OPC-UA and Device Connections

Note: Connecting to a device at this point is an optional step! You can come back to it later. However, the next steps will be more interesting if you add a device now.

Now that we’ve installed Ignition and have logged into the Configuration section of the Gateway web interface, let’s connect to a device. You can find detailed descriptions for many device connections in this User Manual, but they all include the following three main steps:

1. **Create new Device**
   From the left side configuration menu of the Gateway, you go to the **OPC-UA > Devices** section. Once at the Devices page, click on the Create new Device... link at the bottom of the table.
   A device is a named connection to an industrial device like a PLC. There are also Simulator devices in Ignition that you can connect to which act like a real device connection in case you don’t have one available. Device connections are done over the Ethernet for those devices that have an Ignition device driver.

2. **Choose a Driver**
   A list is displayed so you can choose the driver for the device you want. If you don’t have a device that matches one of the available drivers, you can add a Simulator device so you have some data to play with.

3. **Configure the Device**
   After you choose the driver, you give your device a name and set some options. Typically, the options are just an IP address to connect to. After configuring the device, click the Create New Device button to add your device. You can check the connectivity status of your device in the Gateway Status section, under Ignition OPC-UA Server.

Connecting to a Device

See the following links for the type of device connection you are interested in:

- Allen Bradley Ethernet
- Modbus
- Siemens
- UDP and TCP Driver
- Third Party OPC Servers
- Connecting to DNP3 Devices

What is OPC?

OPC is a specification for the transport and use of industrial data. It is published and maintained by the OPC Foundation, an organization comprised of hundreds of member companies that strives to ensure interoperability on the plant floor and beyond.

History

The original OPC specifications used Microsoft DCOM technology to provide a uniform way for industrial applications to share data. There were several separate specifications that provided functions such as Data Access (OPC-DA), Alarms and Events (A&E), Historical data (HDA) and more.

DCOM always proved difficult to work with, and by 2004 it was clear that a more modern solution was needed. Therefore, a new specification was developed that used common networking principals instead of DCOM, was platform independent, and combined the various separate specifications into one: OPC-UA.

**OPC-UA**

OPC-UA is the leading industrial standard for platform and vendor-neutral data access. Connecting any PLC device to Ignition is easy with OPC-UA. Device connections are done over the Ethernet for those devices that have an Ignition device driver. The OPC-UA Module in Ignition, makes Ignition act as an OPC-UA server, serving data collected by its built in drivers to other Ignition modules, as well as to third-party OPC-UA clients.

OPC-UA is the latest revision of the OPC specification, which offers platform and vendor neutral transfer and use of industrial data. The specification plays a crucial role in Ignition, and is the primary data access specification used in the Gateway. Ignition supports connections to any number of OPC-UA servers created by any manufacturer, provided that they are compliant to the specification. The data is then used to drive all aspects of the system. Creating connections to OPC-UA servers is described below in Connecting to OPC-UA.
Distributed Systems with OPC-UA

OPC-UA breaks down boundaries and enables free data flow. Using standard TCP/IP instead of legacy DCOM, OPC-UA makes it easy to securely transfer data between networks and through firewalls. All OPC-UA connections are based on the same technology, which means that a connection to your local machine is not entirely different than a connection to a machine that’s far away. This enables the creation of highly distributed system, and in combination with other features of Ignition can lead to much more connected enterprises.

For example, imagine a corporate network with an office in the center, and remote processes connected through a VPN, which would pass through a variety of connections. Each remote site could have an Ignition installation running only an OPC-UA module that would report data back to a central facility and record it in a database. The overall system cost would be very low, the data could be managed centrally in a single location, and then made available to all interested parties through the Vision module or any application that could access the database.

Servers and Clients

When discussing OPC (as the specifications are often called collectively), it is common to hear about OPC servers and OPC clients. An OPC Server is a piece of software that implements the OPC interface and provides data. An OPC Client is an application which connects to an OPC Server and uses the specification to retrieve and work with data.

The Ignition platform inherently offers OPC-UA client functionality. That is, even with no modules installed, the Gateway can connect to any compliant OPC-UA server and work with data. With the addition of the OPC-UA Module, Ignition becomes an OPC server as well, hosting device drivers that read and publish data.

The OPC-COM module is available to provide client access to older, DCOM based, OPC-DA servers.

Technology

The OPC-UA specification offers a wide range of flexibility in choosing technologies, from the transport mechanism, to the way data is encoded, to the encryption used to secure the data. Ignition supports the UA/TCP transport with the UA/Binary encoding scheme for maximum performance.

Additionally, Ignition supports all of the common encryption schemes. This means that Ignition connects to OPC-UA servers (and allows connections from clients) over TCP/IP, using encryption, and sends data by first encoding it into an efficient format defined by the OPC-UA specification. This is in contrast to other schemes outlined in the specification, which can use web services and XML encoding, and are not as efficient.

Connecting to a OPC Server

OPC-UA Server

An OPC-UA Connection is used to communicate with an OPC-UA compliant server, such as the one the OPC-UA module provides.

The following steps walk-through connecting Ignition (as an OPC-UA client) to a OPC-UA server:
1. On the Configure page of the Gateway, go to OPC Connections > Servers. The OPC Server Connections page is displayed.

2. Click on the Create new OPC Server Connection link.

3. Select OPC-UA Connection from the list and click Next. The Discover OPC-UA Endpoints page will appear.

4. Type in a OPC-UA endpoint for the OPC-UA Server Ignition should connect to. The format should be as follows: opc.tcp://IpAddress:Port. Ignition can connect to its own OPC-UA server's endpoint by using localhost and the default port of 4096: opc.tcp://localhost:4096. A hostname could be used instead of an IP address: opc.tcp://myhostname:4096.

5. Once an endpoint has been entered, click the Discover button. A list of available Security Policies and Message Security options will appear.

![Discover OPC-UA Endpoints](opc.tcp://localhost:4096)

6. Select a Security Policy and Message Security configuration to use when connecting to the endpoint. Basic128 Security Policies and SignAndEncrypt Message Security are more secure than None. If available, Basic256 should be selected.

7. Once an endpoint configuration has been selected, click the Next button. On the New OPC UA Connection Settings page, give the connection a name.

Some OPC-UA servers may require a Username and Password, but this is not always the case. Check with the OPC-UA server's documentation for more details.

By default, Ignition's OPC-UA server requires the following credentials:

<table>
<thead>
<tr>
<th>Username</th>
<th>opcuauser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>password</td>
</tr>
</tbody>
</table>

Once credentials have been entered, click the Create New OPC Server Connection button. Ignition is now connected to the OPC-UA server.

**OPC-UA Server properties**
<table>
<thead>
<tr>
<th><strong>Main</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A name used to identify this connection.</td>
</tr>
<tr>
<td>Description</td>
<td>Short description of this connection.</td>
</tr>
<tr>
<td>Read-only</td>
<td>Puts the connection into read-only mode. All writes sent to this server will fail.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Disable the connection to the OPC server.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Authentication</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Username and Password</td>
<td>If a username and password are specified, they are used as a user identity token when connecting to the specified OPC-UA server. The internal OPC-UA server provided by the OPC-UA module uses an Ignition security profile to govern who can connect to it. This can be configured in the <strong>OPC-UA &gt; Settings</strong> section.</td>
</tr>
<tr>
<td>Change Password?</td>
<td>Enable this property to change the configured password.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Advanced</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Override</td>
<td>When specified, if the endpoint address returned by the OPC server has a different IP address or hostname than the discovered endpoint, the overridden value will be used. Expects just an IP address or hostname. Example: 192.168.1.10</td>
</tr>
<tr>
<td>Max Per Operation</td>
<td>Specify the maximum number of nodes to read, write, subscribe, or unsubscribe to in any given UA server request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Failover Settings</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Failover Enabled</td>
<td>When enabled, if this OPC-UA server connection is faulted, requests will be sent to a failover OPC-UA server.</td>
</tr>
<tr>
<td>Failover Endpoint</td>
<td>The endpoint of the failover server. Example: opc.tcp://192.168.1.0:4096</td>
</tr>
<tr>
<td>Failover Threshold</td>
<td>The number of retry attempts before the failover connection is used.</td>
</tr>
</tbody>
</table>

This feature is new in Ignition version **7.8.3**
Click here to check out the other new features

<table>
<thead>
<tr>
<th><strong>Failover Settings</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Discovery URL</td>
<td>The discovery URL for the backup server's OPC-UA server. Expects the following format: opc.tcp://hostname:port</td>
</tr>
<tr>
<td>Backup Endpoint URL</td>
<td>The full endpoint URL for the backup gateway. An example endpoint would look like the following: opc.tcp://10.20.1.100:4096/iaopcua/Basic128Rsa15</td>
</tr>
</tbody>
</table>

*How do I determine the Endpoint for my backup?*

See step 5 in the Connecting to a **OPC-UA Server** guide above. Clicking the Discover button will show a list of available endpoints to connect to.

**Failover Versus Backup Properties**

The **Failover** properties should be used when a single Ignition gateway needs to connect to a pair of redundant OPC-UA servers. The failover OPC-UA server will be used in the event the primary OPC server goes down. To enable failover, set the **Failover Enabled** property to true, and specify the **Failover Endpoint**. The **Failover Threshold** can be adjusted if desired.

The **Backup** properties should be used when a pair of redundant Ignition gateways are trying to look at the same OPC-server. Both the **Backup Discovery URL** and **Backup Endpoint URL** properties need to be configured.
Connecting to OPC Classic (COM)

Classic OPC is based on COM, which is a technology in Microsoft Windows. Therefore, the information in this section only applies to Ignition Gateways installed on Windows. For other operating systems, OPC-UA must be used.

The OPC-COM module provides the ability to connect to OPC servers that only communicate using the older COM based OPC-DA standard. If you have an OPC server that is not capable of accepting OPC-UA connections and you need to talk to a PLC for which Ignition has no supported driver, you'll have to use the OPC-COM module to make your device data available in Ignition. Connections to OPC servers will be held open while the Ignition Gateway is running. All subscriptions to the server will use the same connection.

This section provides a brief walk-through of how to set-up a new Local or Remote OPC-DA server connection using the COM module. Due to the complications that Windows DCOM security settings can cause, this set-up guide is followed by the Troubleshooting OPC-COM Connections section that deals with an overview of how to deal with a faulted server connection due to DCOM security settings as well as other possibilities.

To Install OPC Core Components

   The OPC-COM module relies on a .dll package provided by the OPC Foundation (www.opcfoundation.org) called the OPC Core Components. You can download the OPC Core Components Redistributable from the OPC Foundation's website under the downloads section. Registration with the OPC Foundation is required before you can download the package, but the registration process is free and painless.

2. Download appropriate OPC Core Components Redistributable package.
   There are two packages to choose from, the 32-bit (x86) and the 64-bit (x64), make sure you get the correct one for the version of Java and Ignition you are running. 64-bit Java and Ignition needs the 64-bit Core Components package and likewise 32-bit installations needs the 32-bit package.

3. Install Core Components on Ignition server.
   It should be noted that if you are going to connect to an OPC server on a remote machine, you must also install the appropriate version of the Core Components on that server as well. The version type, 64-bit or 32-bit, does not need to be the same across the two servers. Just be sure to install the version that is appropriate for the OPC Server and Windows architecture.

4. (Remote) Install Core Components on remote machine running the OPC-DA server.
   Once you have the correct package downloaded you can extract the contents of the .zip file and then run the installer. With the core components installed you can now proceed to setting up your OPC-DA server connection in Ignition.

Connecting to OPC-DA Server

With the OPC Core Components now installed the next step is creating/configuring a new OPC-DA server connection.

To Install OPC-DA server connection

2. Go to OPC Connections > Servers and then select Create new OPC Server Connection.
3. Choose the OPC-DA COM Connection and then select whether you want to make a Local connection or if the OPC server resides on a Remote machine. For the most part, setting up a local or remote connection to an OPC-DA server is the same. There are only a couple of differences for a remote connection that will be highlighted along the way.
   - Local - Selecting a local connection takes you to a screen that contains a list of the available and running OPC servers located on the local machine.
   - Remote - For a remote connection you first have to specify the host name or IP address of the machine the OPC server resides on and then (as of Ignition 7.4) you are redirected to the available servers list.
4. Select the OPC server that you wish to connect to from the list. In the case where your server is not listed, see the OPC server is not listed... the Troubleshooting OPC-COM Connection section.

Unique Remote Connection Settings

Remote connections have a few unique settings that you can specify. You can get to these settings by selecting the Show advanced properties check box. As of Ignition 7.4 these should all be set for you (except for the CLSID which should no longer be necessary but is still available for you to set if you wish).

Remote Server
Specifies that the server is remote and that a DCOM connection will be used.

Host Machine
The computer name or IP address of the machine on which the remote server is running.

CLSID
This is no longer required as of Ignition 7.4, but it is still made available for you. It can be used in place of the ProgId because the ProgId is really just used to lookup the CLSID in the registry. This id can be found in the registry of the machine hosting the server under:
5. All of the settings for the server connection are rather straightforward and each property has a description of its functionality. Most of these settings should be fine when left at their default values. The only setting that could possibly give you some trouble is the ProgId. If you selected your OPC server from the list on the Choose OPC-DA Server page, this will be filled in for you. However, if for whatever reason your server wasn’t listed and you choose the Other Server option, you will have to know the ProgId for your server and specify it here. The ProgId is used to look up the CLSID of the OPC Server in the Windows Registry and without this a connection cannot be made.

6. When you are finished fine tuning these settings click Create New OPC Server Connection. You will be redirected to the OPC Server Connections page and your new server connection should be listed. The status of your connection will read Connected if Ignition was able to successfully connect to the third-party OPC server.

Connection is Faulted

In the case where your connection status is reporting Faulted, the troubleshooting process begins. As previously stated, configuring the DCOM settings on your machine can be a headache. The Troubleshooting OPC-COM Connections section next is an attempt to ease the process of determining why your connection is faulted and how to go about fixing the issue. If after exhausting the options presented to you, you are still having issues getting you server connection up, give our Inductive Automation tech support line a call and one of our representatives will be happy to assist you.

Troubleshooting OPC-COM Connections

This section provides you with a list of common OPC-COM connection problems with their possible solutions. It would be impossible to give an exhaustive list of everything that can go wrong but this should give you a good start on the troubleshooting process. If you do not see your problem listed and your connection status is faulted, try following the steps outlined in the Ignition Server DCOM Settings and OPC Server DCOM Settings sections.

Common Problems

OPC server is not listed in Choose OPC-DA Server list when first creating a connection

There are some cases in which an OPC Server that is installed will not show up in the generated list. This list is generated by the OPC Server Enumerator which is part of the OPC Core Components, so when a server you have installed on the machine does not appear in this list it is likely due to the OPC Core Components not being installed correctly.

Try reinstalling the Core Components and going through the process of creating a new server connection in Ignition again. If the server still does not appear and you have the ProgId (or the CLSID for a remote connection) for the OPC server, you can just select the Other Server option and then click Next. In this situation you will have to enter the ProgId manually on the New OPC-DA Server page.

With all the correct information about the OPC server we can sometimes still make a valid connection to the OPC Server even when it is not detected automatically. This however is rare. Most of the time when the server is not detected, any connection attempts Ignition makes will fail.

Connection status is Connected but data quality is bad or the connection goes Faulted after trying to read tag data

Usually this occurs when the DCOM settings for the machine on which Ignition is running are not correctly configured. DCOM connections go in both directions. Ignition must be able to send requests to the OPC server and the OPC server must also be able to callback to Ignition. If the DCOM settings on the Ignition server are not configured correctly those callbacks will fail and the server connection that initially had a status of “Connected” will either fail or all the tags that you have configured will come back with bad quality.

This is a problem that can affect both local and remote server connections.

Follow the steps outlined in the “Ignition Server DCOM Settings” section to ensure that you have correctly configured the DCOM security settings on the Ignition server machine.

Ignition launches second instance of an already running OPC server and is unable to see any data

It is important to note that Ignition runs as a service under the Windows System account. This can cause some issues with OPC servers that are meant to run interactively, meaning they run under the user account that is currently logged on. When Ignition attempts to make a connection to the OPC server, it will attempt to find an instance running under the same account and if it doesn’t find one it will launch its own instance under the System account. Even if there are other instances running, Ignition will choose the one that was launched under the System account for its connection.

Many OPC servers maintain an instance running under the interactive user account that has been configured by the user and maintains all of the device connection information. When Ignition launches a new instance, this configuration information is lacking and none of the desired data can be seen or accessed. To get around this problem, you must specify in the DCOM settings for the OPC server that it always identify itself with the interactive user. Essentially this will force Ignition to use the currently running instance of the OPC server.

To set the OPC server to run as Interactive User

1. The DCOM settings are found in the Component Services manager. Right-click the entry for your OPC server under the DCOM Config folder and select properties from the popup menu.
2. Select the Identity tab, select the option that reads The interactive user, and click OK.
3. Close out of component services and kill any extra instances of the OPC server you see running in the Task Manager.
4. Go edit and save the OPC server connection in the Ignition Gateway.

Faulted status with E_CLASSNOTREG error reported on OPC connections status page
This is almost always caused by the OPC Core Components not being installed correctly. Download and install the correct version(s) for your system(s) from the OPC Foundation (www.opcfoundation.org). Remember, if you are making a remote connection you must install these components on both the Ignition server as well as the machine on which the OPC server is running.

**DCOM Settings**

**Ignition Server DCOM Settings**

Follow these steps to open up the DCOM security settings on the machine that is running Ignition:

1. Open the Windows Component Services, located in the Administrative Tools section of the Control Panel.
2. Browse down through the Component Services tree until you see My Computer, right-click and select Properties.
3. We want to focus on the COM Security tab. There are two sections, Access Permissions and Launch and Activation Permissions. Each section has an Edit Limits... and Edit Defaults... button. You must add the ANONYMOUS and Everyone accounts under each of the four areas making sure that the Allow option is checked for each of the permission settings. If you skip adding both of these to either the limits or defaults areas under either of the two sections there is a good chance your connection will not be successful.
4. You can also try setting the Default Authentication Level to None and the Default Impersonation Level to Identify on the Default Properties tab. This isn’t always necessary but it can sometimes help.

**OPC Server DCOM Settings**

Follow these steps to open up the DCOM security settings on the machine that is running the OPC server:

1. Open Windows Component Services, located in the Administrative Tools section of the Control Panel.
2. Browse down through the Component Services tree until get to the DCOM Config folder.
3. Locate the entry for your OPC server that you wish to make a connection to, right-click and select properties.
4. Click the Security tab and you will see three sections, Access Permissions, Launch and Activation Permissions, and Configuration Permissions. There are two options to choose from for each section. If you already added the ANONYMOUS and Everyone accounts to the COM Security section from the Ignition Server DCOM Settings section then you can go ahead and just select the Use Default option for each of the three areas. The second option is to edit each of the groups that have Customize selected. You will have to add both the ANONYMOUS and Everyone accounts with all privileges.
5. Now select the Identity tab. You will notice that you can choose which account you want to run the OPC server under. Select the Interactive User option. This ensures that if Ignition launches an instance of the OPC server, it will run under whichever user is currently logged into the system.

**Creating an OPC-HDA Connection**

The process of connecting to an OPC-HDA server is similar to that of a DA server. Instead of going to the “OPC Connections” section, however, you define the server as a Tag History Provider.

1. Navigate to the Ignition gateway configuration section, as outlined above.
2. Under Tags, History, select “Create new Historical Tag Provider...”
3. Select “OPC-HDA Provider”
4. Follow the step outlined above, for DA connections.
5. Once complete, the status on the Tags>History screen will show the state of the connection. If Connected, you should now be able to browse and query the server through the Ignition designer.

**Example - Adding OPC-HDA data to a chart**

1. Open the designer, and create or open a project.
2. Create a window, and add an Easy Chart component.
3. Double-click on the chart, or right click and select Customizers>Easy Chart Customizer to bring up the chart customization window.
4. Next to the “Tag History Pens” table, select the first button, “Browse for tags”. This will display a tree for browsing all historical tags.
5. Browse through your defined HDA server. Once you find a tag, select “ok” to add it to the chart.
6. You may edit the tag to alter its aggregation mode, though the HDA provider will select a supported mode automatically if the specified mode does not exist in the server.
7. Once you save the configuration, the chart should update with the requested data.

A similar procedure can be used anywhere Tag History can be bound or used.

**OPC Quick Client**

You can access the OPC Quick Client from under the OPC Connections section of the Ignition Gateway Config section. It allows for quick, simple testing of any devices connected to the server.

You can browse by expanding tree nodes and read/write to tags by clicking on the [r] and [w] buttons next to those tags.

Subscriptions can be made by clicking on the [s] button. Clicking on the enable live values link will automatically refresh subscriptions and show live value changes (if there are any).
**Ignition OPC-UA Server**

You can see the following OPC-UA Server Settings by going to the Configure section of the Gateway and then choosing **OPC-UA > Settings**.

<table>
<thead>
<tr>
<th>Authentication</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Profile</td>
<td>The User Source that the OPC-UA module will use to authenticate incoming connections against. By default, this is set to the opcua-module User Source. This profile is included in the default installation and has the following as its default settings: user: opcuauser password: password</td>
</tr>
<tr>
<td>Allowed Roles</td>
<td>Roles within the given User Source that are allowed to connect to the server. Multiple roles should be separated by a comma, for example, Administrator,user,manager.</td>
</tr>
<tr>
<td>AllowAnonymous Access</td>
<td>If checked, will allow anonymous connections to the server. Not checked by default.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Port</td>
<td>The port the OPC-UA module runs on.</td>
</tr>
<tr>
<td>Endpoint Address</td>
<td>This is the local address that the Ignition UA server will bind to. It is also the address that will be used in a GetEndpointResponse, so it is important that this be an address reachable by any clients that wish to connect. Requires a module restart to take effect. This is useful if the server machine has a VPN connection or multiple adapters and is returning the wrong address.</td>
</tr>
<tr>
<td>Minimum Sampling Interval</td>
<td>The fastest rate (in milliseconds) that the server will use to sample its underlying data sources. Requires a module restart to take effect. The default is 100.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discovery Server</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery Server Enabled</td>
<td>Select the check box if the local discovery server should be enabled. Requires a module restart to take effect. By default, the check box is selected or True.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expose Configured Tags</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Expose Tag Providers</td>
<td>If enabled, Ignition tag providers will be exposed through the OPC-UA server, allowing third party clients to access the tags configured in the system. By default, the check box is not selected or False.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Stale Threshold</td>
<td>The multiplier by which the server determines that updates from a driver have become stale. This period will be calculated as the fastest sampling rate for that node multiplied by this settings value. Default is 5.</td>
</tr>
<tr>
<td>Allow Untrusted Certificates</td>
<td>Whether or not to automatically accept incoming certificates or to rely upon them being placed into the trusted certificate list before allowing a connection. By default, the check box is selected or True.</td>
</tr>
<tr>
<td>Discovery Server</td>
<td>Address of the discovery server to register with. Default is localhost.</td>
</tr>
<tr>
<td>Discovery Registration Frequency</td>
<td>Frequency (in minutes) at which the server will attempt to register with the discovery server. Default is 10.</td>
</tr>
<tr>
<td>Auto-cast Writes</td>
<td>Attempt to auto-cast incoming writes to the correct DataType before rejecting them with a Bad_InvalidType StatusCode. By default, the check box is selected or True.</td>
</tr>
</tbody>
</table>

**Creating a New Device**

To add a new Device go to the Devices section of the OPC-UA module configuration in the Ignition Gateway, go to the Configure section of the Gateway and then to **OPC-UA > Devices**. Once at the Devices page, click on the Create new Device... link at the bottom of the table. You will be taken to a page where you can select the driver to use. Choose your driver and click **Next**.

The General settings common to all devices are as follows:

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Name | The user-defined name for this Device. The name chosen will show up in OPC Item Paths and under OPC-UA Server > Devices of the Status page of the Gateway. The Device Name must be alphanumeric.
---|---
Enable Device | Only devices that are enabled appear in OPC-UA Server > Devices of the Status page of the Gateway and thus have their tags available for use.

### Connectivity

The settings under **Connectivity** are device-specific settings and unique to each driver.

**Timeout**

The timeout settings refer to the communication between the device driver and the OPC-UA server and usually can be left at their default values.

### Verifying Device Connectivity

Device connectivity can be verified in the following three places:

- In the **Status** page of the Gateway under OPC-UA Server > Devices
- In the **Status** page of the Gateway under Overview > Device Connections box
- In the **Status** page of the Gateway under Overview > OPC Connections box

### Drivers

**Allen Bradley Drivers**

**ControlLogix 5500**

You will see the following **Connectivity** settings on the New Device page accessed from the Configure section of the Gateway under OPC-UA > Devices.

| Hostname | This is the IP Address of the ControlLogix Ethernet module (1756-ENET) to route through to connect a ControlLogix processor. EthernetIP protocol on TCP port 44818 (0xAF12) is used to communicate to ControlLogix processors.
|---|---
| Timeout | After sending a request to the ControlLogix processor, the Communication Timeout setting is the amount of time in msec to wait for a response before treating it as a failure.
| Slot Number | The Slot Number value is the zero based ControlLogix chassis slot number of the ControlLogix processor to connect to.
| Connection Path | The Connection Path value is used to define the route of the PLC-5 processor to connect to. Currently routing through the ControlLogix Ethernet Communication Interface Module (1756-ENET) to the ControlLogix Data Highway Plus-Remote I/O Communication Interface Module (1756-DHRIO) and on to a PLC-5 processor of the DH+ network is supported.

The Connection Path format contains 4 numbers separated by commas. The first number is always 1 and tells the 1756-ENET module to route through the backplane. The second number is the slot number of the 1756-DHRIO module of the DH+ network the PLC-5 processor is connected to. The third number is the channel of the 1756-DHRIO module that the PLC-5 processor is connected to. Use 2 for channel A and 3 for channel B. The final and fourth number is the DH+ node number. This number is in octal and is the same as configured in the PLC-5 processor. See the ControlLogix Ethernet Communication interface Module User Manual for more information.

Connection Path Format: 1,1756-DHRIO slot number,1756-DHRIO channel,DH+ node number

The valid range for the 1756-DHRIO slot number is between 0 and 16 but depends on the chassis size. The 1756-DHRIO channel is either 2 for channel A or 3 for channel B. The DH+ node number range is from 00 to 77 octal.

For a more in depth explanation of connection paths, see [Allen Bradley Connection Paths Explained](#).

### Supported Connection Methods

ControlLogix 5500 connected through 1756-ENET/A or 1756-ENET/B.
**MicroLogix 1100/1400**

**MicroLogix 1100/1400 Connectivity Settings**

<table>
<thead>
<tr>
<th>Hostname</th>
<th>The Hostname value is the IP Address of the MicroLogix 1100 processor, MicroLogix 1400 processor or 1761-NET-ENI Ethernet interface. EthernetIP protocol on TCP port 44818 (0xAF12) is used to communicate to the listed devices.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Timeout</td>
<td>After sending a request to the MicroLogix processor, the Communication Timeout setting is the amount of time in msec to wait for a response before treating it as a failure.</td>
</tr>
<tr>
<td>Browse Cache Timeout</td>
<td>When the data table layout is read from the MicroLogix processor, the Browse Cache Timeout value is the amount of time in msec to cache the results.</td>
</tr>
</tbody>
</table>

**Supported MicroLogix Connection Methods**

MicroLogix 1100 and 1400 direct
MicroLogix 1100 and 1400 connected through 1761-NET-ENI
MicroLogix 1100/1400 connected through Spectrum Controls WebPort 500

Note: MicroLogix 1200 and 1500 are not fully supported. Browsing is not available on these devices.

**PLC-5**

**PLC-5 Connectivity Setting**

<table>
<thead>
<tr>
<th>Hostname</th>
<th>The Hostname value is the IP Address of the PLC-5 processor. The protocol that the PLC-5 processor supports is automatically detected. It will use either CSP protocol on port 2222 (0x8AE) or EthernetIP protocol on port 44818 (0xAF12).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Timeout</td>
<td>After sending a request to the PLC-5 processor, the Communication Timeout setting is the amount of time in milliseconds to wait for a response before treating it as a failure.</td>
</tr>
<tr>
<td>Browse Cache Timeout</td>
<td>When the data table layout is read from the PLC-5 processor, the Browse Cache Timeout value is the amount of time in milliseconds to cache the results.</td>
</tr>
<tr>
<td>Connection Path</td>
<td>The Connection Path value is used to define the route of the PLC-5 processor to connect to. Currently routing through the ControlLogix Ethernet Communication Interface Module (1756-ENET) to the ControlLogix Data Highway Plus-Remote I/O Communication Interface Module (1756-DHRIO) and on to a PLC-5 processor of the DH+ network is supported.</td>
</tr>
</tbody>
</table>

**More Information On Connection Path**

The Connection Path format contains 4 numbers separated by commas. The first number is always 1 and tells the 1756-ENET module to route through the backplane. The second number is the slot number of the 1756-DHRIO module of the DH+ network the PLC-5 processor is connected to. The third number is the channel of the 1756-DHRIO module that the PLC-5 processor is connected to. Use 2 for channel A and 3 for channel B. The final and fourth number is the DH+ node number. This number is in octal and is the same as configured in the PLC-5 processor. See the ControlLogix Ethernet Communication interface Module User Manual for more information.

Connection Path Format: 1,<1756-DHRIO slot number>,<1756-DHRIO channel>,<DH+ node number>

The valid range for the 1756-DHRIO slot number is between 0 and 16 but depends on the chassis size. The 1756-DHRIO channel is either 2 for channel A or 3 for channel B. The DH+ node number range is from 00 to 77 octal.

For a more in depth explanation of connection paths please read: Allen Bradley Connection Paths Explained.

**Supported PLC-5 Connection Methods**
PLC-5 L/20E, L/40E, L/80E direct
All PLC-5 processors connected through DH+ via the 1756-DHRIO module.

**SLC 505**

### SLC Connectivity Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>The Hostname value is the IP Address of the SLC processor. The protocol that the SLC processor supports is automatically detected. It will use either CSP protocol on port 2222 (0x8AE) or EthernetIP protocol on port 44818 (0xAF12).</td>
</tr>
<tr>
<td>Communication Timeout</td>
<td>After sending a request to the SLC processor, the Communication Timeout setting is the amount of time in milliseconds to wait for a response before treating it as a failure.</td>
</tr>
<tr>
<td>Browse Cache Timeout</td>
<td>When the data table layout is read from the SLC processor, the Browse Cache Timeout value is the amount of time in milliseconds to cache the results.</td>
</tr>
<tr>
<td>Connection Path</td>
<td>The Connection Path value is used to define the route of the SLC processor to connect to. Currently routing through the ControlLogix Ethernet Communication Interface Module (1756-ENET) to the ControlLogix Data Highway Plus-Remote I/O Communication Interface Module (1756-DHRIO) and on to a SLC processor of the DH+ network is supported.</td>
</tr>
</tbody>
</table>

**More Information On Connection Path**

The Connection Path format contains 4 numbers separated by commas. The first number is always 1 and tells the 1756-ENET module to route through the backplane. The second number is the slot number of the 1756-DHRIO module of the DH+ network the SLC processor is connected to. The third number is the channel of the 1756-DHRIO module that the SLC processor is connected to. Use 2 for channel A and 3 for channel B. The final and fourth number is the DH+ node number. This number is in octal and is the same as configured in the SLC processor. See the ControlLogix Ethernet Communication Interface Module User Manual for more information.

Connection Path Format: 1,<1756-DHRIO slot number>,<1756-DHRIO channel>,<DH+ node number>

The valid range for the 1756-DHRIO slot number is between 0 and 16 but depends on the chassis size. The 1756-DHRIO channel is either 2 for channel A or 3 for channel B. The DH+ node number range is from 00 to 77 octal.

For a more in depth explanation of connection paths please read: [Allen Bradley Connection Paths Explained](#).

**Supported SLC Connection Methods**

- SLC505 direct
- SLC505, SLC504, SLC503 connected through 1761-NET-ENI
- SLC504 connected through 1756-DHRIO
- SLC505, SLC504, SLC503 connected through Spectrum Controls WebPort 500

**Allen Bradley Connection Paths Explained**

Connections to ControlLogix, CompactLogix, PLC-5, MicroLogix and SLC Allen-Bradley processors through a ControlLogix Gateway require a connection path. The connection path is unique to your setup and is dependent on what modules the connection is being routed through. With there being nearly an endless number of ways to route your connection from device to device it is impossible to give an example of every possible connection path, but in general there is a pattern to how the connection path is specified.

**Follow the Path**

A connection path is exactly what it sounds like. It is a path that when followed will lead a processor residing in a numbered slot of a chassis somewhere on site. You merely have to follow the path and build the connection path as you go. The first connection point between Ignition and the device is a ControlLogix Ethernet module such as an ENET, ENBT or EN2T module. The slot number of this module doesn't matter and there is no need to specify it in the connection path. The first entry in any connection path will be a 1, which specifies moving to the back plane. You then specify the slot of the module you wish to move to, followed by the port or channel of that module that you wish to exit through. Finally you specify the address of your entry point to the next module and the process starts all over again. This process may sound complicated at first but after some practice it will get easier.

**Steps**

1. Move to the backplane.
2. Specify the slot number of the module you are moving to.
3. Specify the exit port or channel.
4. Specify address of entry point (DH+ Station Number / ControlNet Address / IP Address of ethernet module).
5. Move to the backplane.
6. Specify processor slot number OR the slot number of the module you wish to exit through.

**Connection Path Entries for Different Module Types**

...
How you specify your exit point from a module is slightly different depending on which module type you are using. You can only move in two directions once you are “in” a module: out to the back plane, or out through the module port/channel. Ethernet modules have ethernet ports and an IP address; ControlNet modules have ControlNet Ports and ControlNet addresses; DHRIO modules have channels and station numbers. Below is a list of different kinds of modules and what numbers you specify in the connection path when you are exiting or entering those modules. When in a module, an entry of 1 will always take you to the backplane.

**ENET, ENBT, and EN2T:**
- **Exiting**
  1 = Backplane
  2 = Ethernet Port
- **Entering**
  IP Address

**CNB:**
- **Exiting**
  1 = Backplane
  2 = ControlNet Port
- **Entering**
  ControlNet Address

**DHRIO**
- **Exiting**
  1 = Backplane
  2 = DH+ Channel A
  3 = DH+ Channel B
- **Entering**
  DH+ Station Number (an octal value between 0-77)

You use these numbers to specify how to move out of the module, then you specify where you are moving to by either specifying the DH+ station number, ControlNet address, or the IP address of another ethernet module. Your connection path will always be an even number of entries due to the fact that you always move in two steps: out of a module and then in to another module. So if your connection path ends up with an odd number of entries you have missed a step somewhere and you’ll have to go back and trace the path again.

Some examples have been included to help illustrate the process of tracing a connection path. The first three examples illustrate how to build your connection path when going from one ControlLogix Gateway to another. The last example shows connecting through a ControlLogix Gateway to 3 different SLC 5/04 devices via DH+.

### Creating an OPC-HDA Connection

The process of connecting to an OPC-HDA server is similar to that of a DA server. Instead of going to the “OPC Connections” section, however, you define the server as a Tag History Provider.

1. Navigate to the Ignition gateway configuration section, as outlined above.

2. Under Tags, History, select "Create new Historical Tag Provider..."

3. Select "OPC-HDA Provider"

4. Follow the step outlined above, for DA connections.
Once complete, the status on the Tags>History screen will show the state of the connection. If Connected, you should now be able to browse and query the server through the Ignition designer.

Example - Adding OPC-HDA data to a chart

1. Open the designer, and create or open a project.

2. Create a window, and add an Easy Chart component.

3. Double-click on the chart, or right click and select Customizers>Easy Chart Customizer to bring up the chart customization window.

4. Next to the "Tag History Pens" table, select the first button, "Browse for tags". This will display a tree for browsing all historical tags.

5. Browse through your defined HDA server. Once you find a tag, select "ok" to add it to the chart.

6. You may edit the tag to alter its aggregation mode, though the HDA provider will select a supported mode automatically if the specified mode does not exist in the server.

7. Once you save the configuration, the chart should update with the requested data.

A similar procedure can be used anywhere Tag History can be bound or used.

A Frequently Asked Question

How to I get data from my PLC?

Getting data from your PLC into Ignition is a two step process:

1. Add a device, see Connecting to a Device.
2. Add some tags, see Creating Tags.
It requires you to touch both the Ignition Gateway and the Ignition Designer. There are also some limitations as to what kind of devices you can connect to Ignition and these are explained throughout the user manual, however included below is an overview of what you can expect when it comes to compatibility.

**Brief summary of device connection in Ignition**

- Ignition can only connect directly to devices over Ethernet.
- Ignition can only connect directly to devices for which there is an Ignition device driver. Included drivers are:
  - Allen Bradley - ControlLogix 5500, CompactLogix, MicroLogix 11/1400, PLC-5, SLC 505
  - Siemens - S7-300, S7-400, S7-1200
  - Modbus - The Modbus driver connects to any ethernet enabled device that uses the Modbus protocol
- Ignition can connect to third party OPC servers via OPC-UA or OPC-DA (using the OPC-COM module) for devices that do not have a supported driver.

**Adding a Device to Ignition**

**Ignition Supported OPC-UA Device**

Most commonly you will be adding a device that is supported by one of the built-in device drivers. The first step is connecting your device to Ignition. This is done through the Ignition Gateway Configuration section under the OPC-UA -> Devices page.

1. Click "Add a device..."
2. Select the driver for the device you wish to add
3. When adding a device you will notice that there are some common settings that are shared by all devices. You can find an explanation of these settings here: Adding a New Device
4. Specify any of the required device specific settings for the device (For example, hostname, etc.)
5. Check the status of your device to see if it is connected.

As long as all the device information you entered was correct you should see your device in a "Connected" state. The only exception to this is if you chose to add a Siemens or Modbus device. Since these devices don't support the browsing of tags you will have to create and address some tags in the Ignition designer before the device will stop cycling from a connected to disconnected state.

If you need to address your tags for your Siemens or Modbus device you'll want to read about adding Tags in the Ignition Designer as well as how addressing works for the different protocols. You will have to first add a tag in the Ignition designer and then edit the OPC Item Path of the tag using the appropriate addressing scheme.

**Adding Connection to 3rd Party OPC Server via OPC-UA**

If your device does not have an Ignition driver, you can use a 3rd party OPC server to connect to your device and then have Ignition connect to the server as a client. If the OPC server talks OPC-UA, you can add a new OPC-UA server connection in the Ignition Gateway. Configuration will be different depending on what OPC server you are using but the following is an example of a popular solution, connecting to KEPServer via OPC-UA, see Connecting to Kepware OPC-UA.

**Adding Connection to 3rd Party OPC-Server via OPC-COM**

The following section provides a detailed walk-through on how to connect to an OPC server using the OPC-COM module. If Ignition doesn't have a driver for your device and you don't have an OPC server that talks OPC-UA, you have to connect using the OPC-COM module, see Connecting to OPC Classic (COM).

**Adding tags for Allen Bradley devices**

Tags are how Ignition represents your PLC tags. You create Tags in the Ignition Designer and then you can use these tags to store history or display PLC data in your projects.

For the most part, Allen Bradley devices support browsing of tags in the PLC. There are a few exceptions like the MicroLogix 1200/1500 for which you have to manually address your tags. For now we will focus on creating tags from devices that support browsing.

1. Open the Ignition Designer.
2. Drag desired tags from the OPC Browser to the Tag Browser as described here: Creating Tags (if you don't know what Tag Provider means don't worry, merely drag them into the Tags folder). You should now see some tags in the Tag Browser that show the current values of the respective tags in your PLC. Don't stop here. You should read through the related links below so you can learn more about Tags and how they work.

**Simulator Drivers**

**Generic Simulator**

The generic simulator provides a variety of tags that offer different data types and value generation styles. For example, there are ramps, sine waves, and random values. Additionally, there is a set of static writable tags whose values will persist while the device is running.

There are no configurable settings for the generic simulator.

**Simulator tags**

ReadOnly - static values that do not change for read only purpose.
ReadOnlyBoolean1 - false
ReadOnlyBoolean2 - true
ReadOnlyShort1 - 1
ReadOnlyShort2 - 2
ReadOnlyInteger1 - 1
ReadOnlyInteger2 - 2
ReadOnlyLong1 - 1
ReadOnlyLong2 - 2
ReadOnlyFloat1 - 1.1
ReadOnlyFloat2 - 1.2
ReadOnlyDouble1 - 1.1
ReadOnlyDouble2 - 1.2
ReadOnlyString1 - "ABCDEFG"
ReadOnlyString2 - "ZYXWVUT"

Writable - static values that you can read/write to, initial values below.

WritableBoolean1 - false
WritableBoolean2 - false
WritableShort1 - 0
WritableShort2 - 0
WritableInteger1 - 0
WritableInteger2 - 0
WritableLong1 - 0
WritableLong2 - 0
WritableFloat1 - 0
WritableFloat2 - 0
WritableDouble1 - 0
WritableDouble2 - 0
WritableString1 - "" (empty string)
WritableString2 - "" (empty string)

Random - Random values updating at some rate, they follow Java Random(rate) - rate is the seed.

RandomBoolean1 - 10 sec
RandomShort1 - 5 sec
RandomInteger1 - 1 sec
RandomLong1 - 2 sec
RandomFloat1 - 10 sec
RandomDouble1 - 10 sec

Sine - Different sine waves with frequency, amplitude and offset (listed in that order)

Sine1 - 0.1, 100.0, 0.0
Sine2 - 0.01, 50.0, -25.0
Sine3 - 0.02, 10.0, 10.0
Sine4 - 0.04, 100.0, 0.0
Sine5 - 0.08, 100.0, 0.0

Ramp - Ramp signals starting from 0 going up to some value at the specified rate. When they reach their upper limit, they are reset to zero.

Ramp1 - 0 - 100 @ 10 ms
Ramp2 - 25 - 75 @ 100 ms
Ramp3 - 0 - 100 @ 50 ms
Ramp4 - 0 - 100 @ 25 ms
Ramp5 - 0 - 100 @ 12.5 ms

Realistic - Values determined by adding a random number (between -1 and 1) to the current value.

Realistic1 - -50 - 50 @ 500 ms
Realistic2 - -50 - 50 @ 1000 ms
Realistic3 - -50 - 50 @ 1500 ms
Realistic4 - -50 - 50 @ 2000 ms
Realistic5 - -50 - 50 @ 2500 ms
Allen Bradley SLC Simulator

The SLC simulator driver creates a simple device whose address structure mimics a basic SLC structure. There are currently no configurable parameters.

In this section ...