OPC UA Server Service

Information on how to set up a TagProvider connection with OPC UA Server.

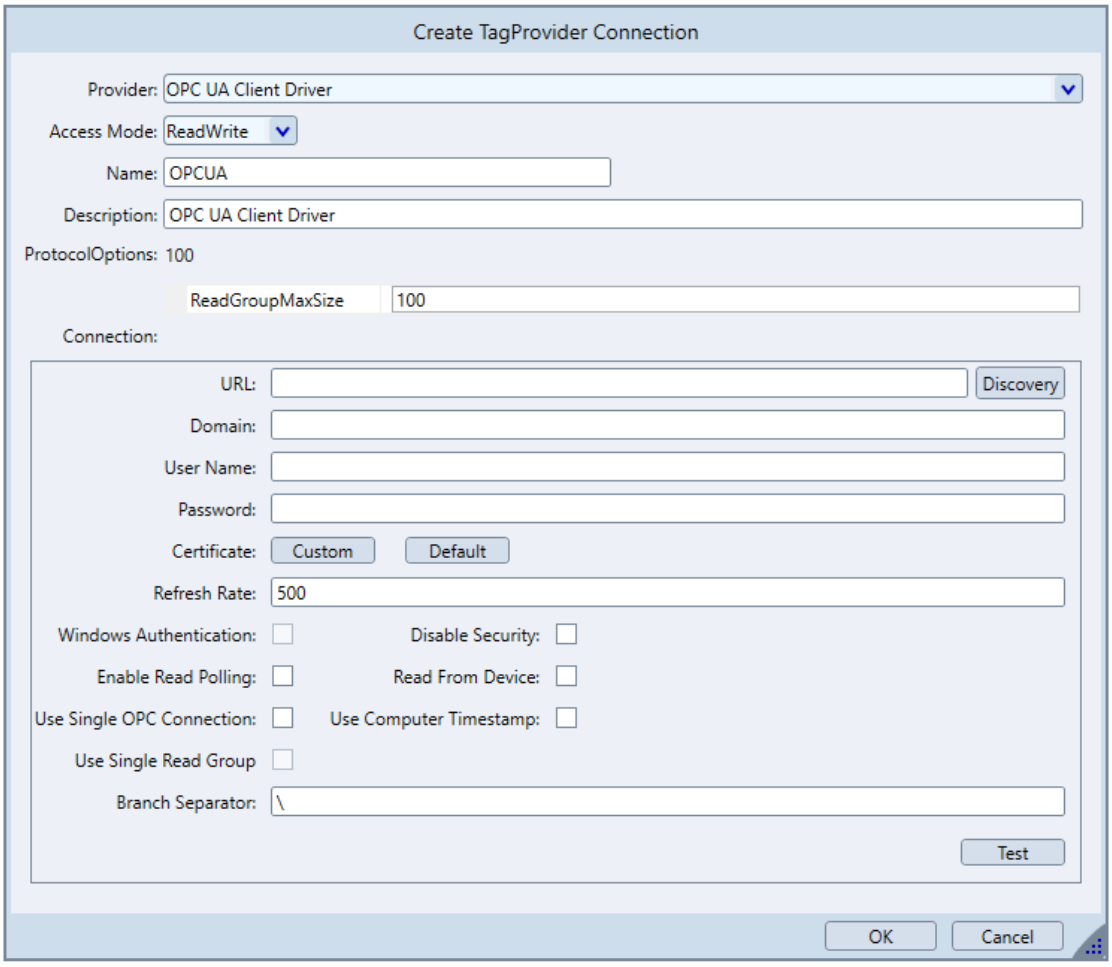
Reference Modules UNS UI Services OPC UA

The OPC UA Service allows you to dynamically connect to servers using the OPC UA protocol in real-time. This enables the integration and visualization of data and other components of your application without the need to create local tags.

Configuration
- Connection Parameters
- Certificate Generator

Configuration

To set up a new TagProvider connection navigate to **Unified Namespace/TagProvider Connections** and press the add button to open the New TagProvider Connection dialog. You can also create your TagProvider Connection directly from the AssetTree. To do this, navigate to **Unified Namespace/Assets Tree** and press  to open the New TagProvider Connection dialog. In the configuration dialog, select the **OPC UA Client** on the Provider field and insert the required connection parameters. After that, press the Ok button to finish the provider configuration. Creating the connection adds no folder to the Asset Tree. To use its data with Asset(), open **Unified Namespace / Assets Tree**, select a folder (create one first if needed) and click **Insert data from TagProvider** on the toolbar: the folder linked to the TagProvider is created inside the selected folder. A linked folder cannot sit at the root of the Asset Tree.



Connection Parameters

Field	Description

Provider	Specifies the system the TagProvider will interface with. In this case, it interfaces with the OPC UA Client Communication Driver.
Access Mode	Determines permissions for interacting with tags. Read allows data reading. Write permits data writing. ReadWrite enables both reading and writing.
Name	User-defined name for the TagProvider instance. Identifies the specific configuration in the system.
Description	Provides a short description for the TagProvider's purpose or the systems it interfaces with. Provides context for its usage.
ProtocolOptions	Defines specific communication settings for the connection to the device.
ReadGroupMaxSize	Specifies the maximum size of the read group.
Connection	Defines connection string for the OPC UA server settings, including URL, domain, username, password, certificate, and additional options.

Field	Description
URL	Specifies the URL of the OPC UA server. On URL, when clicking the Discovery button, open the **OPC UA - Endpoint Browser** dialog. In this dialog, have the IP Address, Port Number, and a search button.
Domain	Specifies the domain for the OPC UA server.
UserName	Specifies the username for authentication with the OPC UA server.
Password	Specifies the password for authentication with the OPC UA server.
Certificate	Specifies the certificate type used for the connection. Options include Custom and Default. When you click the Custom button, the **OPC UA Client Certificate Generator** opens with the following configuration parameters: Application Name, Organization Name, Domain Name (dns), Issuer Key File, Issuer Password, Key Size (options: RSA 1024, RSA 2048, RSA 4096), Hash Algorithm (options: SHA 1, SHA 256), Lifetime (Months), and Server Certificate. The Import button opens a file browser for Der Files (*.der). The Store Path parameter specifies where the certificates will be saved. Each field allows for customization to fit specific requirements for secure communication. Refer OPC UA Client Certificate Generator for detailed information. When you click the Default button, the OPC UA Client Certificate Generator opens with pre-configured parameters. The Application Name is TRunModule, the Organization Name is TATSOFT, and the Domain Name is _ComputerName_. The Key Size is set to RSA 1024 and the Hash Algorithm to SHA1. The Lifetime is 12 months. The Store Path is C:\Users\User\Documents\FrameworX\ToolsSettings\Certificates\Own. The Import button opens a file browser for Der Files (*.der). The Issuer Key File and Issuer Password fields are left blank for manual input.
Refresh Rate	Sets the rate at which data is refreshed from the OPC UA server.
Windows Authentication	Enables Windows authentication for the connection.
Disable Security	Disables security features for the connection.
Enable Read Polling	Enables polling for reading data from the OPC UA server.
Read From Device	Enables reading data directly from the device.
Use Single OPC Connection	Uses a single connection for OPC communications.
Use Computer Timestamp	Uses the computer's timestamp for data entries.

Use Single Read Group	Uses a single read group for all data requests.
Branch Separator	Specifies the character used to separate branches in the OPC UA server's namespace.
Test	Verifies the connection settings to ensure successful communication with the specified server.

Certificate Generator

Parameter	Description	Explanation
Application Name	Specifies the name of the application using the OPC UA client certificate.	Identifies the specific application using the certificate. Important for managing multiple applications within the same organization.
Organization Name	Specifies the name of the organization that owns the application.	Provides the name of the organization, adding a layer of identification and credibility to the certificate.
Domain Name (dns)	Specifies the domain name associated with the application.	Ensures the certificate is tied to a specific domain, helping to prevent impersonation and unauthorized access.
Issuer Key File	Specifies the file path to the issuer's key file.	Contains the cryptographic key used by the certificate authority (CA) to sign the certificate, validating its authenticity.
Issuer Password	Specifies the password required to access the issuer's key file.	Protects the issuer key file from unauthorized use by requiring a password to access it.
Key Size	Specifies the size of the RSA key used in the certificate. Options: RSA 1024, RSA 2048, RSA 4096.	RSA 1024: Moderate security, less secure by modern standards due to advancements in computational power. RSA 2048: Currently considered secure and widely used. Balances security and performance. RSA 4096: Higher security but requires more processing power. Suitable for applications needing maximum security.
Hash Algorithm	Specifies the hash algorithm used to create the certificate signature. Options: SHA 1, SHA 256.	SHA 1: Produces a 160-bit hash value. Considered insecure due to vulnerabilities allowing collision attacks. SHA 256: Produces a 256-bit hash value. Part of the SHA-2 family and currently considered secure, offering robust protection against collision and preimage attacks.
Lifetime (Months)	Specifies the validity period of the certificate in months.	Defines how long the certificate remains valid before it needs to be renewed or replaced, ensuring ongoing security.
Server Certificate	Specifies the server certificate file used for secure communication.	The actual certificate used by the server to establish a secure connection. Verifies the server's identity to the client.
Import Button	Opens a file browser dialog to select and import DER files (*.der) for the server certificate or other relevant files.	Provides an easy way to import existing certificates or related files, enhancing usability and efficiency by allowing users to browse and select files from their system.
Store Path	Specifies the file path where the generated or imported certificates will be stored.	Specifies where the certificates are saved, aiding in organization and retrieval for future use.
RSA (Rivest-Shamir-Adleman)	Asymmetric cryptographic algorithm used for secure data transmission. Involves a pair of keys: a public key for encryption and a private key for decryption.	Algorithm: Uses two keys that are mathematically linked. The public key can be shared openly, while the private key must be kept secret. Key Generation: Generated using two large prime numbers. Security relies on the difficulty of factoring their product. Encryption/Decryption: Data encrypted with the public key can only be decrypted with the private key. Suitable for securing sensitive data, digital signatures, and key exchange in protocols such as SSL/TLS.

SHA (Secure Hash Algorithm)	Family of cryptographic hash functions designed to ensure data integrity. Produces a fixed-size hash value (digest) from variable input data, unique to the input data.	Algorithm Variants: Includes SHA-1, SHA-256, SHA-512, each differing in the length of the hash value and security level. Hash Function: Takes input and produces a unique, fixed-size string of characters. Any change in the input results in a completely different hash. Collision Resistance: Designed to be collision-resistant, making it infeasible to find two different inputs that produce the same hash output. Used in security applications and protocols, including SSL/TLS, digital signatures, and integrity verification.

In this section...