OPC-UA Installation and Integration Guide

This document is intended for installers of an Ayyeka OPC-UA (Open Platform Communications - United Architecture) Agent for the purpose of receiving data collected by the Wavelet devices directly to their SCADA system.

Contents

Scope of Operation
Prerequisites
Preliminary Step - Installing the OPC-UA service
Integrating Ayyeka Server with SCADA System
    Choosing the Integration Method
    Integrating Ayyeka Server Directly with SCADA OPC-UA
    Enable the Ayyeka Server's OPC-UA Agent
    Integrating Ayyeka Server with SCADA OPC-UA using OPC UA client
    Install and Configure an OPC-UA Client
Recommended Practices
Troubleshooting TLS
Appendix - Enhanced Security

Scope of Operation

The Agent supports OPC-UA DA (direct access) and HA (historical access). If the Client is configured for OPC-UA DA, the Cloud (or on-premises) server will send the last data sample from all requested streams. If you need to check historical information, you must specify the dates in your SCADA system.

Prerequisites

- Perform the following prerequisite actions:
  - Download the zip file containing the OPC_UA Agent .MSI Windows package installer file.
  - Install Microsoft .NET Framework 4.8.
  - Check that the directory C:Program Files (x86)Common FilesOPC FoundationUAv1.0Bin exists. If it does not exist, create it.
  - Copy the Opc.Ua.CertificateGenerator.exe file into the above directory (if the file does not already exist there).
  - Get the API Client Key and Secret by doing the following steps in the UI:

a. In the left pane, click **API**, and then click the **API Clients** tab.
b. In the API Clients window, click **+Generate API Key**.
c. In the Generate API Key window, select **REST**, type in a comment, and then click **Generate**.

Important: Record the API Client Key *and* the API Client Secret in a secure place because there is no way to access the API Client Secret in the future. You need them in step 4 below.

- The default secure protocol TLS v1.2 must be enabled for REST API clients and Microsoft Windows machines that host any of the CSV, DNP3, and OPC-UA agents. To check if TLS 1.2 is enabled, read this article. To enable TLS 1.2, see this article.

> ⚠ **TLS v1.2 must be manually installed on Windows Server older than 2019 - it is not installed by default.**
>
> **TLS v1.2 is automatically installed on Windows Server 2019 and newer.**

- You must have an Account/Organization Owner role or an Account/Organization Administrator role to generate the REST API keys. You must not generate the REST API keys when logged in as a user with the Partner role.

> For an on-premises system, you must not generate the REST API keys when logged in as the (super) Admin user. The keys generated by the Admin user will not work.

Preliminary Step - Installing the OPC-UA service

Install the OPC-UA Windows service in the same network in which the SCADA system is installed. Do the following steps:

1. Extract the supplied archive to a temporary folder (for example: C:tempAyyekaInstallation).
2. Double-click the **Ayyeka.Agents.OpcUa.msi** file (accessible from the downloaded zip file).
3. The Ayyeka OPC-UA setup wizard is launched. Follow the on-screen instructions. If you wish to specify an installation folder that is different from the folder (C:AyyekaAyyeka.Agents.OpcUa), select **Change** installation, and enter the destination folder.

> It is highly recommended to use the folder (C:AyyekaAyyeka.Agents.OpcUa).

4. Open the **Ayyeka.Agents.OpcUa.exe.config** file (located in the Agent installation folder) with a text editor, and in the **userSettings** section, specify the Key (Client ID) and Client Secret that was generated in #4 of Prerequisites.

```
<userSettings>
    <Ayyeka.Agents.OpcUa.Properties.Settings>
      <setting name="ClientId" serializeAs="String">
        <value>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</value>
      </setting>
      <setting name="ClientSecret" serializeAs="String">
<value>YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY=</value>
      </setting>
    </Ayyeka.Agents.OpcUa.Properties.Settings>
  </userSettings>
```

The default values for the Cloud are in the **Configuration for Cloud** below, so you typically do not need to change anything. If you are using an on-premises system, you will need to modify the configuration file according to the **Configuration for On-Premises** section below.

Configuration for Cloud:

```
<applicationSettings>
    <Ayyeka.Agents.OpcUa.Properties.Settings>
      <setting
name="Ayyeka_Agents_OpcUa_Rest_Authentication"
serializeAs="String">

<value>https://restapi.ayyeka.com/auth/token</value>
      </setting>
      <setting name="Ayyeka_OpcUa_Rest_API"
serializeAs="String">
        <value>https://restapi.ayyeka.com/v2.0/</value>
      </setting>
    </Ayyeka.Agents.OpcUa.Properties.Settings>
</applicationSettings>
```

Configuration for on-premises:

> If you configured your on-premises server for SSL communication internally, use HTTPS. Otherwise, use the default HTTP protocol as shown below.

```
<applicationSettings>
    <Ayyeka.Agents.OpcUa.Properties.Settings>
      <setting
name="Ayyeka_Agents_OpcUa_Rest_Authentication"
serializeAs="String">
        <value>http://<your_URL>:85/auth/token</value>
      </setting>
      <setting name="Ayyeka_OpcUa_Rest_API"
serializeAs="String">
        <value>http://<your_URL>:85/v2.0/</value>
      </setting>
    </Ayyeka.Agents.OpcUa.Properties.Settings>
</applicationSettings>
```

5. Open the **Log4net.config** file (located in the Agent installation folder) with a text editor, and make sure that the directory paths for the following files exist. If you need to change directory paths or values for any of the parameters, change them in this file:

    - Ayyeka**UaServer–all.log**
    - Ayyeka**UaServer–err.log**

```
<?xml version="1.0" encoding="utf-8"?>
<log4net>
  <appender name="GeneralLog"
type="log4net.Appender.RollingFileAppender">
    <file value="C:/temp/logs/AyyekaUaServer-
all.log"/>
    <threshold value="DEBUG"/>
    <appendToFile value="true"/>
    <maxSizeRollBackups value="10" />
    <maximumFileSize value="10MB" />
    <rollingStyle value="Size" />
    <datePattern value="_yyyy-MM-dd_HH" />
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date [%t] %-6p %-10c
%m%n"/>
    </layout>
  </appender>
  <appender name="ErrorLog"
```

```
type="log4net.Appender.RollingFileAppender">
    <file value="C:/temp/logs/AyyekaUaServer-
err.log"/>
    <threshold value="WARN"/>
    <appendToFile value="true"/>
    <maxSizeRollBackups value="10" />
    <maximumFileSize value="10MB" />
    <rollingStyle value="Size" />
    <datePattern value="_yyyy-MM-dd_HH" />
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date [%t] %-6p %-10c
%m%n"/>
    </layout>
  </appender>
  <root>
    <level value="ALL" />
    <appender-ref ref="GeneralLog"/>
    <appender-ref ref="ErrorLog"/>
  </root>
</log4net>
```

Integrating Ayyeka Server with SCADA System

When you use the OPC-UA communication protocol to communicate with your SCADA system, the Ayyeka server acts as an OPC-UA server, with the SCADA system securely pulling the data from the Ayyeka server. The data is pulled via HTTPS if the server is configured for SSL, otherwise via HTTP.

Choosing the Integration Method

If your SCADA system supports OPC-UA out of the box, follow the instructions in Integrating Ayyeka Server Directly with SCADA OPC-UA.

If your SCADA system does not support OPC-UA, you need to install and configure an OPC-UA client (such as a Kepware server) to act as a communication mediator between the Ayyeka server and your SCADA system. Follow the instructions in Integrating Ayyeka Server with SCADA OPC-UA using OPC UA client.

Integrating Ayyeka Server Directly with SCADA OPC-UA

If your SCADA system supports OPC-UA out of the box, do the following steps:

1. Enable the Ayyeka Server's OPC-UA Agent.
2. Exchange certificates and configure SCADA to go to the Ayyeka OPC-UA server endpoints.

Enable the Ayyeka Server's OPC-UA Agent

The Server Agent serves as the OPC-UA Server.

To enable the OPC-UA Agent, do the following steps:

1. On the machine hosting the Ayyeka server, start the Ayyeka.Agents.OpcUa service, and switch the **Startup Type** to **Automatic**.
2. Validate that Ayyeka.Agents.OpcUa is running by verifying that no errors appear in the log file **AyyekaUaServer–all.log**. The location of this log file is specified in the **Log4net.config** file (which is located in the Agent installation folder). The default location of the log file is `C:/temp/logs/`.
3. Copy the "Listener on (opc-tcp and http) " URLs from the log. These are the endpoints to which the OPC-UA clients are going to connect.

Integrating Ayyeka Server with SCADA OPC-UA by using OPC UA client

If your SCADA system does not support OPC-UA out of the box, but you wish to integrate the Ayyeka server with your SCADA system using OPC-UA, you need to:

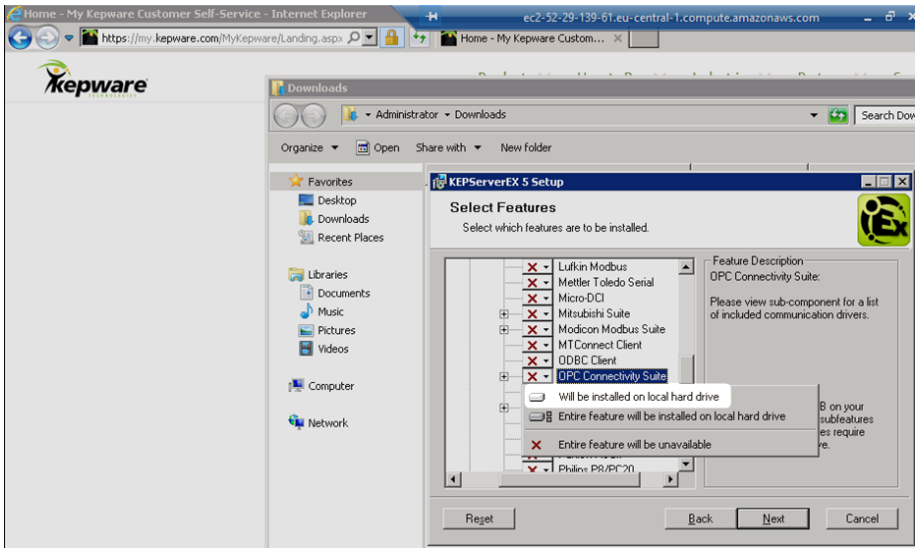
1. Enable the Ayyeka Server's OPC-UA Agent
2. Install and Configure an OPC-UA Client

Install and Configure an OPC-UA Client

This section provides instructions for installing and configuring an OPC-UA client to act as a communication mediator between the Ayyeka server and your SCADA system. There are various types of OPC-UA clients; this section provides instructions for the case where the OPC-UA client is a Kepware server.

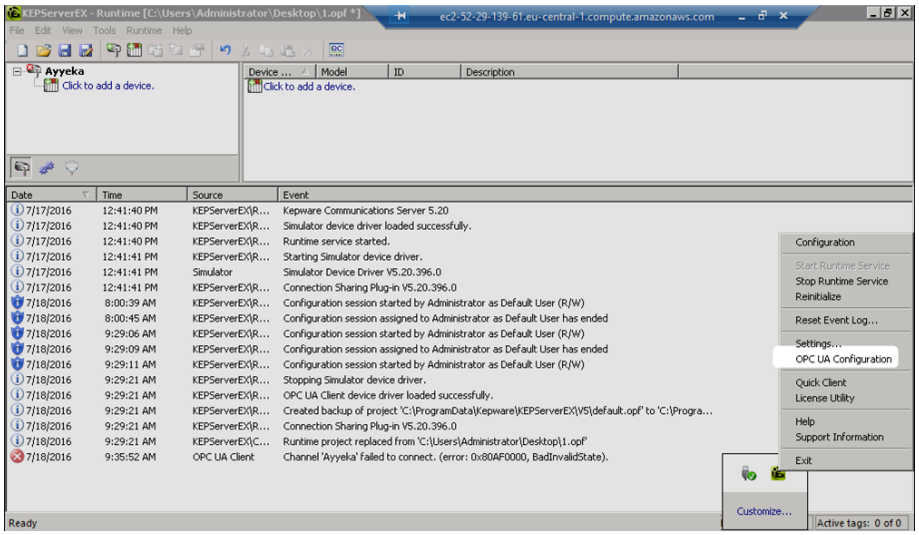
Modify the specific instructions to adapt them to the OPC-UA client of your choice.

1. Install Kepware with the OPC Connectivity Suite.

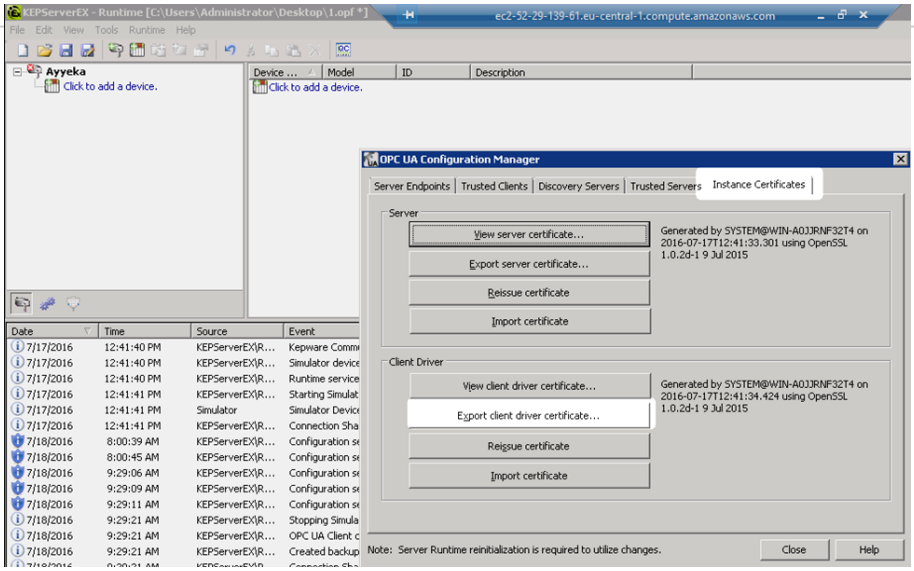


2. Add the KepwareServerEx certificate to the Ayyeka server's trusted certificate list, as follows:

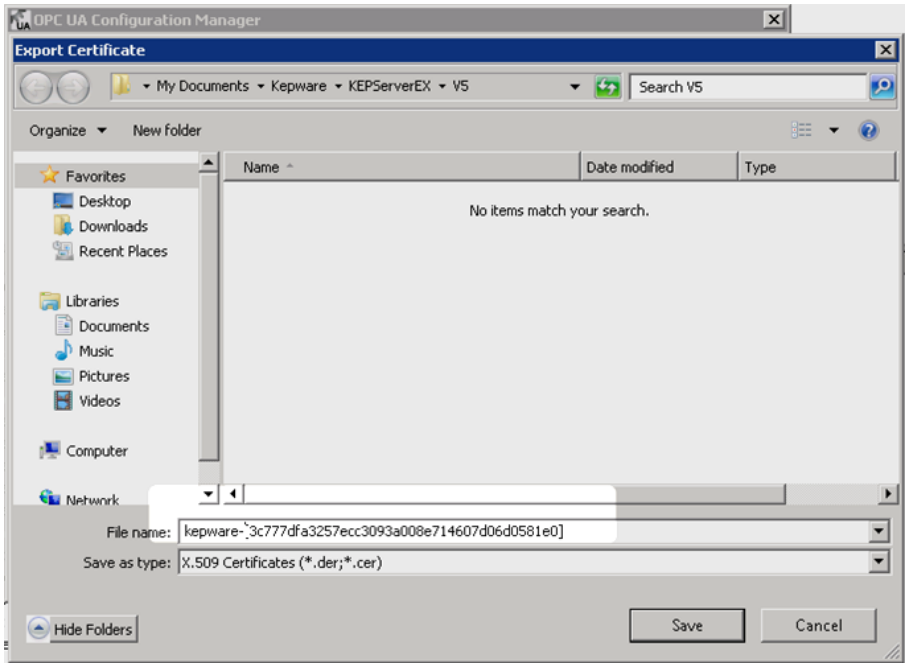
   a. Click **OPC UA Configuration**.



   b. Go to **Instance Certificates,** and then click **Export client driver certificate**.



c. Save the certificate.



d. Copy the certificate file to the following path on the Ayyeka Server:
`C:ProgramDataAyyekaCertificateStoresUA Applicationscerts`

3. Create a new channel for the Ayyeka OPC UA Server, as follows:

a. Select **Click to add a channel**. The New Channel wizard is launched.
b. In the Identification screen, name the **Channel name**. Click **Next**.
c. In the New Channel - Device Driver window, select the **Device driver** to be **OPC UA Client**. Click **Next**.
d. Do not change the Write Optimizations default settings. Click **Next**.

e. In the New Channel-UA Server window, define the Ayyeka OPC UA Server Endpoint URL, as follows:

Endpoint URL: opc.tcp://<host-name>:32160/AkOpcUaServer

Note: Make sure the host is reachable from the remote machine.

Security Policy: Basic256

Message Mode: Sign and Encrypt

f. If prompted, click **Yes** to trust the Ayyeka OPC UA Server certificate. Click **Next**.

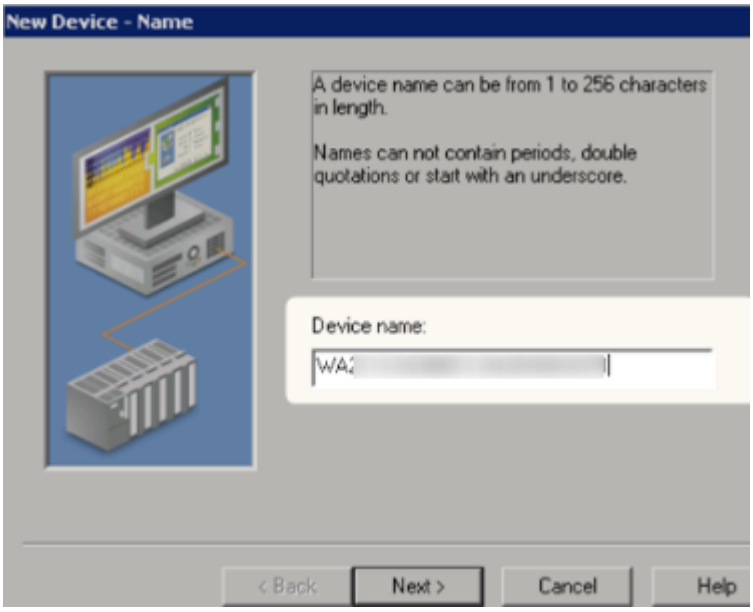
g. Do not change the Timeouts defaults settings. Click **Next**.

h. In the New Channel - Authentication window, enter the credentials provided by Ayyeka, of the pre-defined “User” user. Review the settings, and then click **Finish**.

> You can change these credentials using the Ayyeka Management UI, but in that case you must also change them accordingly in the OPC-UA agent config file (contact support@ayyeka.com for help).

i. Define a device for each Ayyeka Site:

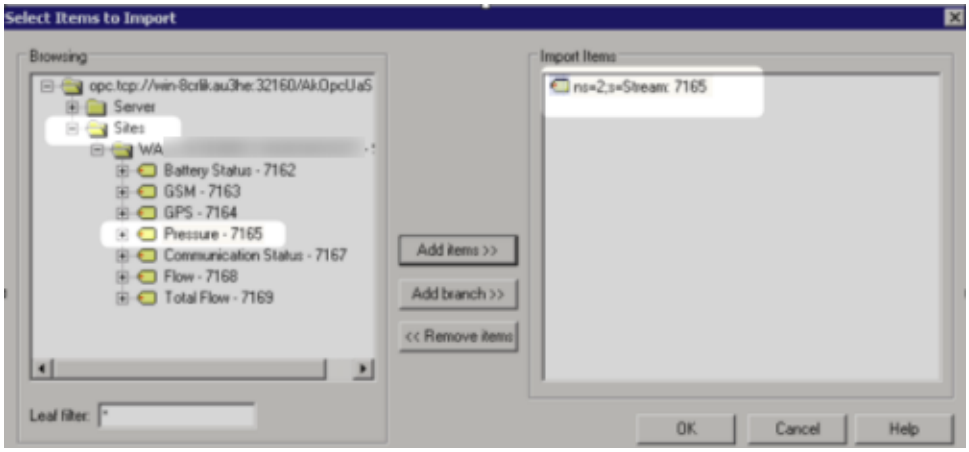
i. Define the **Device name** to be the Ayyeka site name.



> **Important:** Do not change the default settings in any wizard screens until the Import screen.

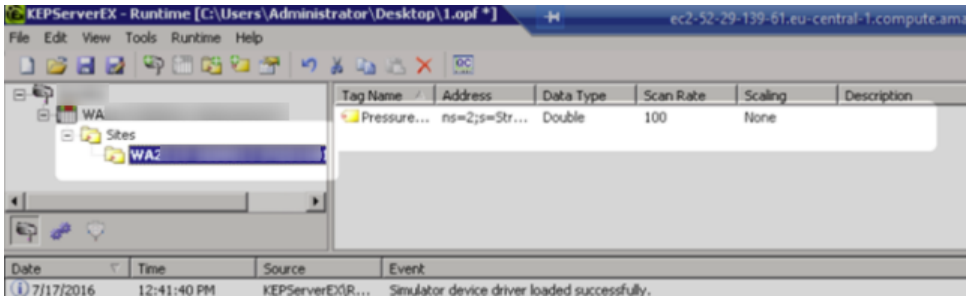
ii. In the Import screen, click **Select import items**.

iii. Select the Data Streams that you need to import.



iv. In the Summary screen, review the configuration, and then click **Finish**.

v. Verify that all the data streams appear in the Tags table.



Recommended Practices

There are two common use cases for OPC-UA HA:

- Send information between two timestamps: start time and end time
- Send information of the most recent time period as specified by the last number of minutes. For example, information from the last 30 minutes, every minute.

For the second use case, ensure that the OPCA-UA cyclical update timestamp is older than the device's last transmission, and the OPCA-UA update interval is shorter than the device's transmission interval.

Troubleshooting TLS

- Restarting TLS service gives error message about key container

  2021-09-09 18:11:08,691 [7] ERROR **Shutting down because of an error.**

Shutdown reason : The requested key container was not found.
System.Security.Cryptography.CryptographicException: The requested key container was not found.
at System.Security.Cryptography.CspKeyContainerInfo.get_UniqueKeyContainerName()
at Opc.Ua.CertificateFactory.Load(X509Certificate2 certificate, Boolean ensurePrivateKeyAccessible)
at Opc.Ua.CertificateIdentifier.Find(Boolean needPrivateKey)
at Opc.Ua.ServiceHost.InitializeSinglePolicy(Type contractType, ApplicationConfiguration configuration, BindingFactory bindingFactory, EndpointConfiguration endpointConfiguration, List`1 endpoints, MessageSecurityMode securityMode, String securityPolicyUri)
at Opc.Ua.ServerBase.CreateSinglePolicyServiceHost(IDictionary`2 hosts, ApplicationConfiguration configuration, BindingFactory bindingFactory, IList`1 baseAddresses, ApplicationDescription serverDescription, MessageSecurityMode securityMode, String securityPolicyUri, String basePath)
at Technosoftware.UaServer.Base.GenericServer.InitializeServiceHosts(ApplicationConfiguration configuration, BindingFactory bindingFactory, ApplicationDescription& serverDescription, EndpointDescriptionCollection& endpoints)
at Opc.Ua.ServerBase.Start(ApplicationConfiguration configuration)
at Technosoftware.UaServer.UaServer.Start(IUaServerPlugin uaServerPlugin, String configurationSection, String[] args)
The "key" referred to in this message does not refer to an API key, but rather to TLS.
If you need to restart the TLS service and then you receive this error message, it indicates a problem with your TLS configuration.

- Ensure that TLS is configured and enabled correctly.

- Review the list of common TL issues.

Appendix

Enhanced Security

In the **OpcServer.Config.xml** file, comment out the following XML properties:

<SecurityPolicies>

<!– <ServerSecurityPolicy>

<SecurityMode>**None_1**</SecurityMode>

<SecurityPolicyUri>http://opcfoundation.org/UA/SecurityPolicy#None</SecurityPolicyUri>

<SecurityLevel>0</SecurityLevel>

</ServerSecurityPolicy>

–>

<!-- <Allows anonymous users -->

<!– <ua:UserTokenPolicy>

<ua:TokenType>Anonymous_0</ua:TokenType>

</ua:UserTokenPolicy>

–>