# FactoryTalk Linx OPC UA Connector analysis for namespace from a configured OPC UA Server.

# Introduction

A FactoryTalk Linx OPC UA Connector acts as a gateway between one or more Open Platform Communications Unified Architecture (OPC UA) Servers and a FactoryTalk application, providing OPC UA data as tags, alarms delivered via FactoryTalk Alarms, and method calls using tags to trigger the method. The FactoryTalk Linx OPC UA Connector will translate requests from FactoryTalk Live Data clients into appropriate OPC UA client requests to an OPC UA Sever and translate the resulting OPC UA Server responses to deliver data to FactoryTalk Live Data clients.

OPC UA Clients connect to endpoints exposed by the UA Server to create a session, through which clients can access data, alarms, events, methods, and other functionality. An OPC UA server can expose multiple endpoints, each configured for different security settings, ranging from none to fully encrypted. This provides flexibility in how clients access the server.

Each endpoint consists of the transport protocol, a Universal Record Locator (URL) or IP address, a communications port, and security settings (cipher suite). Endpoints are specified using the syntax *<transport>://URL:port/optionalSuffix*. For example, *opc.tcp://hostname:4840* or *opc.tcp://127.0.0.1:4840/MyServer*.

The FactoryTalk Linx OPC UA Connector aggregates data from one or more UA Servers, and delivers tag and alarm values to FactoryTalk applications. One OPC UA Server can be accessed by multiple UA Connectors. Similarly, multiple OPC UA Servers can be accessed by a single FactoryTalk Linx OPC UA Connector.

The FactoryTalk application, FactoryTalk Linx OPC UA Connector, and OPC UA Servers can reside on the same computer, separate computers, or on a combination of computers. However, only one FactoryTalk Linx OPC UA Connector service will run on each computer.

## OPC UA Namespace

All items in an OPC UA Server are called *Nodes*. A *Node* may be a folder, a data item or process variable, a *DataType* definition, a method, an alarm, etc. Attributes in the *Node*, and properties attached to the *Node*, are used to provide meaning to what the *Node* represents.

Each *Node* is uniquely identified within the UA Server by a *NodeId*. A *NodeId* is a value constructed from several parts – a namespaceURI, an identifier that is unique within the namespaceURI, and the type (string, integer, *Guid*, or *ByteString*) of identifier. The combination of the namespaceURI and identifier creates a unique *NodeId* within the UA Server, which eliminates all ambiguity in identifying *Nodes*.

## LiveData Namespace

All items in LiveData are identified by a tag, which is a string that contains a path component and ends with an identifier unique to the final portion of the path.

## Namespace upload

To facilitate communication, the FactoryTalk Linx OPC UA Connector must fully upload the OPC UA namespace (all *Node* and *DataType* definitions) from each OPC UA server. This is needed for two reasons.

First, FactoryTalk clients make data requests using tag names but OPC UA items are accessed using the *NodeId* assigned to each *Node* in an OPC UA server. The FactoryTalk Linx OPC UA Connector receives requests from LiveData clients using tag names, and performs lookups to determine the appropriate *NodeIds* to use.

Second, the FactoryTalk Linx OPC UA Connector will add all the OPC UA Servers tag definitions into the FactoryTalk Live Data namespace to permit FactoryTalk Live Data clients to browse the OPC UA servers tag list, and choose data items to be requested.

## Namespace upload failure

To upload an OPC UA server’s namespace, the FactoryTalk Linx OPC UA Connector will traverse the entire namespace of the OPC UA Server layer by layer retrieving tag , methods, and *DataType* definitions. This can be a time-consuming process that could fail, exclude some expected data, or have some data items to be interpreted differently than expected:

Types of failures:

* OPC UA server rejects the connection
* Limited OPC UA Server processing bandwidth
* Invalid data definitions in the OPC UA Server
* Large OPC UA Server namespace(s)
* Missing OPC UA Server tags
* Values from tags with unsupported DataTypes are converted to strings (not really a failure)

### OPC UA server rejects the connection

The FactoryTalk Linx OPC UA Connector must first establish a connection to the OPC UA Server before it uploads the namespace. The connection attempt can fail for a range of reasons – broad categories include communications problems, security, internal errors, and resource constraints:

* Incorrect path to the OPC UA Server
	+ Verify FactoryTalk Linx OPC UA Connector’s endpoint configuration has correct OPC UA servers Universal Record Locator (URL) string and TCP port
	+ Use an OPC UA Sample client to confirm the URL
	+ Typical error codes:
		- *Bad\_ServerUriInvalid, Bad\_SecureChannelIdInvalid, Bad\_NonceInvalid*
* System clocks that are unsynchronized and too far apart.
	+ Both the OPC UA Server and Client have a maximum allowable drift so expiry times for *Certificates* and *Messages* can be checked. A suitable default is usually around 5 minutes.
* Incorrect security policy configuration
	+ Verify both the OPC UA Server and the FactoryTalk Linx OPC UA Connector endpoint have identical security policy configurations
	+ Switch the security policy and mode in both the UA Server and the FactoryTalk Linx OPC UA Connector to “none” for testing.
* The FactoryTalk Linx OPC UA Connector’s certificate is not approved on the OPC UA Server
	+ Use the OPC UA server’s user interface to verify and approve the certificate
	+ Switch the security policy and mode in both the UA Server and the FactoryTalk Linx OPC UA Connector to “none” for testing.
* The FactoryTalk Linx OPC UA Connector’s certificate has expired or is invalid
	+ Regenerate or import a new FactoryTalk Linx OPC UA Connector certificate
	+ Switch the security policy and mode in both the UA Server and the FactoryTalk Linx OPC UA Connector to “none” for testing.
* The OPC UA Server’s certificate has expired or is invalid
	+ Regenerate or import a new certificate in the OPC UA server’s configuration user interface
	+ Switch the security policy and mode in both the UA Server and the FactoryTalk Linx OPC UA Connector to “none” for testing.
	+ Note the FactoryTalk Linx OPC UA Connector will automatically trust a valid certificate for an incoming OPC UA server the UA Connector is configured to interface with.
* The FactoryTalk Linx OPC UA Connector’s user ID and password credentials are invalid
	+ Verify the credentials used by the FactoryTalk Linx OPC UA Connector match the settings in the OPC UA Server
	+ Switch the user authorization in both the UA Server and the FactoryTalk Linx OPC UA Connector to anonymous for testing
* Some possible security error codes:
	+ *Bad\_SecurityChecksFailed, Bad\_CertificateTimeInvalid,* *Bad\_CertificateIssuerTimeInvalid,* *Bad\_CertificateHostNameInvalid, Bad\_CertificateUriInvalid, Bad\_CertificateUseNotAllowed, Bad\_CertificateUntrusted, Bad\_CertificateRevocationUnknown, Bad\_CertificateIssuerRevocationUnknown, Bad\_CertificateRevoked, Bad\_CertificateIssuerRevoked*
* OPC UA Server exceeded connection capacity
	+ Verify the OPC UA Server supports the number of OPC Clients attempting to connect with it. Typical error codes:
		- *BadTooManySessions*

### Limited OPC UA server Bandwidth

The FactoryTalk Linx OPC UA Connector namespace upload process can fail or take an excessive length of time because the OPC UA server gets overloaded and is unable to respond quickly enough.

The FactoryTalk Linx OPC UA Connector uses data stored in the UA Server to set upper limits on the amount of data per-request. These are located in the UA Server’s address space below *Root\Objects\Server\ServerCapabilities*. These are optional and not all servers supply these – if a value is not supplied by the UA Server, the UAC assumes it can handle any load. Even when a constraint is supplied there is no guarantee that the values they contain will work as runtime conditions change.

A UA Client like *UAExpert* can be used to examine this area in the *UA Server’s* address space. The UAC Connector creates a text file that contains the latest values it read from the *UA Server*. This file is located at *C:\ProgramData\Rockwell\OpcUaConnector\RnaUaClientDataProvider\cache* and is named *<ServerName>.ServerInfo.txt*. This can be opened using Notepad.exe or any other text viewer.

If these values are not accurate the UAC may overload the server.

The FactoryTalk Linx OPC UA Connector provides predefined diagnostic tags that can be used to help diagnose problems caused by this.

* *@OPC\_NamespaceCacheUploadErrorsQty*: any value other than 0 indicates problems with uploading the address space.
* *@OPC\_NamespaceCacheUploadMetricTotalTimeMsg @OPC\_NamespaceCacheUploadServerReplyDelayTimeMsg:* If the *ServerReplyDelayTimeMsg* is much larger than the *MetricTotalTimeMsg*, this indicates that the server is taking a long time to respond to upload/browse requests. The UA Server may be overloaded.
* *@OPC\_NamespaceCacheUploadRequeuedContinuationPointQty:* any value other than 0 indicates the server lacks the necessary resources to respond to all the requests.

Upload problems are often caused because the server is overload and/or lacks resources (memory, processing power, etc.). Another problem we’ve run into is when the server sends too much data at one time, in which case the UAC receives the error: *BadEncodingLimitsExceeded*.

Two predefined tags can be used to reduce the load the UAC puts on the server to help workaround these problems.

* *@Diagnosic Items\Retained\@OPC\_OverrideMaxBrowseReplyDataPerNode*:
	+ Used to limit how much reply data the server will return to the UAC per-response.
	+ Start with a value = 100.
* *@Diagnosic Items\Retained\@OPC\_OverrideMaxNodePerBrowse*:
	+ Used to limit how many *Nodes* are queried at one time in the UA Server for their children. If the server is overloaded, and particularly when *@OPC\_NamespaceCacheUploadRequeuedContinuationPointQty* has a non-zero value, this may help limit the number of retries and reduce errors.
	+ Start with a value = 10.
		- The value should be equal to or less than the number of continuation points the server supports.

### Invalid *DataType* definitions in the OPC UA Server

In a few support cases, an OPC UA server contained *Variables* with *DataTypes* that were not compliant with the OPC UA specification. While the FactoryTalk Linx OPC UA Connector can handle many cases, it is difficult to anticipate the many ways vendors can diverge from the standard. In one such case the Siemens S7-1500 embedded controller contained an invalid *DataType* (null) for timers used pervasively throughout its data model ([See Knowledgebase 1135641](https://rockwellautomation.custhelp.com/app/answers/answer_view/a_id/1135641) ). In another case a Python based OPC Server’s data model diverged to the point where the FactoryTalk Linx OPC UA Connector was unable to communicate with the server and required the OPC UA Server to be re-built to address the issues.

Common problems include NULL *DataType* – this is where the *NodeId* for the *DataType* is 0. Another problem could be a custom *DataType* is used but the UA Server does not contain a definition for it.

If this is suspected, evaluating the OPC UA server with a sample client is an option (e.g. Unified Automation’s UA Expert). However, it should be noted that this can provide misleading results because the OPC UA sample client may have been enhanced to ignore the incorrect data model (this was the case with UA Expert and the two examples above). To diagnose this, the best option is run the FactoryTalk Linx OPC UA Connector’s namespace analysis function detailed in this document.

If the UAC detects an invalid *DataType* in a *Variable*, it will add a warning to the FTDiagnostics log like this:



### Large OPC UA Server namespace(s)

The FactoryTalk Linx OPC UA Connector uploads all tag and *DataType* definitions for all configured OPC UA Server endpoints and saves them in memory. Once the upload operation is complete, the FactoryTalk Linx OPC UA Connector builds an in-memory cache and writes it to the computers disk drive (*C:\ProgramData\Rockwell\OpcUaConnector\RnaUaClientDataProvider\cache*).

The FactoryTalk Linx OPC UA Connector is a 32-bit application, and can theoretically allocate up to 2GB of memory for operation (in practice it’s usually closer to 1.5GB). Through testing, it was determined the FactoryTalk Linx OPC UA Connector could handle around a total of 250,000 tag definitions for all OPC UA Server endpoints used by the UAC.

If the total number of tags exceeds this limit, failures up to and including crashes due to running out of memory are possible. There will be events and messages in the FTDiagnostics log that provide information that are related to running out of memory or other resources. Look for messages that are warnings or errors.

### Missing OPC UA Server tags

When the FactoryTalk Linx OPC UA Connector initially released in version 6.00, it was limited to scalar data items. Over several releases the FactoryTalk Linx OPC UA Connector was enhanced to handle additional forms of information, including structures and custom subtypes.

* V6.00 scalar numeric and string data
* v6.11 array access to numeric and string data
* v6.20 tags that are children of other tags
* v6.21 subtypes of standard *DataTypes*
* v6.30 Structures, alarms and option for other types to be converted to string
* V6.31 simple methods without arguments

The FactoryTalk Linx OPC UA Connector checks each tag’s *DataType* it uploads from an OPC UA server to determine if it can be supported. Items that cannot be supported will be excluded from the tags available in FactoryTalk Live Data. To help diagnose unsupported (missing) tags, FactoryTalk Linx OPC UA Connector provides predefined diagnostic tags and an analysis operation to provide information about the namespace for each OPC server it’s configured to access. See namespace analysis section detailed in this document.

### Some items converted to string tags

As mentioned previously, the FactoryTalk Linx OPC UA Connector checks each tag’s *DataType* it uploads from an OPC UA server to determine if it can be supported. Partial, read-only support is provided for some *DataTypes*. The values for these tags will be converted to strings so that clients can read and display the values, but writes are not supported.

This conversion is automatically performed – configuration is not needed.

Some of the OPC UA *DataTypes* that are converted to read-only strings include:

* NodeId
* ExpandedNodeId
* QualifiedName
* ByteString
* Guid
* DateTime, when converted to string
* Abstract *DataTypes*

An Abstract *DataType* acts as both a container of other *DataTypes*, a base type from which other *DataTypes* can be derived, and a wild card that matches to any of its derived types. For example, the *DataType Integer* contains the *DataTypes* for *Int8, Int16, Int32*, and *Int64*. If a Variable Node’s *DataType* attribute is set to *Integer*, then the value the UA Client receives may be any of those *DataTypes.*

## Namespace analysis

The FactoryTalk Linx OPC UA Connector can perform an analysis of an OPC UA Server’s namespace and generate a file listing the definitions if found, the resulting items that can be supported by the FactoryTalk Linx OPC UA Connector and indicate which definitions are not supported. One form of the namespace analysis is initiated either from the FactoryTalk Linx OPC UA Connector’s user interface (see the [Analyze Tag Names] button on the endpoint screen in the Tag information section). Alternatively, the analysis can be triggered by writing a value into a predefined diagnostic tag to generate different versions of the report.

*@OPC\_NamespaceAnalysisCmd*

* 0 - save all namespace items and folders (default, report generated from the user interface button starting with v6.31))
* 1 - save only namespace items with supported data types, no folders
* 2 - save only namespace items with unsupported data types, no folders (report generated from the user interface button prior to v6.31)
* 3 - save only folders
* 4 - save all namespace items with supported data types that contain scalar or array values.
* 5 – Save only methods

The *DataTypes* defined in the UA Server are appended to the end of all reports. This includes all structured types, both custom and OPC-defined, enumerations, integral and abstract types, and subtypes.

Tip: use the FactoryTalk Live Data Test client to trigger this operation by writing any of the above values to *@OPC\_NamespaceAnalysisCmd*.

 

Note there are additional predefined diagnostic tags that provide a wealth of information as the names is being processes and summary counts as the analysis is executing.

If the FactoryTalk Linx OPC UA Connector is configured to access alarms from the OPC UA server there are additional predefined tags to support them.

 

There are additional predefined tags for calling Methods in UA Servers (released in 6.31).

