

# Managing files with MQTT Transmission and MQTT Engine

## Prerequisites:

- Knowledge of Ignition and Module installation process: [Cirrus Link Module Installation](#)
- Installation of the following Cirrus Link MQTT modules selecting the [version compatible](#) with your Ignition installation.  
Modules can be downloaded from the [Ignition Strategic Partner Modules](#) download page:
  - MQTT Distributor
  - MQTT Engine
  - MQTT Transmission

## Overview:

Transmission is an MQTT module for Ignition that can publish files using MQTT Sparkplug to be processed by MQTT engine.

Files are transferred using Sparkplug over MQTT and which files get transferred and when they get transferred can be configured in a number of different ways. The file transfer configuration also defines how [quickly a file transfer occurs](#) and how long before a timeout will occur on failure to receive host side acknowledgements which can be adjusted as required based on the size of files you are sending and any bandwidth constraints that you may have.

This tutorial shows how to configure the MQTT modules to publish files using both the manual and auto publish methods.

- [Install the MQTT modules](#)
- [Configure Primary Host on MQTT Engine and MQTT Transmission](#)
- [Config MQTT Engine File Handling](#)
- [Configure MQTT Transmission to publish files](#)
- [Manual Publish of Files](#)
- [Auto Publish of Files](#)
- [Viewing the messages in the Ignition logs](#)
- [Extra Activities](#)
  - [Adjust the file transfer configuration to define how quickly a file transfer occurs and how long before a timeout will occur on failure to receive host side acknowledgements based on the size of files you are sending and any bandwidth constraints that you may have.](#)
  - [Setup two Ignition systems - a host system installed with MQTT Engine and MQTT Distributor and an edge system installed with MQTT Transmission](#)

## Install the MQTT modules

Install the three MQTT modules listed in the pre-requisites onto your Ignition system following the [Cirrus Link Module Installation](#) guide.

By default, both MQTT Engine and MQTT Transmission are configured to connect to MQTT Distributor on [tcp://localhost:1883](#) and will show as Connected under their respective Servers configuration setting in the Ignition UI. MQTT Transmission will also have an Example Transmitter configured pointing to a set of tags that are configured in the Ignition "default" tag provider.



Review the [MQTT Transmission Transmitters and Tag Trees](#) tutorial for additional information on how Transmitter configurations interact with Ignition tag trees to create the Sparkplug IDs required.

This allows the three modules to automatically connect and provide a starting base for the tutorial.

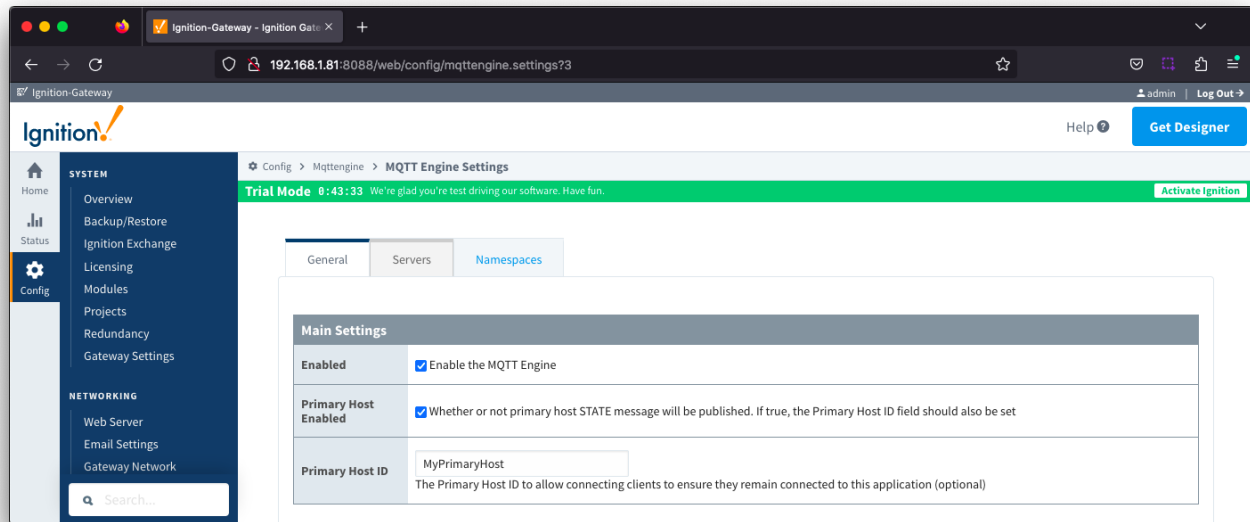
## Configure Primary Host on MQTT Engine and MQTT Transmission



Primary Host ID must be configured for File Transfer to work

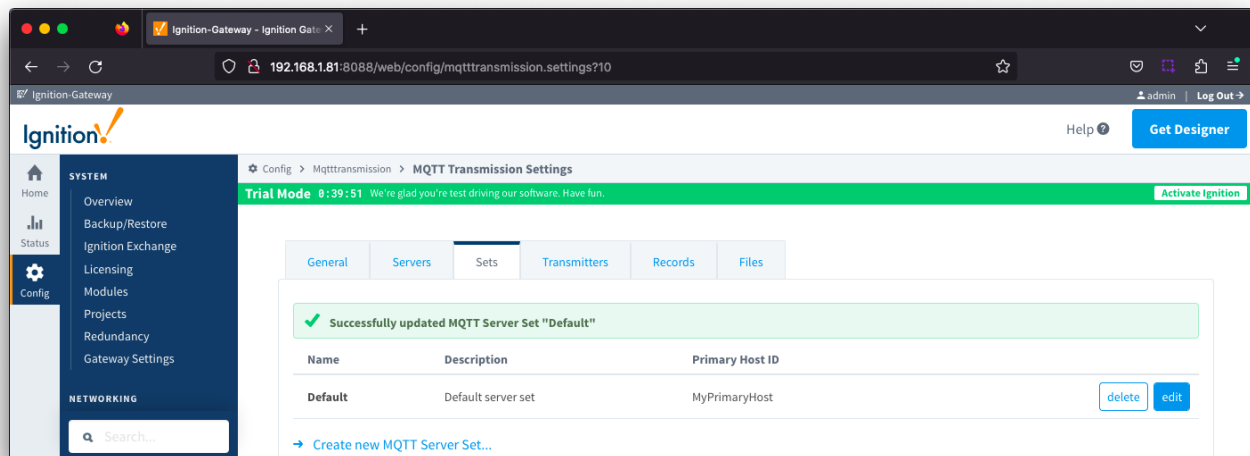
Navigate to the MQTT Engine > Settings in the left side of the Ignition Gateway UI and select the General tab.

Set Primary Host Enabled and add a Primary Host ID as shown below:



Navigate to the MQTT Transmission > Settings in the side of the Ignition Gateway UI and select the Sets tab.

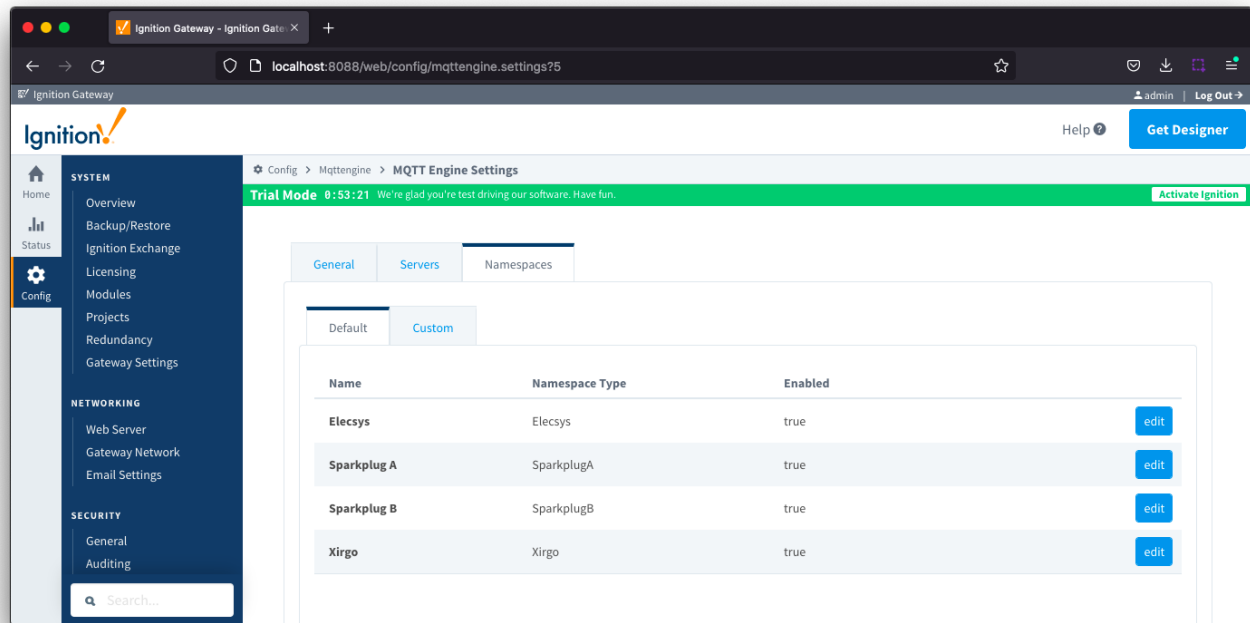
Edit the Default Set and add a Primary Host ID as shown below:



## Config MQTT Engine File Handling

Now we can configure the MQTT Engine module to process the published files.

Navigate to the MQTT Engine > Settings in the left side of the Ignition Gateway UI and select the Namespaces tab. From the Default tab, edit the Sparkplug B namespace shown below:



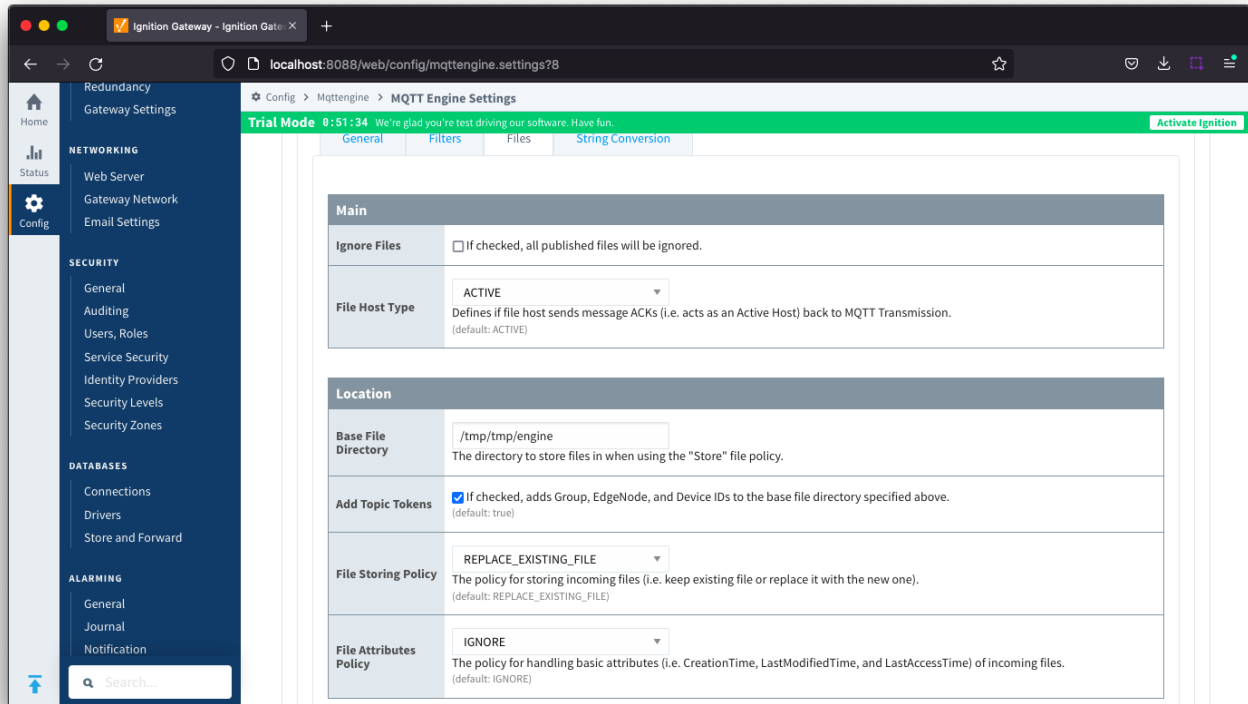
Now select the 'Files' tab and set the following

- Ignore Files
  - Set this property to unchecked so that publish files will not be ignored
- File Host Type:
  - ACTIVE to specify that a message ACK is sent back to MQTT Transmission
- Base File Directory
  - Set to some directory file path (i.e. 'C:\My Files' or something similar) where the files will be stored
- Add Topic Tokens
  - Leave this property as checked to add the Group ID, Edge Node ID and Device ID to the base file directory path
- File Storing Policy
  - Set this policy to REPLACE\_EXISTING\_FILE
- File Attributes Policy
  - Leave this property at the default value of IGNORE



Review the [MQTT Engine Configuration](#) guide for information on all the File properties


When complete, it should look similar to the following. Save the changes after confirming.

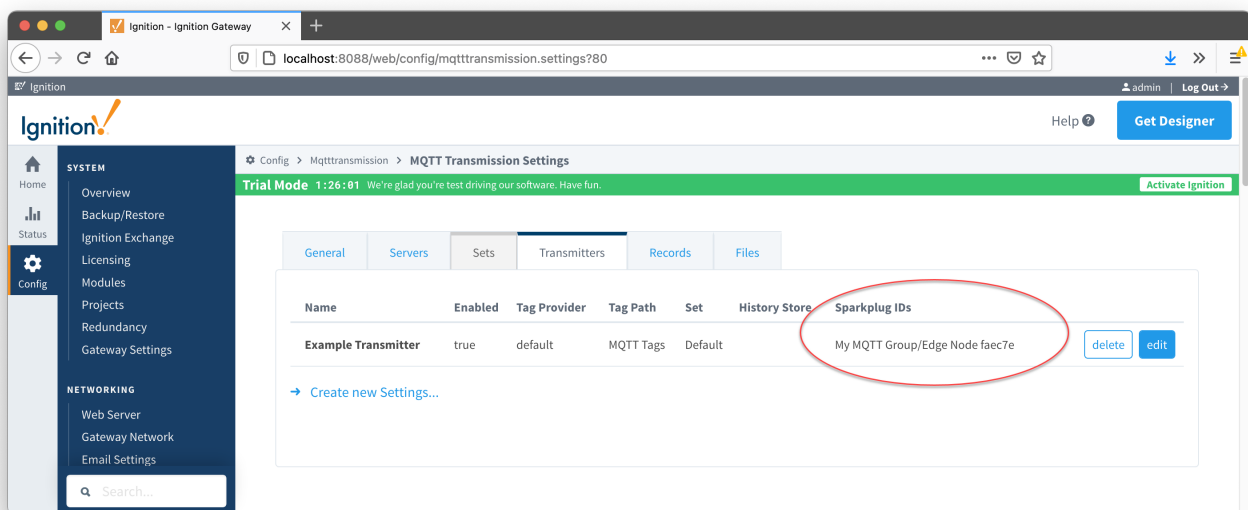


## Configure MQTT Transmission to publish files

Now we can configure the MQTT Transmission module to publish files using a configured Transmitter. Navigate to the MQTT Transmission > Settings in the left side bar of the Ignition Gateway UI and select the Transmitters tab.

Make note of the Sparkplug IDs configured for your transmitter. For our example we have the Group ID as *My MQTT Group* and the Edge Node ID as *Edge Node faec7e*

 If you have an existing Transmitter configured where the Sparkplug IDs are created dynamically from the Ignition tag path, see [MQTT Transmission Transmitters and Tag Trees](#), you will need to identify the Edge Node Descriptor being used and include the GROUP\_ID and EDGE\_NODE\_ID in the Files configuration.



Now on the 'Files' tab, select the Create new Files... link and configure the following fields

- Name
  - By default a unique name will be set for this File Record. You can edit this if you choose but it must be unique.
- Tag Provider
  - This is the tag provider where the publish control and information tags will be created. In our example, set to 'default'.
- Tag Folder Path
  - This is the tag folder in the specified tag provider where the publish control and information tags will be created.
  - In our examples, set to 'files\_manual' if configuring for manual publish or 'files\_auto' if configuring for auto publish
- Enable Auto-Publish
  - This will enable auto\_publish.
  - In our examples, leave unchecked if configuring for manual publish and check if configuring for auto publish.
- Group ID
  - Use the Group ID from your Transmitters Sparkplug ID. For this example, My MQTT Group
- Edge Node ID:
  - Use the Edge Node ID from your Transmitters Sparkplug ID. For this example, Edge Node faec7e



Review the [MQTT Transmission Configuration](#) guide for information on all the File properties

When complete, it should look similar to the following. Save the changes after confirming.

Ignition Gateway - Ignition Gate X

localhost:8088/web/config/mqtttransmission.settings?9

Config > Mqtttransmission > MQTT Transmission Settings

Trial Mode 1:21:41 [Activate Ignition](#)

**Tag Settings**

Name	FT_d163269a-2f2 A unique name for the Files Record
Tag Provider	default The Name of the tag provider
Tag Folder Path	files_manual The path to the Tag folder

**File Settings**

Enable Auto-Publish	<input type="checkbox"/> Enable auto-publishing
File Scan Rate	60 The rate to scan the 'Files' directory specified in the 'Publish Files Path' tag for files to publish (default: 60)
File Scan Rate Time Unit	SECONDS Time Unit for the 'File Scan Rate' configuration parameter (default: SECONDS)

**Sparkplug Settings**

Group ID	My MQTT Group Group ID
Edge Node ID	Edge Node faec7e Edge Node ID
Device ID	 Device ID

**Modules**

- Projects
- Redundancy
- Gateway Settings

**Config**

- Web Server
- Gateway Network
- Email Settings

**SECURITY**

- General
- Auditing
- Users, Roles
- Service Security
- Identity Providers
- Security Levels
- Security Zones

**DATABASES**

- Connections
- Drivers
- Store and Forward

**ALARMING**

- General
- Journal
- Notification
- On-Call Rosters
- Schedules

**TAGS**

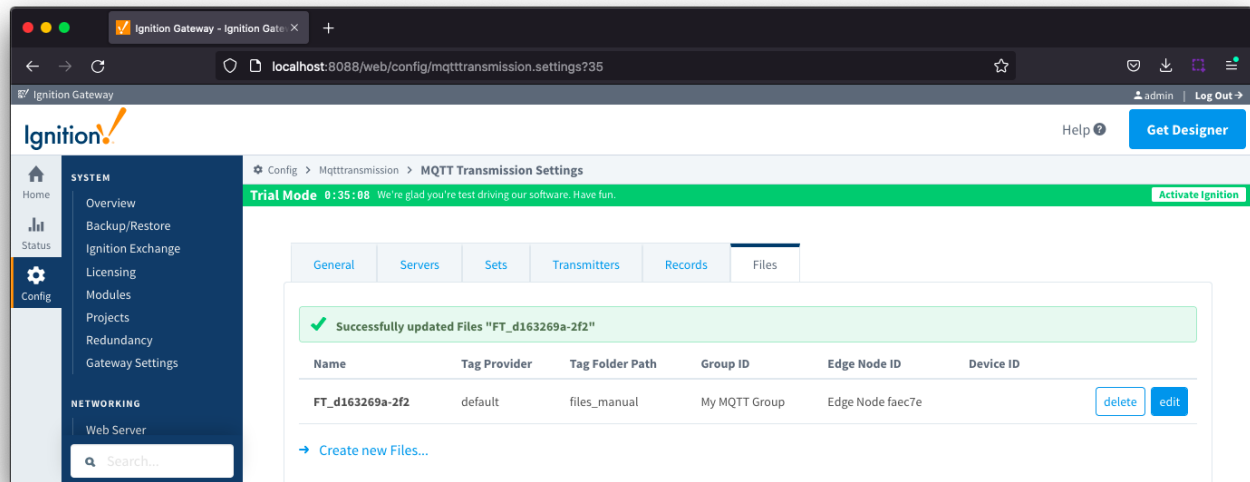
- History
- Realtime

**OPC CLIENT**

- OPC Connections
- OPC Quick Client

Search...

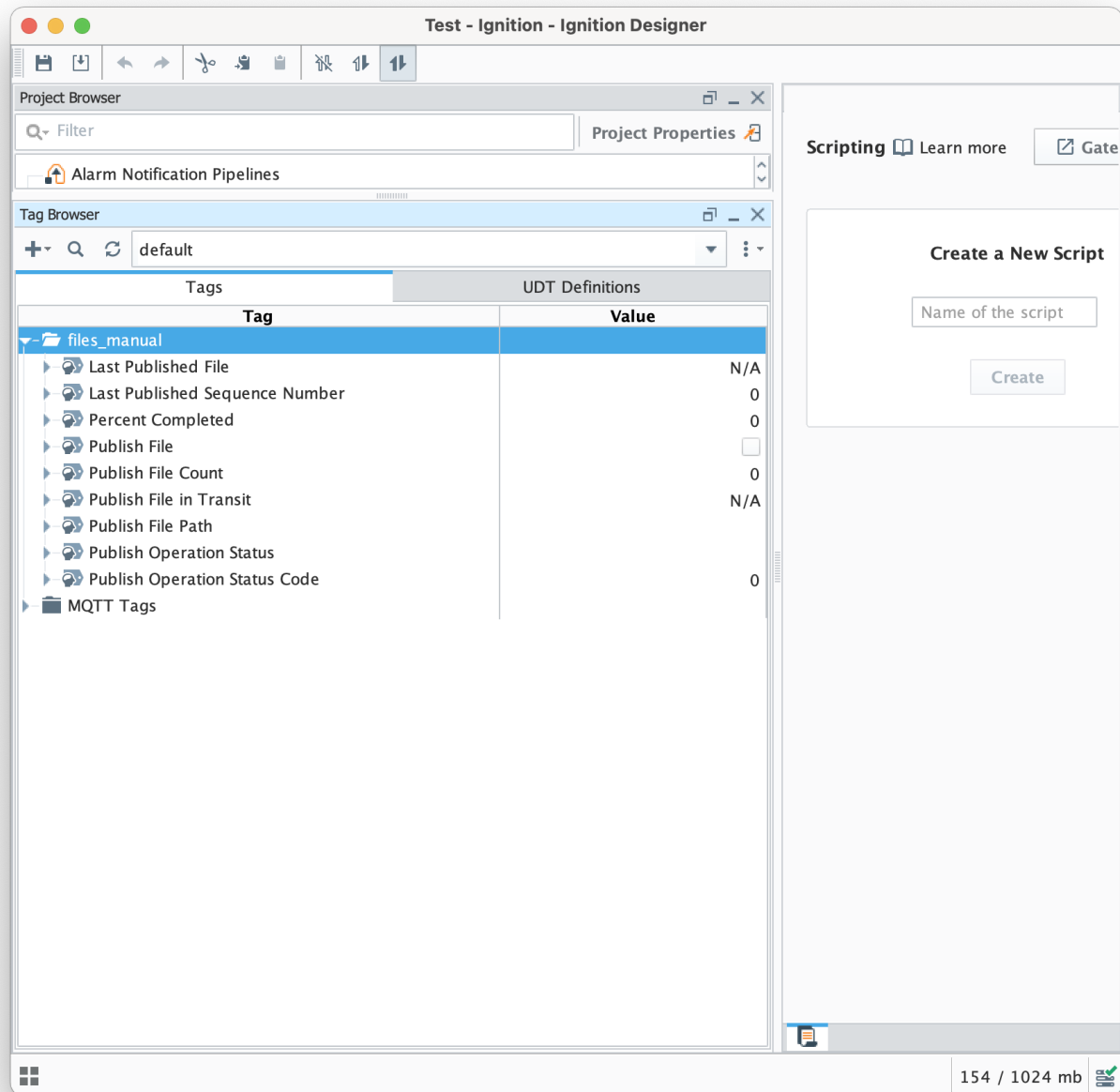
After the configuration is saved, it should look similar to below:



## Manual Publish of Files

With a File record created following the steps in [Configure MQTT Transmission to publish files](#) with the 'Enable Auto-Publish' unchecked and the Tag Folder Path set to "files\_manual", everything should be configured to send files from MQTT Transmission to MQTT Engine.

Open Ignition Designer on the system running MQTT Transmission. You should see the control and information tags created in the specified tag provider and folder as shown below:



The control and information tags created in the folder are:

Name	Type	Description
Last Published File	String	Name of last published file
Last Published Sequence Number	Integer	Sequence number of last published file since last reset of metrics
Percent Completed	Byte	Publish completion percent for file being published
Publish File	Boolean	Manual trigger to publish file
Publish File Count	Long	Number of files published since last reset of metrics
Publish File in Transit	String	Name of current file being published
Publish Files Path	String	Full path to the target file to publish over MQTT
Publish Operation Status	String	Status description of current publish operation



Publish Operation Status Code	Integer	Status code for current publish operation
Reset	Boolean	Trigger to reset publish metrics

At this point, we just need to tell MQTT Transmission which file to send. Download and unzip this sample file to some location and note that location.

This [sample\\_file.zip](#) contains two text files which are:

- `sample_file.txt`
  - This is a text file with an arbitrary string. But, it could be a file of any type. This is purely for demonstration purposes.
- `sample_file.txt.md5`
  - This is a text file that contains only the md5 sum of the `sample_file.txt`



The md5 sum file is only required when using the auto publish method. This is needed to ensure that the file to be transferred is complete and ready to be published.



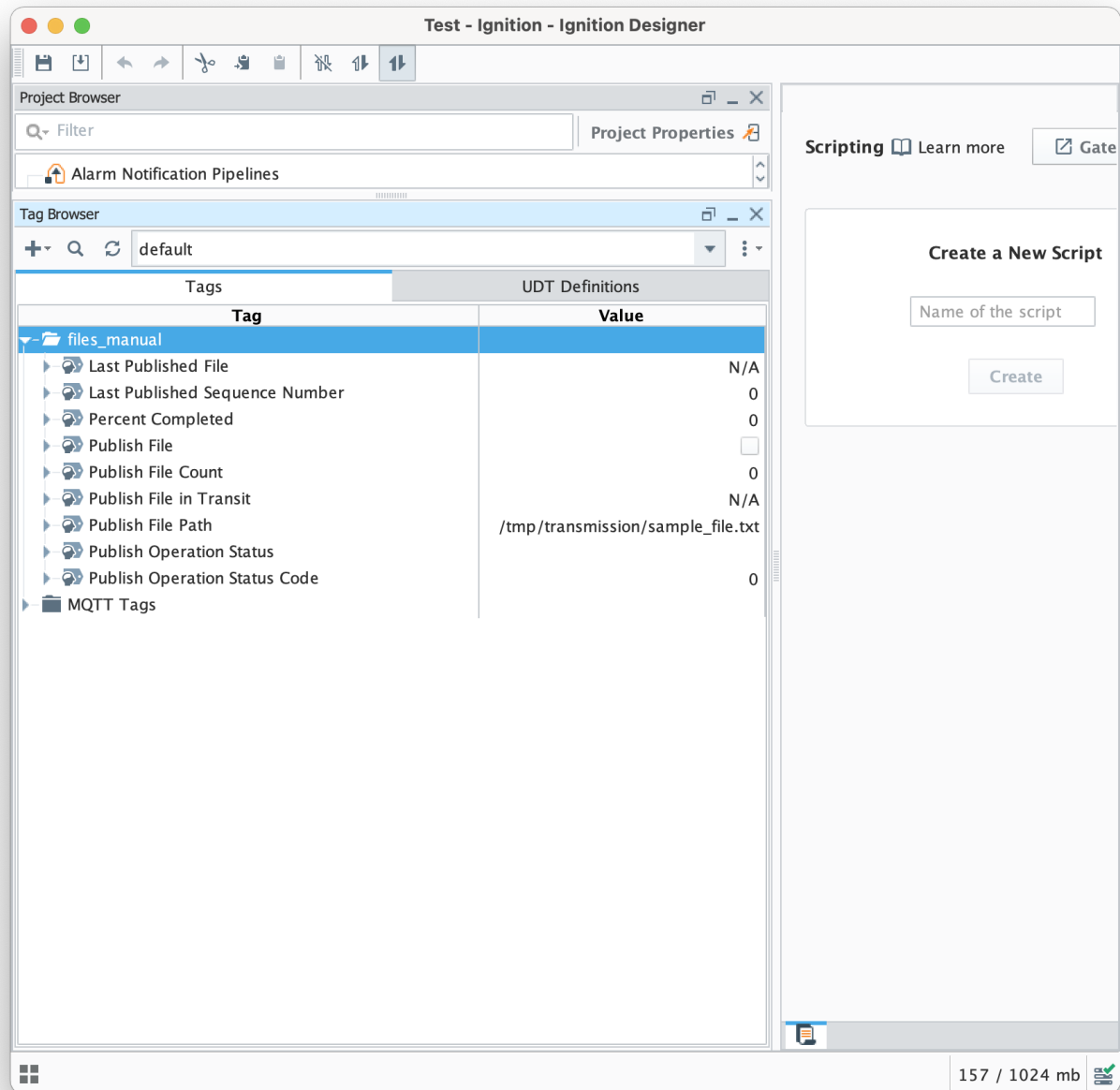
If you are testing with a different file, certain characters have special meanings when used in filenames such as "\*" for wildcards, and "\" in filename paths. If a file you are trying to publish contains any of the characters listed below, it will prevent files from being written to the file system.

" \* : < > ? / \ |

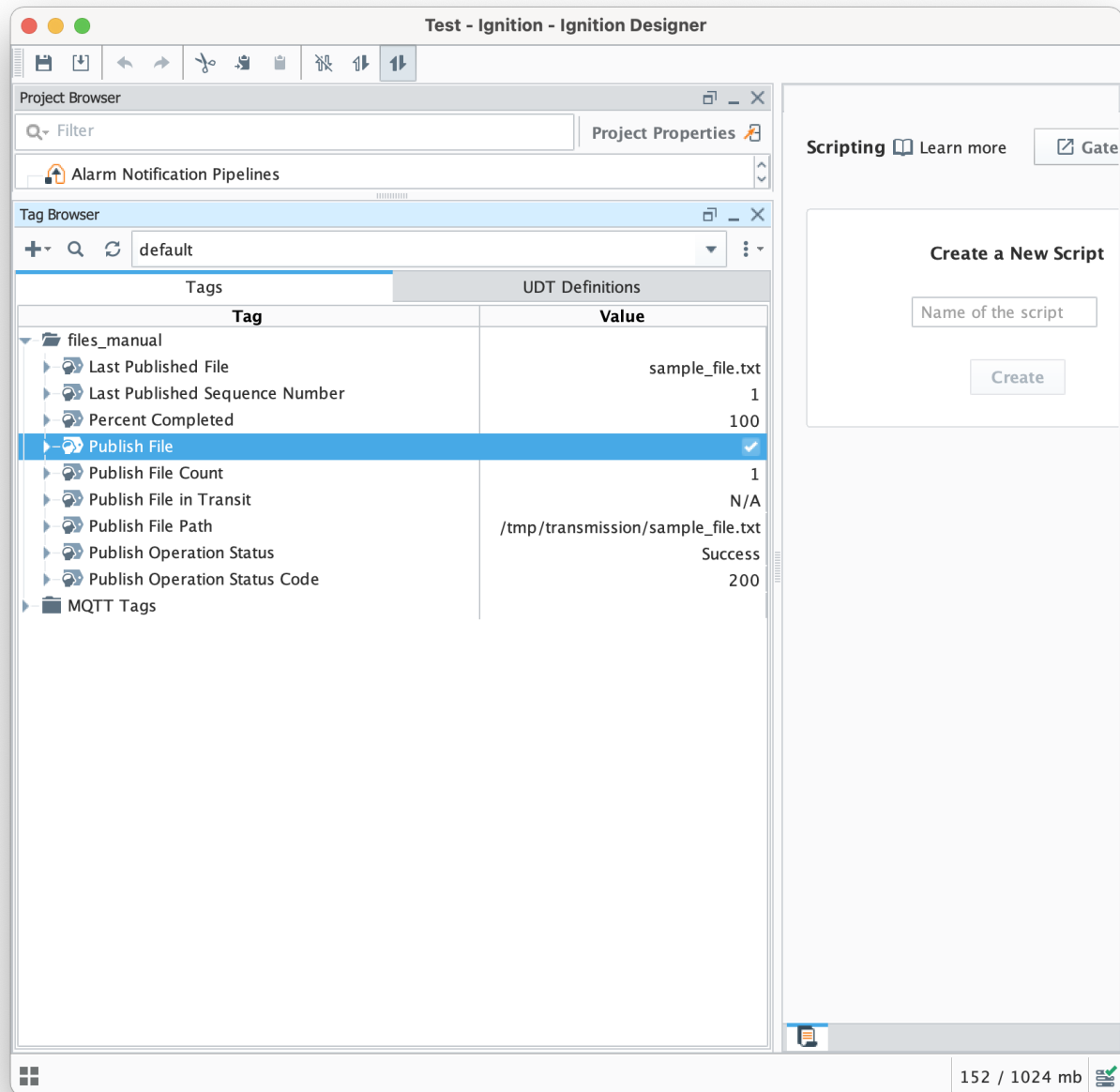
Leading and trailing spaces in filenames and filenames ending in '.' are also not supported.

With the file to be transferred in place, we can now set the 'Publish File Path' tag.

This path must be the full path to the target file to publish over MQTT. In this example, we're using the path of `/tmp/transmission/sample_file.txt`. However, on a Windows system that would look something like `C:\fullpath\to\my\sample_file.txt`. When this is set, you should see something like this



Finally, click the 'Publish File' boolean tag. This will begin the file publish process which will cause the file to be published. After doing so, you should see something similar to what is shown below.



You should also now see the file to the location you set as the 'Base File Directory' in your MQTT Engine namespace configuration for Sparkplug B.

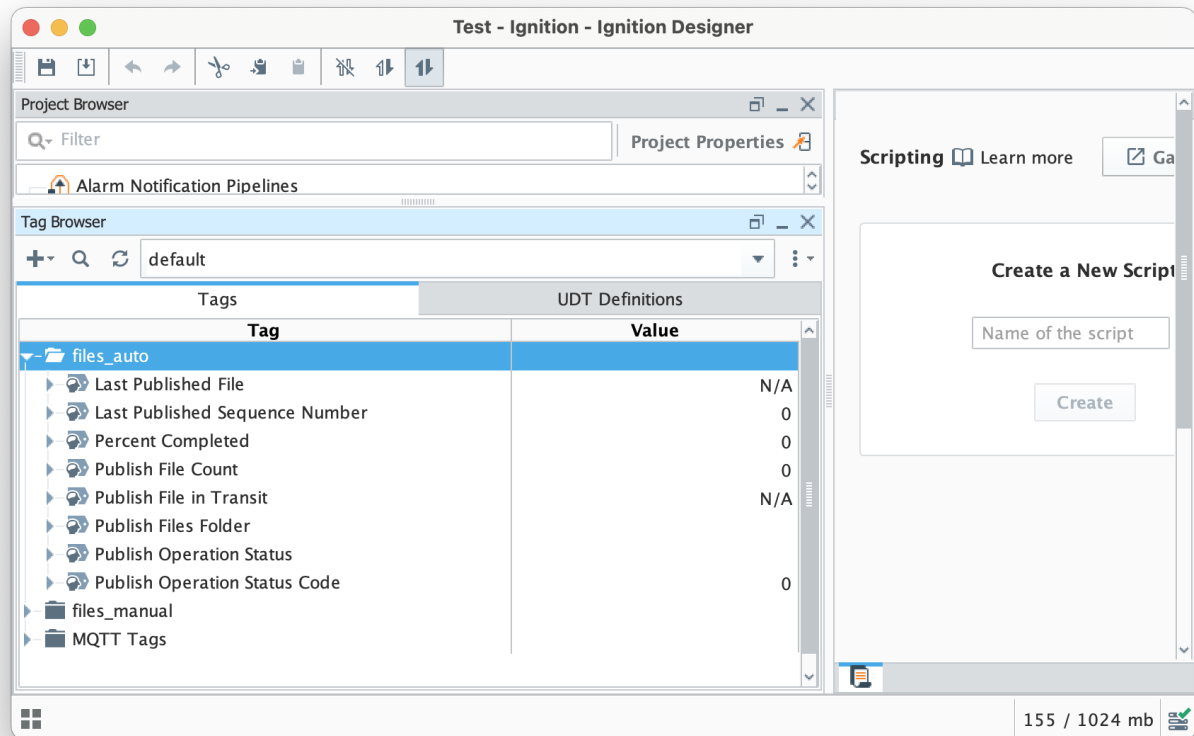


The files will be left on the originating filesystem after a successful publish and will need to be removed manually

## Auto Publish of Files

With a File record created following the steps in [Configure MQTT Transmission to publish files](#) with 'Enable Auto-Publish' checked and the Tag Folder Path set to "files\_auto", everything should be configured to send files from MQTT Transmission to MQTT Engine.

Open Ignition Designer on the system running MQTT Transmission. You should see the control and information tags created in the specified tag provider and folder as shown below:



The control and information tags created in the folder are:

Name	Type	Description
Last Published File	String	Name of last published file
Last Published Sequence Number	Integer	Sequence number of last published file since last reset of metrics
Percent Completed	Byte	Publish completion percent for file being published
Publish File Count	Long	Number of files published since last reset of metrics
Publish File in Transit	String	Name of current file being published
Publish Files Folder	String	Full path to the target folder containing the files to publish over MQTT
Publish Operation Status	String	Status description of current publish operation
Publish Operation Status Code	Integer	Status code for current publish operation
Reset	Boolean	Trigger to reset publish metrics

At this point, we just need to tell MQTT Transmission which folder contains the file(s) to publish. Download and unzip this sample file to some location and note that location.

This [sample\\_file.zip](#) contains two text files which are:

- `sample_file.txt`
  - This is a text file with an arbitrary string. But, it could be a file of any type. This is purely for demonstration purposes.
- `sample_file.txt.md5`
  - This is a text file that contains only the md5 sum of the `sample_file.txt`



If you are testing with a different file, certain characters have special meanings when used in filenames such as "\*" for wildcards, and "\" in filename paths. If a file you are trying to publish contains any of the characters listed below, it will prevent files from being written to the file system.

" \* : < > ? / \ |

Leading and trailing spaces in filenames and filenames ending in '.' are also not supported.



When transferring files using auto\_publish, MQTT Transmission requires two files to be present before it will transfer the target file. The first is the file itself. The second is a file that has the same name as the target file followed by a '.md5' extension. The file containing the md5 sum of the target file will not be transferred.

The contents of that file must contain the Message Digest Algorithm 5 (or MD5 sum) of the file. The MD5 sum can be calculated using command line utilities on most operating systems or through scripting in Ignition. Here are some examples:

#### Linux

```
ubuntu@linux-host:~$ md5sum myfile.bin
07180622a24ebf905cf5f770cd54197a  myfile.bin

# In the above example, the md5 sum is: 07180622a24ebf905cf5f770cd54197a
```

#### OSX

```
user@osxhost:~$ md5 sample_file.txt
MD5 (sample_file.txt) = 85324ffbcc7d97c478adf53796aff787

# In the above example, the md5 sum is: 85324ffbcc7d97c478adf53796aff787
```

#### Windows

```
Get-FileHash -Algorithm MD5 .\some_file.iso

Algorithm      Hash
Path
-----
-----
MD5            80FD169D3FDADBC97E66C168F796B1BF      C:
\temp\some_file.iso

# In the above example, the md5 sum is: 80FD169D3FDADBC97E66C168F796B1BF
```

#### Ignition Script


```
import hashlib

# File to create md5 sum
file_name = "D:\MyFiles\test_file.txt"

# Open,close, read file and calculate MD5 on its contents
with open(file_name, 'rb') as file_to_check:
    # read contents of the file
    data = file_to_check.read()
    # pipe contents of the file through
    md5_returned = hashlib.md5(data).hexdigest()


# Save md5 sum file
f = open(file_name + ".md5", "w")
f.write(md5_returned)
f.close()
```

MQTT Transmission will scan the folder specified in the tag, with a frequency defined by the File Scan Rate, and publish all files that appear in this directory automatically.

 When the file is successfully transferred both the file and associated md5 sum file will be removed from the originating filesystem.

Viewing the messages in the Ignition logs

The records published from MQTT Transmission can be viewed in the Ignition Gateway logs by setting the `com.cirruslink.mqtt.engine.gateway.sparkplug.SparkplugPayloadHandler` logger to TRACE.

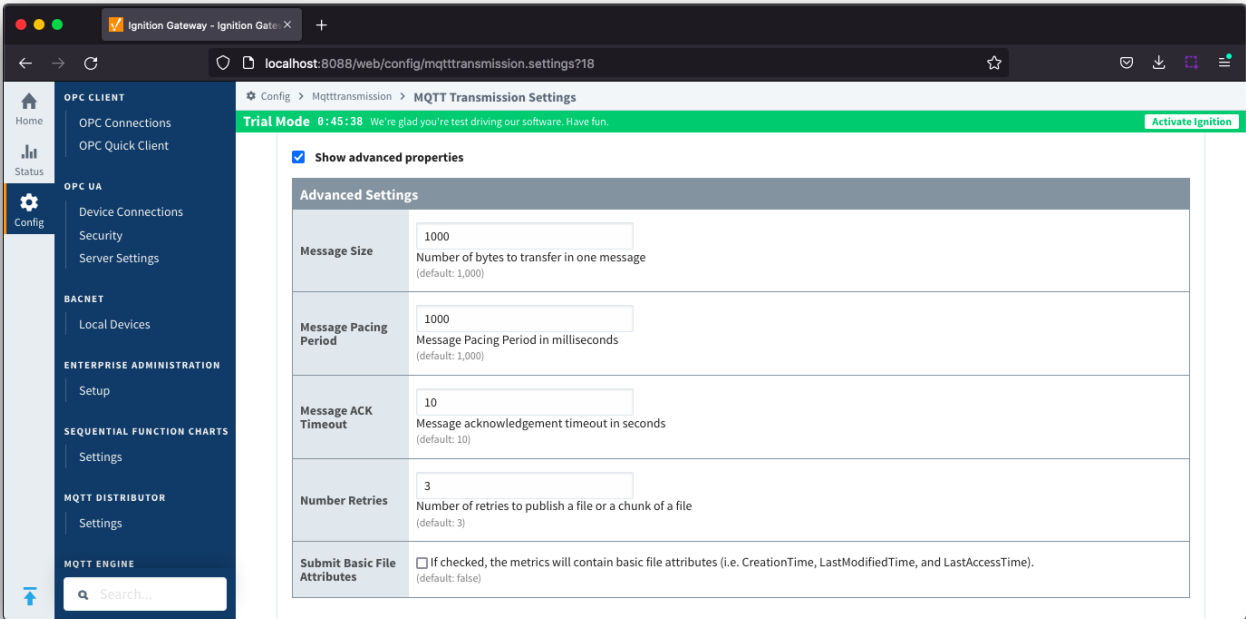
 Review the [Gateway Loggers](#) from Ignition for details on how to do this.

D	SparkplugPayloadHandler	22Dec2022 10:48:26	Got Sparkplug message: spBv1.0/My MQTT Group/NDATA/Edge Node 994161
T	SparkplugPayloadHandler	22Dec2022 10:48:26	On topic=spBv1.0/My MQTT Group/NDATA/Edge Node 994161: Incoming payload: Sparkplug8Payload [timestamp=1671727706485, metrics=[Metric [name=sample_file.txt, alias=null, timestamp=1671727706485, dataType=File, isHistorical=null, isTransient=null, metaData=MetaData [isMultiPart=false, contentType=application/octet-stream, size=110, seq=1, fileName=sample_file.txt, fileType=, md5=85324ffbcc7d97c478adf53796aff787, description=], properties=PropertySet [propertyMap=[filePublishingTagFolderPath=PropertyValue [type=String, value=my_files/, isNull=false], filePublishingTagProvider=PropertyValue [type=String, value=default, isNull=false]], value=File [fileName=sample_file.txt, bytes=[84, 104, 105, 115, 32, 105, 115, 32, 97, 32, 115, 97, 109, 112, 108, 101, 32, 102, 105, 108, 101, 32, 116, 111, 32, 116, 101, 115, 116, 32, 102, 105, 108, 101, 32, 116, 114, 97, 110, 115, 102, 101, 114, 32, 99, 97, 112, 97, 98, 105, 108, 105, 116, 105, 101, 115, 32, 111, 102, 32, 77, 81, 84, 84, 32, 84, 114, 97, 110, 115, 109, 105, 115, 115, 105, 111, 110, 32, 116, 111, 32, 77, 81, 84, 84, 32, 69, 110, 103, 105, 110, 101, 32, 117, 115, 105, 110, 103, 32, 83, 112, 97, 114, 107, 112, 108, 117, 103, 33, 10]], isNull=false]], seq=5, uuid=null, body=null]

Extra Activities

At this point you have a fully functional system which can be expanded or modified as required. Below are some addition configuration options to try:

**Adjust the file transfer configuration to define how quickly a file transfer occurs and how long before a timeout will occur on failure to receive host side acknowledgements based on the size of files you are sending and any bandwidth constraints that you may have.**



## Setup two Ignition systems - a host system installed with MQTT Engine and MQTT Distributor and edge system installed with MQTT Transmission

By default, MQTT Transmission is configured to connect to MQTT Distributor on <tcp://localhost:1883>. As the MQTT Distributor is installed on a different server, we will need to edit the configuration to allow MQTT Transmission to connect to the host server.



The MQTT Server must be accessible to the Ignition system running MQTT Transmission. In order for this to work port 1883 must be open to outside connections. Make sure this is allowed via your operating system's firewall configuration and any anti-virus software you may have installed.

This configuration is not recommended for production systems. If running this in production, you should be using TLS encryption on port 8883. More information on how to configure TLS can be found [here](#).

Navigate to the MQTT Transmission > Settings in the left side bar of the Ignition Gateway UI and select the Servers tab.

Change the server URL to point to the server IP address or domain name of the Ignition system running MQTT Distributor and save the configuration.

## Additional Resources

- Inductive Automation's Ignition download with free trial
  - [Current Ignition Release](#)
- Cirrus Link Solutions Modules for Ignition
  - [Ignition Strategic Partner Modules](#)
- Support questions
  - Check out the Cirrus Link Forum: <https://forum.cirrus-link.com/>
  - Contact support: [support@cirrus-link.com](mailto:support@cirrus-link.com)
- Sales questions
  - Email: [sales@cirrus-link.com](mailto:sales@cirrus-link.com)
  - Phone: +1 (844) 924-7787
- About Cirrus Link
  - <https://www.cirrus-link.com/about-us/>