulla confidenzialità dei dati tramite firma e cifratura. Ai dati scambiati viene prima applicata la firma digitale del mittente, dopodichè vengono criptati (Application Instance Certificate X.509).

Le Security Policy invece indicano la lunghezza della chiave e l'algoritmo usato nel Security Profile; ad esempio: Basic128Rsa15, Basic256, Basic256Sha256, Aes256-Sha256-RsaPss, etc...
La combinazione di Security Policy e Security Profile forma il Security Level che indica il grado di sicurezza del Livello di Comunicazione, dove il Level 0 indica il livello più basso. Un Client quindi potrà scegliere la Security Policy e il Security Profile semplicemente confrontando i Security Level.

Session definition.

A Livello Applicativo invece i meccanismi di sicurezza che consentono l'autenticazione dell'utente (e di conseguenza la sua autorizzazione) garantiscono l'accesso per un utente specifico (e quindi il suo ruolo) durante la configurazione della Sessione. Sono

disponibili a questo livello tre diversi modalità di autenticazione: anonima, tramite credenziali (username/password), tramite firma digitale (User Certificate X.509)

Secure Channel and Session establishment

La creazione di un SecureChannel si basa prima di tutto sulla scelta del Session Endpoint. Ogni OPC UA Server può offrire uno o più Session Endpoints caratterizzati da:

- Endpoint Url: indirizzo di rete dell'endpoint utilizzato dal client per stabilire un SecureChannel
- Application Instance Certificate del Server: contiene la chiave pubblica del Server
- Security Policy: lunghezza della chiave e l'algoritmo usato nel Security Mode
- Security Mode: none, Sign o Sign&Encrypt
- User Authentication: anonymous, username/password, User Certificate X.509
- Transport Protocol: specifica delle caratteristiche stack usato dall'EndPoint: encoding, security, transport.

Per stabilire una Session su una connessione sicura tra un OPC-UA Client e un OPC-UA Server si devono seguire i seguenti passi.

1. Il primo passo consiste nella scelta di un Session Endpoint. Il client seleziona un Session Endpoint e procede a validare l'Application Instance Certificate del Server. Se il certificato è attendibile si prosegue.

2. Il Client invia una richiesta di Open Secure Channel in accordo alla Security Mode del Session Endpoint selezionato.
    - Se il Security Mode è "None" allora la richiesta è inviata senza meccanismi di sicurezza
    - Se il Security Mode è "Sign" allora la richiesta è inviata usando come firma la Private Key associata all'Application Instance Certificate del Client
    - Se il Security Mode è "Sign&Encrypt" allora la richiesta è inviata usando come firma la Private Key associata all'Application Instance Certificate del Client ed applicando in aggiunta la codifica usando la Public Key dell'Application Instance Certificate del Server (oltre alla firma).

3. Il Server riceve il messaggio di Open Secure Channel e valida l'Application Instance Certificate del client contenuto nel messaggio. Se il certificato è ritenuto valido, il Server procede ad interpretare la richiesta in accordo con la Security Policy e la Security Mode scelta.
    - Se il Security Mode è "None" allora la richiesta viene ricevuta senza applicare meccanismi di sicurezza
    - Se il Security Mode è "Sign" allora la firma della richiesta è verificata usando la Public Key associata all'Application Instance Certificate del Client
    - Se il Security Mode è "Sign&Encrypt" allora la richiesta è verificata usando la Public Key associata all'Application Instance Certificate del Client e poi decriptata usando la Private Key associata all'Application Instance Certificate del Server

    Se il Server accetta, invia la risposta al Client nelle stesse modalità usate per la request e viene stabilito il Secure Channel.

4. Una volta stabilito il Secure Channel il Client procede ad inviare una richiesta di Create Session al Server. Una volta creata si passa all'attivazione della stessa passando al Server le User Credential

Appendice: crittografia asimmetrica e firma digitale

Per rispondere a diverse esigenze, i messaggi tra due interlocutori possono essere crittografati o firmati (o anche firmati e poi crittografati).

La crittografia asimmetrica, conosciuta anche come crittografia a coppia di chiavi o crittografia a chiave pubblica è un tipo di crittografia dove i messaggi vengono codificati da degli attori che possiedono una coppia di chiavi:

- la "chiave privata", personale e segreta, che viene utilizzata per decodificare un documento criptato;
- la "chiave pubblica", che deve essere distribuita, e serve a cifrare un documento destinato alla persona che possiede la relativa chiave privata.

In un sistema di crittografia a chiave pubblica, chiunque può cifrare un messaggio usando la chiave pubblica del destinatario, ma tale messaggio può essere decifrato solo con la chiave privata del destinatario.

L'idea base della crittografia asimmetrica diviene più chiara se si usa un'analogia postale, in cui

- il mittente è Alice
- il destinatario Bob
- i lucchetti fanno le veci delle chiavi pubbliche
- le chiavi dei lucchetti recitano la parte delle chiavi private.

La comunicazione avverrebbe secondo questi passaggi:

1. Alice chiede a Bob di spedirle il suo lucchetto, già aperto. La chiave dello stesso verrà però gelosamente conservata da Bob.
2. Alice riceve il lucchetto e, con esso, chiude il pacco e lo spedisce a Bob.
3. Bob riceve il pacco e può aprirlo con la chiave di cui è l'unico proprietario.
4. Se viceversa Bob volesse mandare un pacco ad Alice, dovrebbe chiederle il lucchetto in modo da chiudervi il pacco. Alice ricevuto il pacco lo aprirebbe con la chiave privata di cui è l'unica proprietaria.

Si può notare come per segretare i pacchi ci sia bisogno del lucchetto (chiave pubblica) del destinatario mentre per decrittare viene usata esclusivamente la propria chiave del lucchetto (chiave privata), rendendo l'intero processo di criptazione/decriptazione asimmetrico.

La firma digitale invece, viene solitamente identificata come la tecnologia con cui si certifica l'autenticità di un documento o un dato che viene firmato da chi lo ha emesso. Il sistema della firma digitale prevede che il mittente che genera il messaggio utilizzi la sua chiave privata per generare un'informazione che, associata al messaggio, ne certifica la provenienza.

E' importante notare come sia la cifratura che la firma, si fondino sullo scambio della chiave pubblica. Nel caso del messaggio cifrato, la chiave pubblica è quella del destinatario, nel caso di un messaggio firmato invece è quella del mittente. In entrambi i casi il valore delle chiavi pubbliche non è confidenziale e la criticità sta nel garantire la loro l'autenticità. Si deve essere certi quindi che una certa chiave pubblica appartenga effettivamente all'interlocutore per cui si vuole cifrare o di cui si deve verificare la firma. Se, infatti, una terza parte prelevasse la chiave pubblica del destinatario sostituendola con la propria, il contenuto dei messaggi cifrati sarebbe svelato e non si riuscirebbe a verificare la validità di una firma digitale.

La distribuzione delle chiavi pubbliche è, pertanto, il problema cruciale della tecnologia a chiave pubblica che viene risolto tramite l'impiego dei certificati elettronici. I certificati a chiave pubblica (X.509) costituiscono, infatti, lo strumento affidabile e sicuro

attraverso cui le chiavi pubbliche vengono distribuite e rese note agli utenti finali con garanzia di autenticità ed integrità.
Cosa contiene un certificato X.509:

- Version
- Serial Number
- SignatureAlgorithm: algoritmo di crittazione usato
- Issuer: identifica la CA che ha prodotto e firmato il certificato
- Valid from/to: validità del Certificato
- Public Key: Chiave Pubblica del Certificato
- Signature: firma digitale creata dall'Issuer. La firma digitale viene posta utilizzando la Private Key dell'Issuer

Ad ogni certificato X.509 viene poi associata una Private Key.

1.5. OPC UA Server

1.5.1. OPC UA Server

Il Server di Movicon.NExT è conforme alla specifica OPC UA e consente la connettività di qualsiasi altra piattaforma o dispositivo OPC UA Client di terze parti supportando le specifiche:

- DA (Data Access)
- AC (Alarms and Conditions)
- HA (Historical Access)

La funzionalità OPC UA Server è opzionale, e richiede apposita abilitazione sulla licenza di prodotto acquistata.

Certificazione del Server

Il Server OPC UA di Platform.NExT è stato sottoposto con successo alla certificazione dal parte di OPC Foundation, tramite l'apposito OPC Certification Test Lab. L'ente preposto ha pertanto eseguito i testi di validazione per la verifica della piena conformità allo standard (rel. 1.02) e dei test di Performanc Stress e Load.
Platform.NExT ha quindi conseguito nel 2015 la Certificazione da parte di OPC Foundation, garantendo così agli utenti la garanzia di:

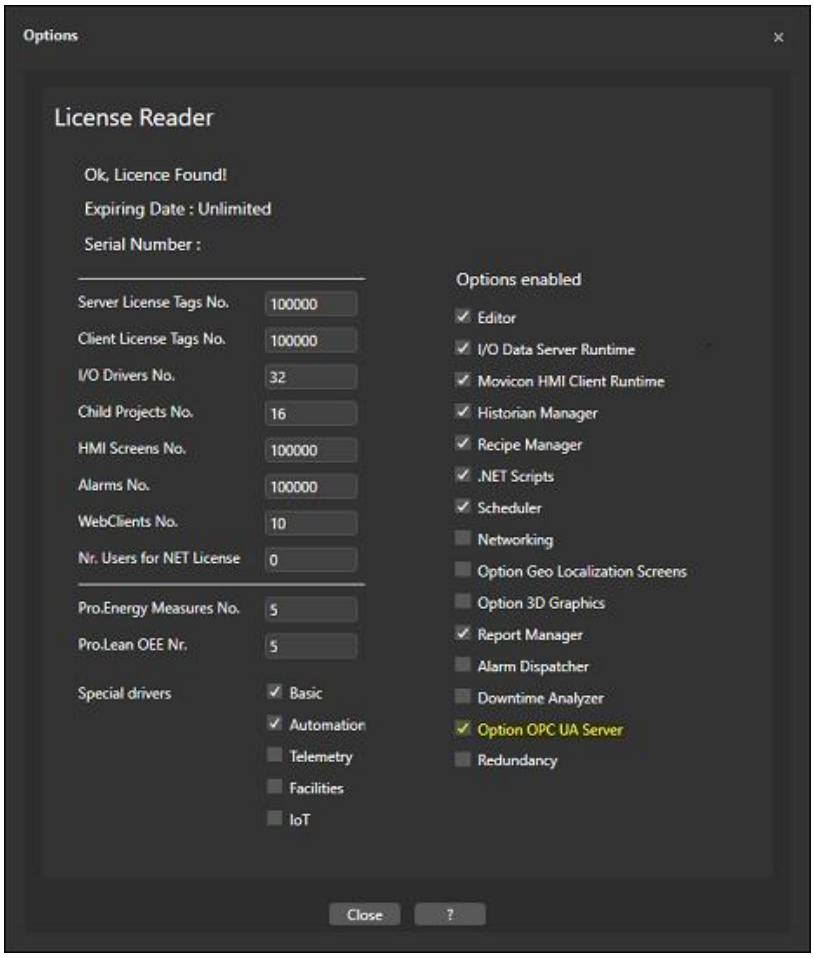
- Compatibilità
- Interoperabilità
- Robustezza
- Usabilità
- Efficienza

1. **Certificate Number: 1506CS006A**

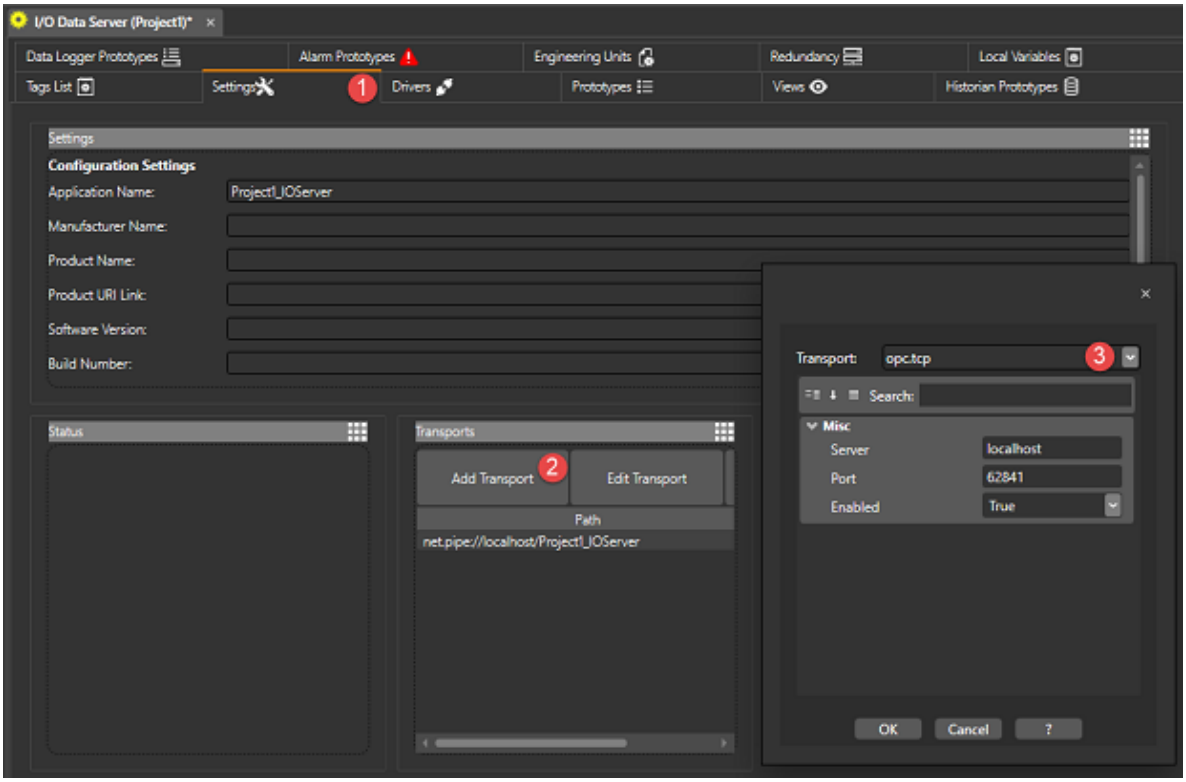
Accesso con Client OPC UA di terze parti

Affinchè le Tag dell'I/O Data Server del progetto di Platform.NExT possano essere visibili e collegate tramite il browser di un Client OPC UA di terze parti, è necessario tenere conto dei seguenti punti:

1. Nella licenza Runtime del Server deve essere abilitata l'opzione OPC UA Server



2. Nei Settings dell'I/O Data Server deve essere configurato un tipo di trasporto di rete adatto alla comunicazione OPC UA quali opc.tcp, http o https. Come illustrato nell'immagine seguente, dopo aver aperto la tab Settings dell'I/O Data Server (1), nell'area della definizione dei trasporti selezionare Add Transport (2). Nella dialog di definizione del trasporto di rete (3) selezionare il tipo di trasporto desiderato e configurarne le proprietà. Il nome del server è consigliabile rimanga "localhost" in modo che venga utilizzato l'hostname definito dal sistema operativo.



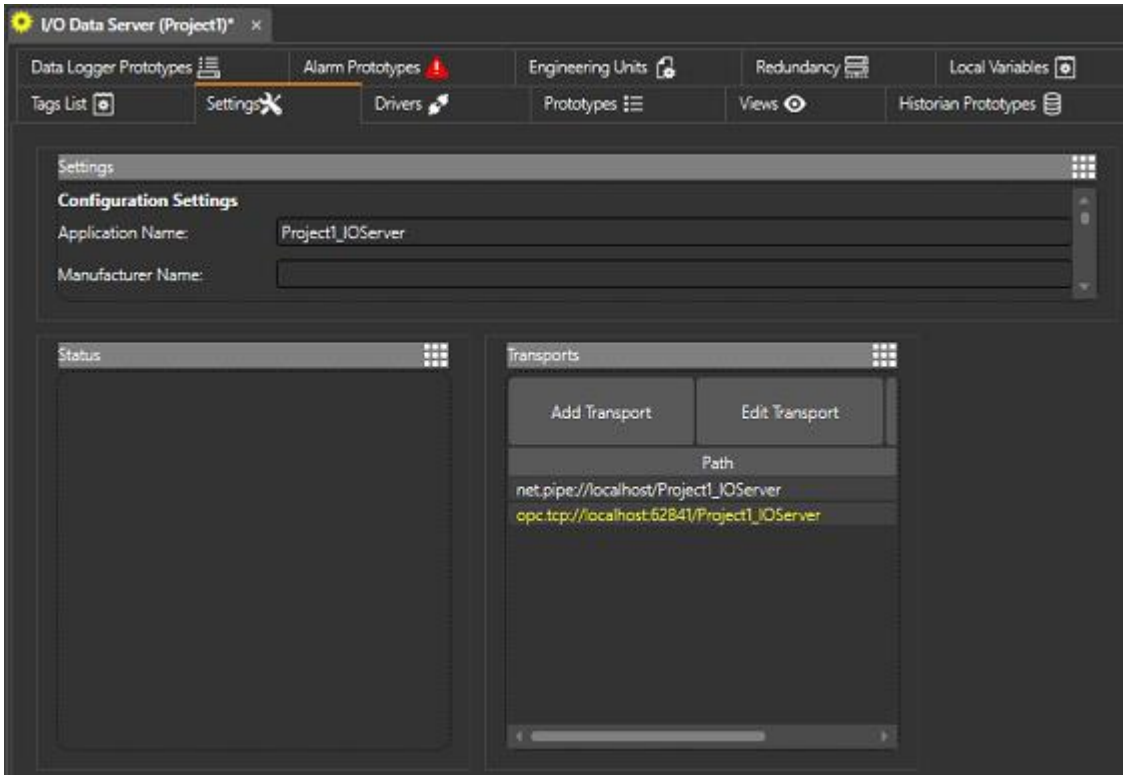
Con il Setup di Movicon.NExT viene installato anche l'OPC UA Local Discovery Server (LDS) dell'OPC Foundation. Questo Server fornisce l'infrastruttura necessaria per esporre i Server OPC UA avviati su una macchina e renderli visibili agli eventuali Client OPC UA. Nel caso si presentassero problemi di connessione o browsing tra Client e Server si consiglia di verificare che il servizio OPC UA Local Discovery Server (opcualds.exe) sia correttamente avviato e funzionante (il processo è presente nell'elenco dei servizi del Task Manager di Windows con il nome UALDS).

Session Endpoint, Security Mode e Certificati

L'I/O Data Server di NExT espone tre Endpoint con le seguenti opzioni dal punto di vista dell'Autenticazione lato Applicazione:

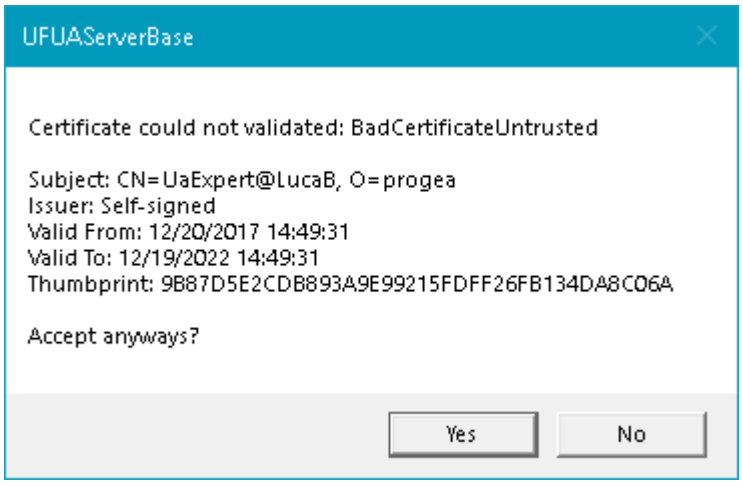
Security Mode	Security Policy	Security Level
None	-	Level0
Sign	Basic256	Level2
SignAndEncrypt	Basic128Rsa15	Level3

Per quanto riguarda l'autenticazione Utente invece sono supportati tutti i tipi di autenticazione definiti dallo standard OPC UA: Anonymous, Username/Password, User Certificate X.509
E' possibile modificare queste opzioni andando ad editare il file "UFUAServer.UAServer.Config" presente nella cartella di installazione di Movicon.NExT
Il Session Endpoint da utilizzare nell'OPC UA Client per l'attivazione del Secure Channel con il Server è uno tra quelli definiti nella lista dei Trasporti nella sezione Settings dell'I/O Data Server.



In riferimento all'immagine precedente il il Session url da utilizzare sarebbe: opc.tcp://<hostname_del_server>:62841

Se la Security Mode è di tipo Sign o SignAndEncrypt, il Channel verrà aperto dopo lo scambio dell'Application Instance Certificate. Il Server di Movicon.NExT supporta sia l'invio automatico del proprio certificato all'OPC UA Client ma anche di ricevere ed accettare il certificato del Client OPC UA in modo interattivo proponendo una dialog come quella mostrata nell'immagine seguente.



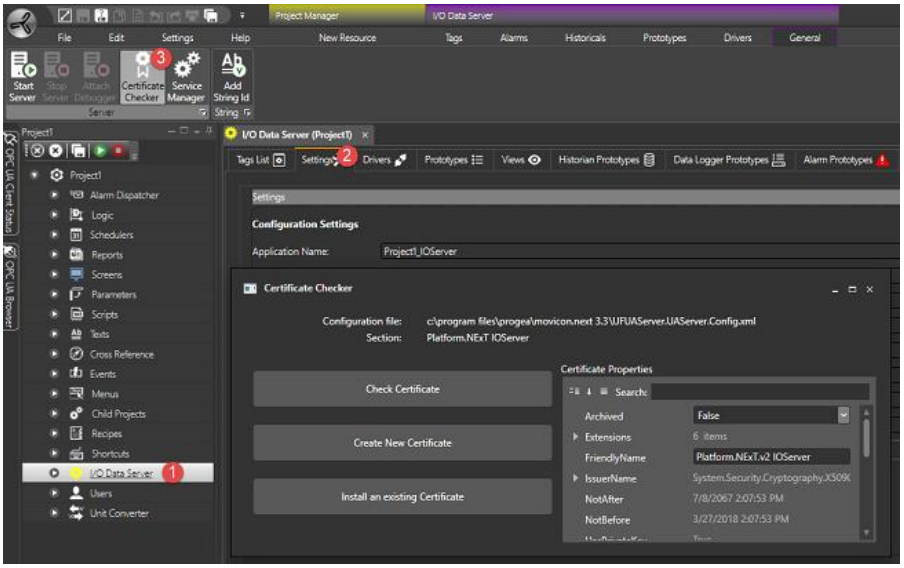
La dialog di conferma del certificato del Client è disponibile solo se l'I/O Data Server non è stato avviato come servizio. E' comunque possibile far sì che il Server accetti automaticamente certificati Untrusted impostando l'opzione <AutoAcceptUntrustedCertificates>true</AutoAcceptUntrustedCertificates> nel file "UFUAServer.UAServer.Config" presente nella cartella di installazione di Movicon.NExT

Se si volesse rendere il certificato del Client OPC UA come Trusted in modo permanente sul Server è necessario copiare il certificato del Client nella directory "%ProgramData%\OPC Foundation\CertificateStores\UA Applications\certs".

Se dovesse essere necessario fornire all'OPC UA Client l'Application Instance Certificate dell'I/O Data Server è possibile recuperare il file del certificato denominato "Platform.NExT.v2 IOServer" nella cartella "%ProgramData%\OPC Foundation\CertificateStores\MachineDefault\certs\"

Gestione del certificato

Il certificato della parte Server di Movicon.NExT denominato "Platform.NExT.v2 IOServer" può essere controllato, rinnovato o sostituito tramite il tool di configurazione accessibile (1) espandendo l'I/O Data Server, (2) selezionando la voce Settings ed infine (3) Selezionare Certificate Checker.



1.6. OPC UA Client

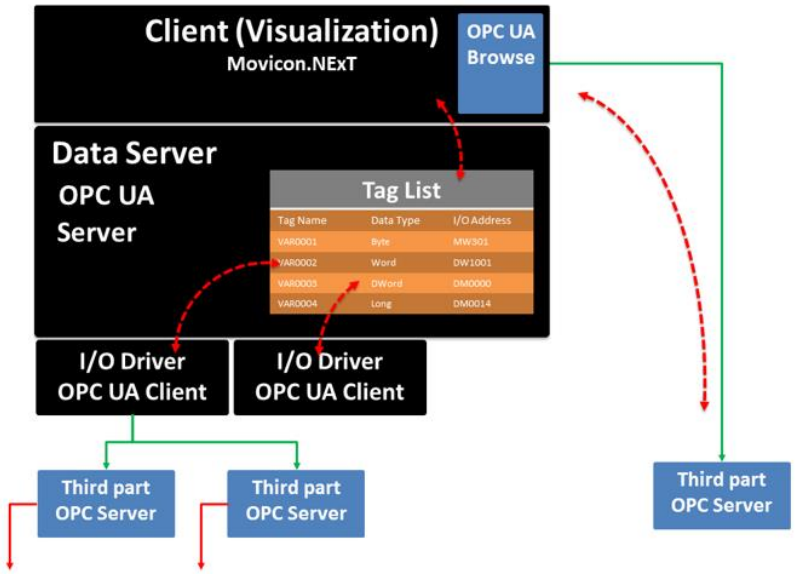
1.6.1. OPC UA Client

Il client di Movicon.NExT è conforme alla specifica OPC UA e consente la connettività verso qualsiasi altra piattaforma o dispositivo OPC UA Server di terze parti supportando le specifiche:

- DA (Data Access)
- AC (Alarms and Conditions)
- HA (Historical Access)

La connettività ai Server OPC UA di terze parti

La connettività Client-Server secondo lo standard OPC UA può avvenire sia lato Client della piattaforma Movicon.NExT che dal lato Server, a seconda della necessità.



L'integrità e la confidenzialità dei dati scambiati con OPC UA Server di terze parti sono assicurate dallo scambio dei Certificati Applicazione in fase di Autenticazione dell'Applicazione (Sign o SignAndEncrypt).
A livello di Autenticazione Utente invece non è supportata l'Autenticazione tramite Certificati X.509 pur rimanendo valide le modalità Anonymous e Username/Password.

Connessione lato Client, tramite OPC UA Browser

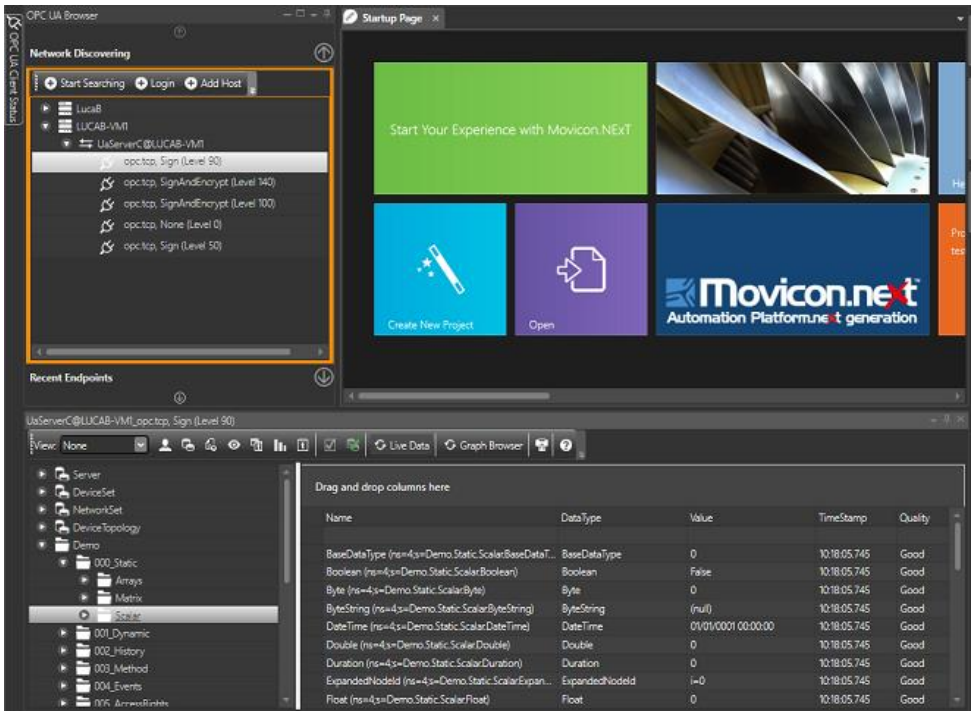
Nel client di visualizzazione è possibile associare simboli ed oggetti grafici presenti nei sinottici
non solo a Tag presenti nell'Address Space del Server ma anche connettendoli direttamente ad un Server OPC UA di terze parti tramite l'OPC UA Browser.
Questa modalità consente infatti di eseguire l'associazione di un simbolo od oggetto grafico, ad un Tag scelto tramite il browsing live degli item di un OPC UA Server.
In questo modo è anche possibile utilizzare architetture "solo Client", dove viene utilizzata solo la parte grafica della piattaforma: il Client di visualizzazione Movicon.NExT.

Connessione lato Server, tramite Driver OPC UA Client

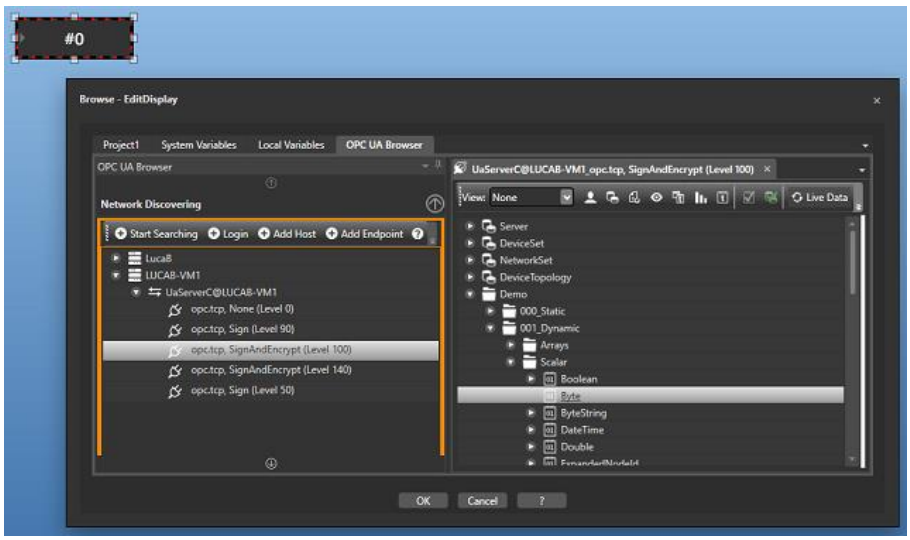
Per poter rendere disponibili i dati provenienti da altri OPC UA Server, tramite il Server di Movicon.NExT è necessario invece utilizzare l'apposito driver di comunicazione OPC UA Client.
In questo modo, i dati saranno disponibili tramite le Tag definite/importate nell'Address Space della piattaforma, e quindi disponibili all'intero sistema.

1.6.2. Connessione lato Client, tramite OPC UA Browser

E' possibile associare direttamente, da qualsiasi oggetto o simbolo di visualizzazione o comando inseribile nei sinottici, un item esposto da un qualsiasi Server OPC UA compatibile. In questo modo, l'elemento grafico sarà associato direttamente all'Item del Server OPC UA, e non ad una Tag di progetto.
Per ottenere questo, occorre semplicemente utilizzare la finestra di selezione dei Tags, e da qui selezionare il Tab corrispondente al browser OPC UA.



ma anche per associare direttamente un item a qualsiasi oggetto, simbolo o comando presente in un sinottico.
In questo modo, l'elemento grafico sarà associato direttamente all'item del Server OPC UA e non ad una Tag del progetto. Per ottenere questo, occorre semplicemente utilizzare la finestra di selezione dei Tag e da qui selezionare il Tab corrispondente al OPC UA Browser.

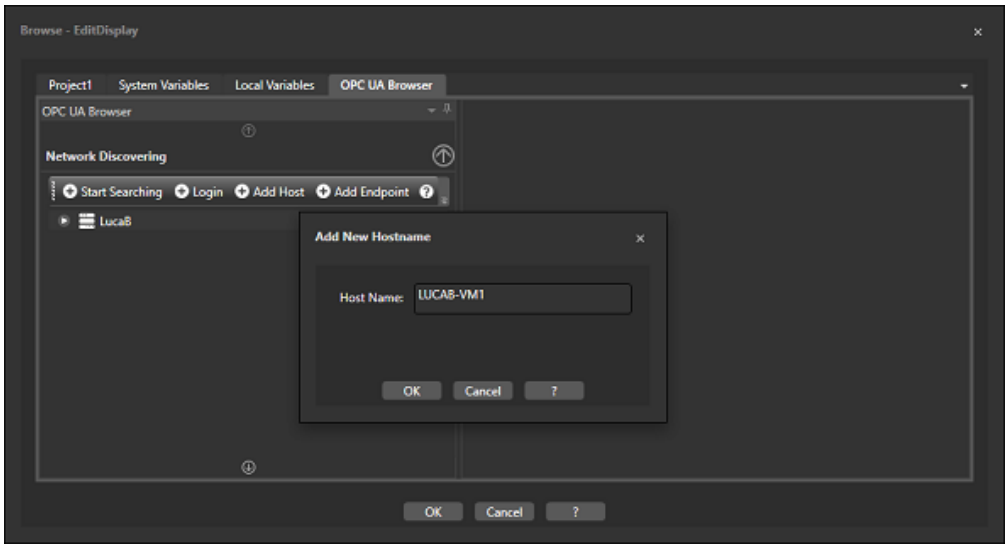


L'OPC UA Browser è diviso in due sezioni: a sinistra la finestra di selezione dell'Endpoint offerti dall'OPC UA Server e a destra la finestra di selezione delle Tag relativamente al Secure Channel stabilito.
Nell'immagine precedente, sono ad esempio visibili due Local Discovery Server nominati "LUCAB" e "LUCAB-VM1". Nel Local Discovery Server "LUCAB-VM1" risulta registrato un OPC UA Server con Application Name "UaServerC@LUCAB-VM1" il quale espone cinque diversi Session Endpoint. Ogni Session Endpoint viene identificato tramite la terna: "Transport Protocol, Security Mode (Security Level)" ad esempio "opc.tcp, SignAndEncrypt (Level 100)"
Nella finestra di selezione dell'Endpoint sono disponibili i seguenti comandi:

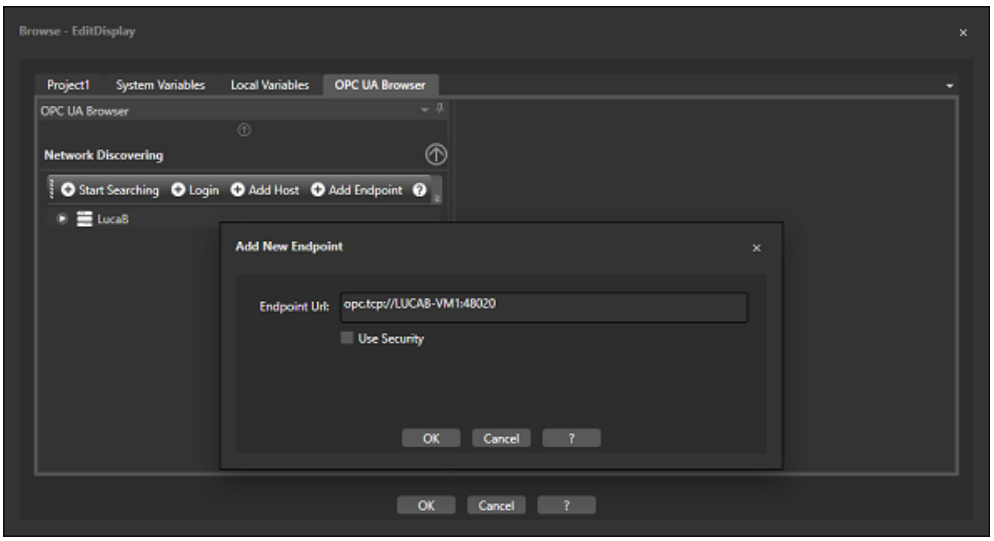
- Start Searching: effettua una scansione della rete locale per verificare la presenza di eventuali OPC UA Local Discovery Server
- Login: permette di effettuare un Accesso Utente tramite Credenziali (username/Password) all'Endpoint selezionato, se non richiesto automaticamente in fase di connessione
- Add Host: permette di definire manualmente l'Hostname (o l'indirizzo IP) di un Local Discovery Server dal quale ottenere tutti i Session Endpoint disponibili del Server
- Add Endpoint: permette di definire manualmente l'url dell'Endpoint al quale ci si vuole connettere.

Connessione ad un Endpoint

Al fine di selezionare una item dal OPC UA Server dalla finestra di selezione, è necessario prima stabilire un Secure Channel tramite l'attivazione di un Session Endpoint.
La selezione ad un Endpoint avviene quindi tramite uno dei tre comandi "Start Searching", "Add Host" oppure "Add Endpoint".
L'aggiunta manuale dell'Host è utile quando non si conoscono gli Endpoint messi a disposizione del Server:



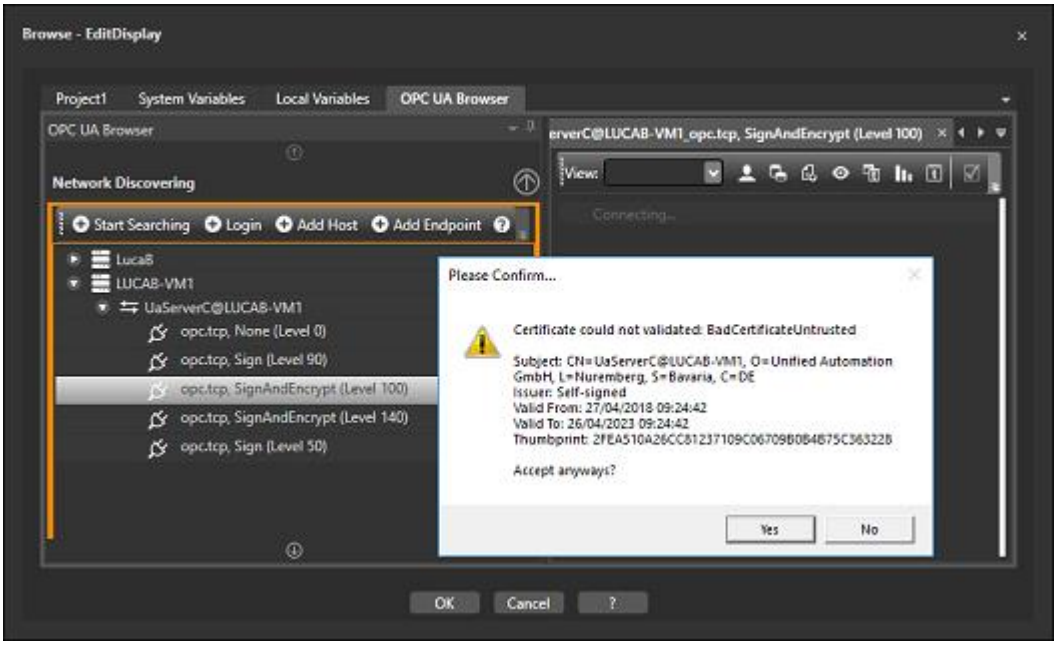
mentre se si conosce già l'url dell'Endpoint è possibile selezionare "Add Endpoint" dove definire l'url nella forma: "<Transport Protocol>://<hostname>:<port>", come ad esempio "opc.tcp://server1:48020".



Selezionando l'opzione "Use Security", verrà scelto l'Endpoint con Security Level maggiore tra quelli resi disponibili dal Server con un Security Mode di tipo Sign o SignAndEncrypt.

L'uso dell'hostname anzichè dell'indirizzo IP sia nella definizione manuale dell'Host che del Endpoint è preferibile specie nei casi in cui l'Endpoint selezionato utilizzi Security Mode di tipo Sign o SignAndEncrypt. La creazione di un Secure Channel in questi casi, infatti, prevede lo scambio dei certificati pubblici tra Client e Server. La necessità di utilizzare hostname (o l'indirizzo IP) è dipendente da come è definito il certificato di sicurezza dell'OPC UA Server a cui ci si connette.

Dopo aver aggiunto un Endpoint oppure un Host che espone diversi Endpoint, è quindi necessario selezionarne uno dall'albero di sinistra con doppio click per attivare il canale di comunicazione verso l'OPC UA Server.
Se la Security Mode è di tipo Sign o SignAndEncrypt, il Channel verrà aperto dopo lo scambio dell'Application Instance Certificate. L'OPC UA Browser di Movicon.NExT consente di ricevere ed accettare il certificato del Server OPC UA in modo interattivo proponendo una dialog come quella mostrata nell'immagine seguente:



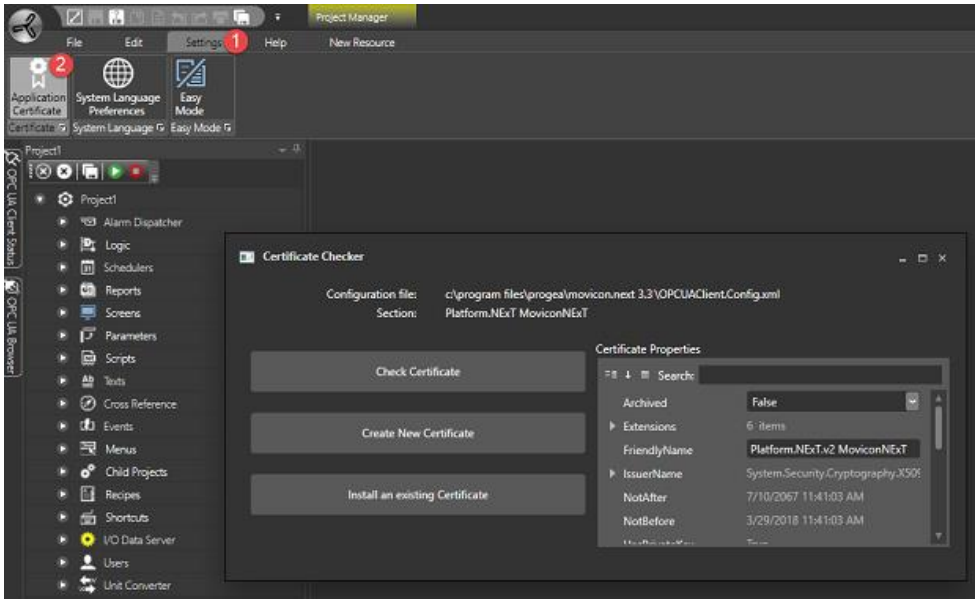
In questo modo l'ambiente di sviluppo Movicon.NExT (che agisce come Client) considererà valido il certificato per tutta la durata della Session.

E' comunque possibile far sì che il l'OPC UA Browser accetti automaticamente certificati Untrusted impostando l'opzione <AutoAcceptUntrustedCertificates>true</AutoAcceptUntrustedCertificates>
nel file "OPCUAClient.Config" presente nella cartella di installazione di Movicon.NExT

Movicon.NExT oltre a ricevere il certificato contentente la chiave pubblica del Server, restituisce al Server stesso anche il proprio certificato. A seconda della configurazione però l'OPC UA Server potrebbe rifiutare il certificato e posizionarlo nella propria lista dei certificati rejected. Si rende necessario quindi spostare il certificato dalla lista dei certificati Rejected alla lista dei certificati Trusted oppure copiare a mano il file del certificato di Movicon.NExT sul Server secondo le procedure previste.
Il certificato da utilizzare in questo caso è il file "Platform.NExT.v2 MoviconNExT" che si trova nella directory "%ProgramData%\OPC Foundation\CertificateStores\MachineDefault\certs\"

Gestione del certificato

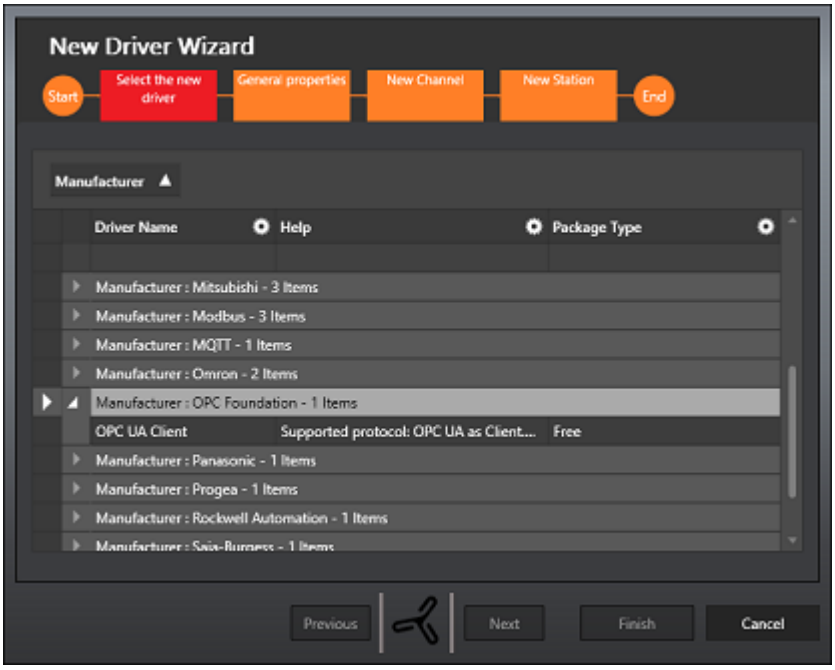
Il certificato della parte Client di Movicon.NExT denominato "Platform.NExT.v2 MoviconNExT" può essere controllato, rinnovato o sostituito tramite il tool di configurazione accessibile dal Ribbon tramite la tab Settings (1) e selezionando la voce Application Certificate (2).



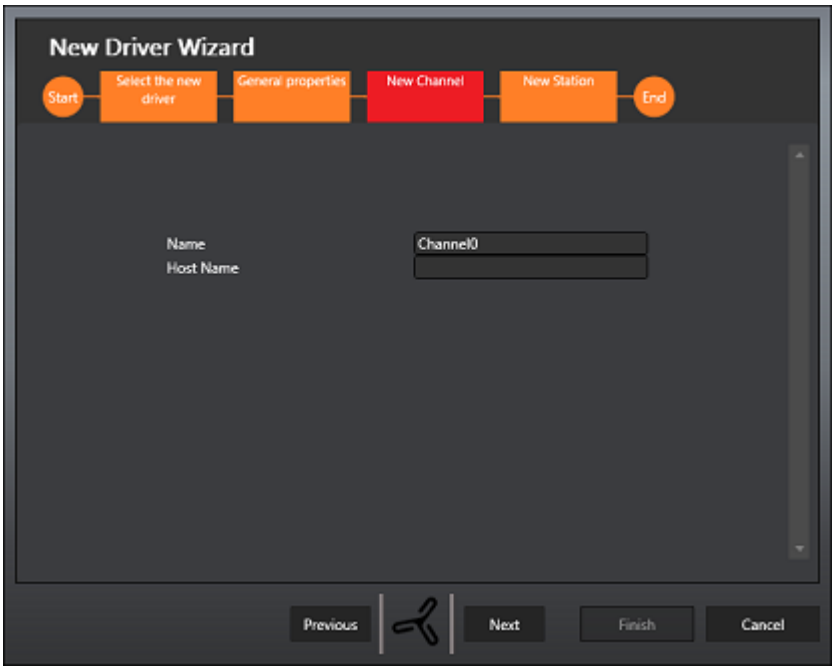
1.6.3. Connessione lato Server, tramite Driver OPC UA Client

Tipicamente la connettività verso un OPC UA Server avviene tramite il driver di comunicazione OPC UA Client.
In questo modo il Server di Movicon.NExT diventa Client di un secondo Server OPC UA.

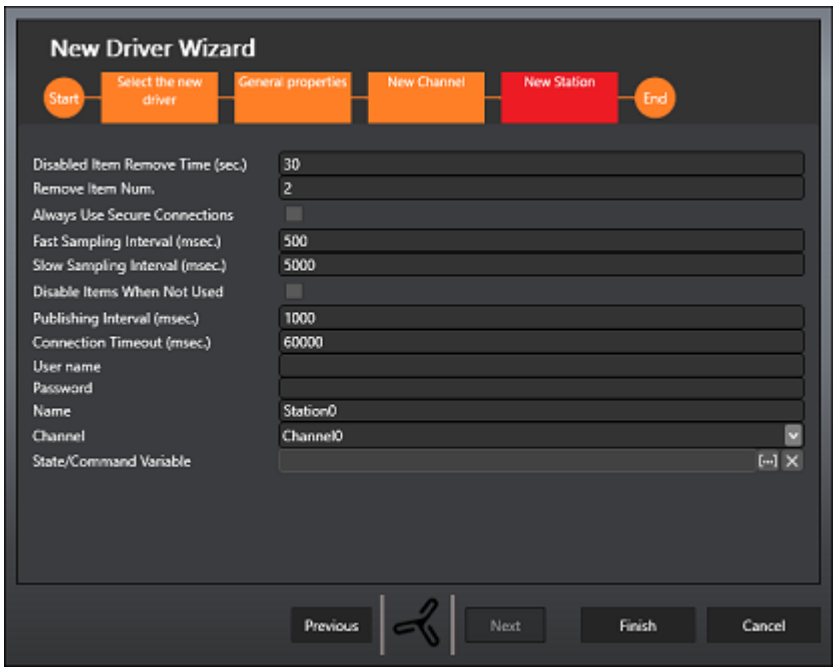
Tramite il driver OPC UA Client, gli item selezionati saranno quindi disponibili come Tag nell'Address Space dell'I/O Data Server di progetto.



La definizione del Channel prevede, oltre la definizione del Nome, la definizione dell'Hostname che verrà utilizzato nella connessione agli item. Se le tag vengono importate tramite OPC UA Browser, questo Hostname sarà utilizzato per sostituire l'hostname definito nell'url dell'item.



Nella definizione della Station sono disponibili i seguenti parametri:



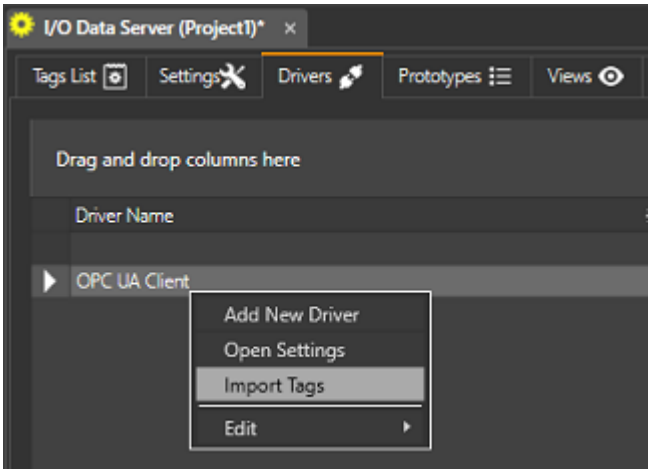
1. Disabled Item Remove Time (Sec.): Imposta il tempo di ritardo con il quale verranno rimosse le sottoscrizioni OPC UA degli item non attivi. Il valore è espresso in secondi
- Remove Item Num.: Imposta il numero di Item OPC UA non attivi da rimuovere ad ogni intervallo di tempo
- Always Use Secure Connections: seleziona l'Endpoint con Security Level maggiore tra quelli resi disponibili dal Server con un Security Mode di tipo Sign o SignAndEncrypt.

In caso di Security Sign o SignAndEncrypt è necessario scambiare i certificati con le chiavi pubbliche. Si dovrà quindi fornire all'OPC UA Server il certificato usato dal driver OPC UA Client denominato "Platform.NExT.v2 IOServer" disponibile nella cartella "%ProgramData%\OPC Foundation\CertificateStores\MachineDefault\certs\" mentre si dovrà copiare il certificato ".der" dell'OPC UA Server nella cartella "%ProgramData%\OPC Foundation\CertificateStores\UA Applications\certs\"

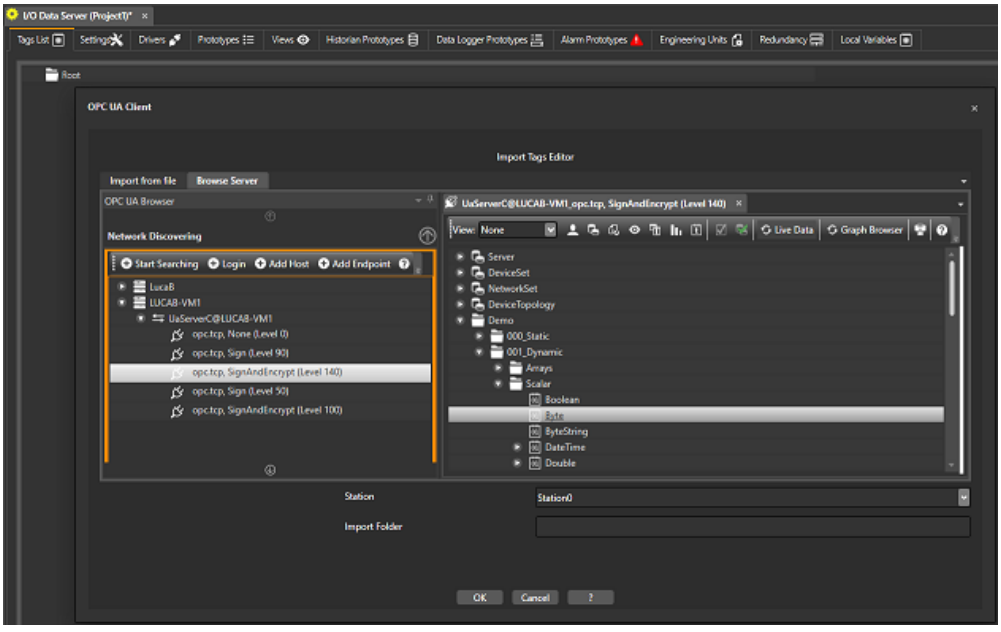
- Fast Sampling Interval: Definisce la frequenza d'aggiornamento per una variabile esistente e in uso
- Slow Sampling Interval: Definisce la frequenza d'aggiornamento per una variabile esistente che sta per andare non in uso.
- Disable items When Not Used: Imposta i Tag come "Inactive" quando non sono in uso.
- Publishing Interval: Tempo di notifica dei Tag verso il Server. Il valore è espresso in millisecondi.
- Connection Timeout: timeout della connessione al Server
- Username: Username da utilizzare per l'Autenticazione OPC UA a livello utente
- Password: Password da utilizzare per l'Autenticazione OPC UA a livello utente
- Name: Nome della Station
- Channel: Canale a cui la Station fa riferimento
- State/Command Variable: Assegnando il nome ad una variabile numerica di supervisione (tipo Byte consigliato) a questa proprietà, è possibile controllare lo stato di comunicazione del canale selezionato.

Bit 0 (State)	Connection Channel 0= connected 1= not connected
Bit 1 (State)	Primary Host Error State 0=Active 1=Inactive
Bit 2 (State)	Backup Host Error State 0=Active 1=Inactive
Bit 3 (State)	Connected Host 0=Active 1=Inactive

Una volta definita la configurazione del driver OPC UA Client, è possibile importare le Tag dal Server OPC UA selezionando Import Tags dal menu contestuale o dal Ribbon:



L'importazione può avvenire tramite "Import from file" oppure tramite "Browse Server" che sfrutta l'OPC UA Browser già visto in precedenza.



⚠ **Nel caso in cui si utilizzi il driver OPC UA Client per la comunicazione con un OPC UA Server e si importino le Tag nel progetto tramite l'OPC UA Browser, potrebbe essere necessario copiare sull'OPC UA Server entrambi i certificati "Platform.NExT.v2 IOServer" (usato dal driver) e "Platform.NExT.v2 MoviconNExT" (usato dall'OPC UA Browser) entrambi presenti in "%ProgramData%\OPC Foundation\CertificateStores\MachineDefault\certs\"**

Gestione del certificato

Il certificato usato dal driver OPC UA Client è lo stesso della parte Server di Movicon.NExT denominato "Platform.NExT.v2 IOServer". Fare riferimento a capitolo "Gestione del Certificato" nel capitolo OPC UA Server

INDUSTRIAL AUTOMATION SOFTWARE

progea

Progea Srl
Via D'Annunzio, 295
I-41123 Modena
info@progea.com
Tel +39 059 451060

Progea International SA
via Sottobisio, 28
6828 Balerna (CH)
international@progea.com
Tel +41 91 96 76 610

Progea Deutschland GmbH
Marie-Curie Str., 12
D-78048 VS Villingen
info@progea.de
Tel +49 (0)7721 99838 0

Progea North America Corp.
2380 State Road 44, Suite C
Oshkosh, WI 54904
info@progea.us
Tel. +1 (888) 305-2999