GENESIS64
Connecting to Third Party OPC UA Servers

APPLICATION NOTE

June 2017

Description: Guide to connecting to a third party OPC UA Server with GENESIS64.
General Requirement: GENESIS64 and a third party OPC UA server; network connectivity to the OPC UA server if it is remote from GENESIS64.

Introduction

GENESIS64 is capable of connecting to a wide range of devices, servers, etc. You can also easily access OPC UA servers running on your machine, your local network, or even on the Internet. This application note lists all necessary details an engineer should know to connect to a local or a remote OPC UA Server.

The FrameWorX64 Server

Before trying to access data from an OPC UA Server, we need to introduce an important component in the GENESIS64 Suite called the ICONICS FrameWorX64 Server. This component acts as a server that provides data to GENESIS64 applications.

While the FrameWorX Server sends data to GENESIS64 components, it also acts as a client to other OPC servers. The relationship between different GENESIS64 components and the OPC servers is shown in Figure 1. In this case, the FrameWorX (FwxServer) Server acts as an aggregation data server that bridges between GENESIS64 applications and generic OPC UA Servers.

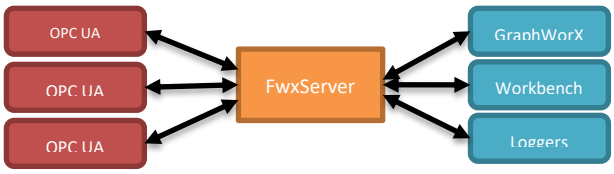

Figure 1 - GENESIS64 Applications Communicating to OPC Servers

As you can see, the FwxServer is a *client* for the OPC UA Servers on the left and a *server* for the GENESIS64 applications on the right.

The OPC UA Data Browser

In GENESIS64, there are various locations where you need a data source. It could be when you are creating a Process Point in a GraphWorX64 display, an alarm definition in AlarmWorX64 server, or a Hyper Historian tag. These are just a few examples, but in each of these cases, you will most likely browse for your data source using the GENESIS64 Data Browser.

The Data Browser contains several tabs above the address bar, each of which is used to access a different type of data source such as global aliases, local simulation tags, etc. In order to browse for any OPC data use the "OPC UA" tab.

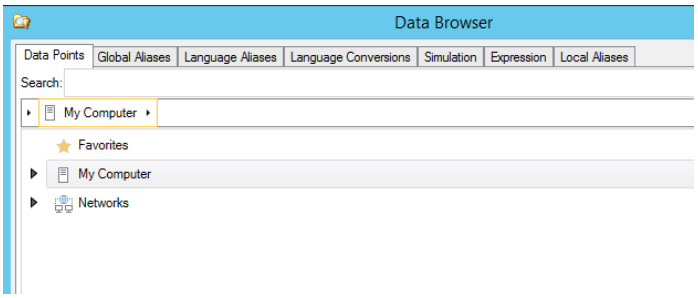

Figure 2 - The OPC UA Browser

Note that the FrameWorX64 Server is transparent to the user. All OPC/OPC UA servers you see in the OPC UA Browser are actually accessed through the FrameWorX64 Server.

UA Discovery

If a client wants to connect to an OPC UA server, it has to know the URL for that server's endpoint. Each OPC UA server exposes one or more endpoints the clients can use to connect. OPC UA servers register their endpoints' URLs at the Local Discovery Server (LDS) by OPC foundation on startup. LDS is a specialized server, which runs on a well-known port. UA clients can connect to the LDS and get a list of UA servers and their endpoints running on a computer. The LDS is similar to the OPCEnum module in Classic OPC.

In order to browse for a generic OPC UA server you need to start the LDS on the machine first, so that it can report all running OPC UA servers registered at the LDS. The LDS is installed with GENESIS64 by default as UA Local Discovery Server, and it is registered as an automatic service, so it should start automatically whenever the machine boots.

The FrameWorX64 server can browse the local network and connect to LDS on remote computers. Double-click on **OPC UA Servers** in the OPC UA Browser, then double-click **Network** to get list of computers in the network neighborhood.

When you expand a computer, the FwxServer connects to the LDS on that computer and a list of all running OPC UA servers appears. Expanding a UA server discovers the URLs of endpoints exposed by that server. If there is an endpoint that does not require security from the connecting OPC UA client, you will be able to connect to that endpoint and browse the UA server's address space. In case the server requires security, you can load it using certificates.

Certificates and Local Discovery Server

Technically, the Local Discovery Server (LDS) is just a specialized UA server developed by the OPC Foundation. Just like any other OPC UA server, the LDS requires a certificate to run.

When you start the LDS for the first time, it might not find its certificate because there is none yet, and will ask for permission to create a self-signed certificate. LDS cannot run without the certificate. Note that only users with administrative permissions with UAC off are allowed to create self-signed certificates.

To view a certificate, open a Microsoft Management Console (MMC) by going to **Start** typing "**mmc**" (without the quotes) into the search box. (If your operating system doesn't have a search box, go to **Start** → **Run** instead.) Once the Console opens, select the **File** menu and choose **Add/Remove Snap-in**.

In the Add or Remove Snap-in dialog, choose **Certificates** on the left-hand side and click **Add**. In the Certificates snap-in dialog, select **My user account** then click Finish. Add Certificates again and select **Computer account**, then **Local Computer**. You should now have two "Certificates" items in the "Selected snap-ins" box.

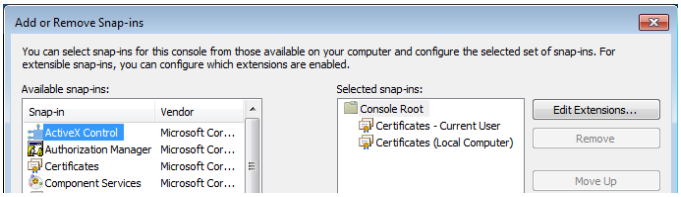

Figure 3 - Adding Certificates Snap-in

When you click OK to the Add or Remove Snap-ins dialog, it will return you to the MMC where you should see the two certificate nodes.

> **NOTE:** Should an OPC UA server need to access the LDS server, it needs to save the certificate at Local computer → UA Applications → Certificates as depicted in Figure 4 for ICONICS certificates.

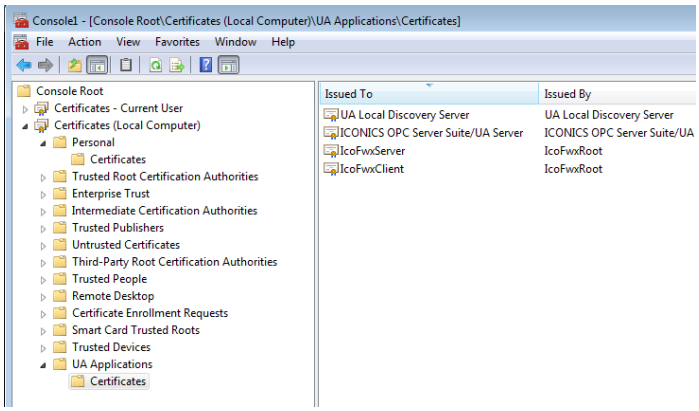

Figure 4 - MMC with Certificates Installed

The Client certificate (for accessing the ICONICS FrameWorX64 Server) and the Server certificate (which is required for starting the FwxServer itself and for accessing LDS) are stored in **Personal** store as well as **UA Applications** store as shown in Figure 4 - MMC with Certificates Installed Figure 4 and Figure 5.

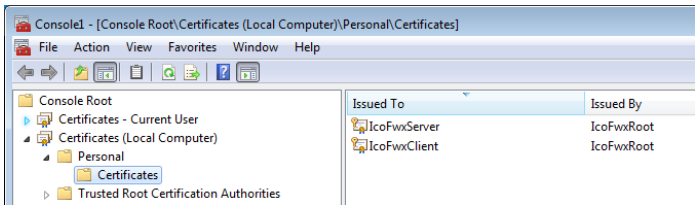

Figure 5 - MMC Showing Certificates Installed in Personal Store

Only those UA servers that register their certificates at the LDS are allowed to connect to LDS. This is a security feature designed by the OPC Foundation to protect the LDS against rogue UA servers. Certificates of ICONICS FrameWorX64 Server are copied to that store during installation.

To import the certificate from one computer to another, you must first export the certificate from the computer that already contains it. Browse to the certificate that should be exported, right-click on it and select **All Tasks** → **Export**. Then you can export it in different formats (DER X.509, Base-64 X.509, PKCS #7 or PKCS #12). By default, the DER format is selected.

> **NOTE:** It is not always necessary to export the private key with the certificate. However, if you are asked for the private key then you should export the certificate with the private key.

The exported certificate can be imported by right-clicking on the folder to which the certificate should be imported, and select **All Tasks** → **Import**. You can then browse for the exported certificate and place this certificate into the desired store.

Certificates and Third party UA servers

Each UA client has a certificate. In GENESIS64 it is the FwxServer that acts as a client to third party UA servers. Its certificates are created during installation.

Some third party UA servers may need the client's certificate in order to allow the client to connect. Then, it is necessary to provide the FwxServer's certificate to the third party UA server. The administrator of the UA server has to put the client certificate into the proper location to make sure the UA server can recognize the client.

The location of the certificate store for "allowed" client certificates depends on the third party UA server implementation. Usually, the third party UA servers use one of the following procedures to get the client certificate:

- The client administrator exports the certificate and the server administrator imports it to the proper location.
- The client application tries to connect to the server. During the connection process, client sends its certificate to the server. The server rejects the client connection and puts the certificate into a "rejected' store. The server administrator manually moves the certificate from the "rejected" store into "allowed" store.

In either case, a manual interaction of the server administrator is required.

UA Transport Protocols

OPC UA defines two transport protocols: HTTP and OPC.TCP.
UA servers and clients based on .NET SDK support both, while those based on ANSI C SDK support only OPC.TCP transport protocol.

The advantage of HTTP protocol is it may run on port 80 and thus go through firewalls. The OPC.TCP protocol is about 20% faster, but it requires corresponding ports to be open on firewalls.

Accessing Third Party OPC UA Servers over the Network

If the OPC UA Server you want to access is not directly reachable, or cannot be found using the Auto Discovery service, you can still browse it by typing its URL directly in the address bar of the data browser.

For example, in our network we have a UA Server installed on another machine with IP address 192.168.2.92 (Name: WIN-17J76EDMQSQ). The server exposes its services on port 48010 and uses opc.tcp as the communication protocol. To browse that server we just need to open the communication port in the firewall and type the full URL in the OPC UA Browser, as shown in Figure 6.

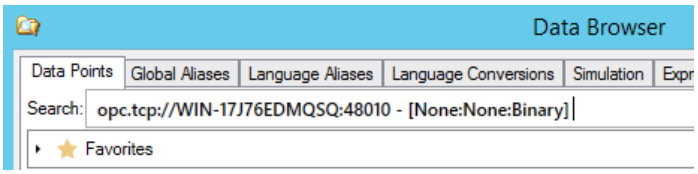

Figure 6 - Manually typed URL for an OPC UA Server

You can then press Enter to let the OPC UA Browser contact that OPC UA Server and present you its address space.

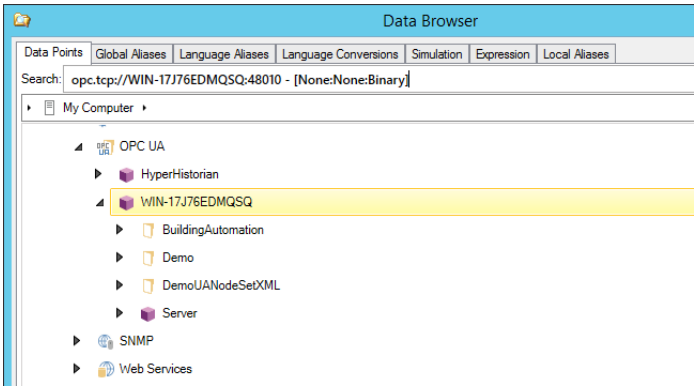

Figure 7 - Browsing an OCP UA Server on the Network

NOTE: [None:None:Binary] indicates a non-secured OPC UA connection.

From now on you can browse it the same way as you normally do with your local servers, simply double-clicking on the nodes.

Accessing Third Party OPC UA Servers over the Internet

If the OPC UA Server you have to access is exposed over the Internet, the procedure you need to follow is no different from accessing an OPC UA server within your network.

Suppose that the OPC UA Server you want to access is located at http://MyServerName.com

You cannot just type this URL into the UA Browser because this is a website address and usually does not host the OPC UA Server itself. There are two ways for you to access a server at that location:

- Use the OPC UA Discovery Server URL, which may have a path that looks like: http:// MyServerName.com:52601/UADiscovery
- Use The OPC UA Server direct URL, which may have a path that looks like: http:// MyServerName.com:51221/UA//ScadaServer

If you already have the second link available you don't need to pass through the Discovery Server; you can simply type this URL into the OPC UA Browser.

Troubleshooting

In this client/server scenario, especially when working over a network or over the Internet, it may be easy to overlook some settings, especially related to security. This may prevent successful communication between clients and servers.

Moreover, OPC UA does not disclose much information about the communication errors that might happen. This is done intentionally to prevent giving too much information to potential attackers who may be trying to access a server.

The following items present common items you may want to look at for troubleshooting in case your communication is not working as you expect.

Clock Synchronization

The OPC UA Server and client should have their clocks synchronized. This is necessary for UA to work. For more information, you can refer to the application note entitled *GENESIS64 – Synchronizing Machine Time*.

Certificates

Please refer to the sections "Certificates and Local Discovery Server" and "Certificates and Third party UA servers".

Firewalls

Please check that all GENESIS64 and OPC UA Server communication ports are open. For more details about the port numbers please see GENESIS64 and the OPC UA server documentation.

IP vs. Machine Name

Depending on your network configuration (e.g. DNS server) you might fail in contacting an OPC UA Server by referring to it using its machine name. For example, you may have problems using: opc.tcp://**MyServer**:4841 while you can successfully connect to: opc.tcp://**192.168.1.11**:4841. You should try this easy test in case you cannot access a server.