

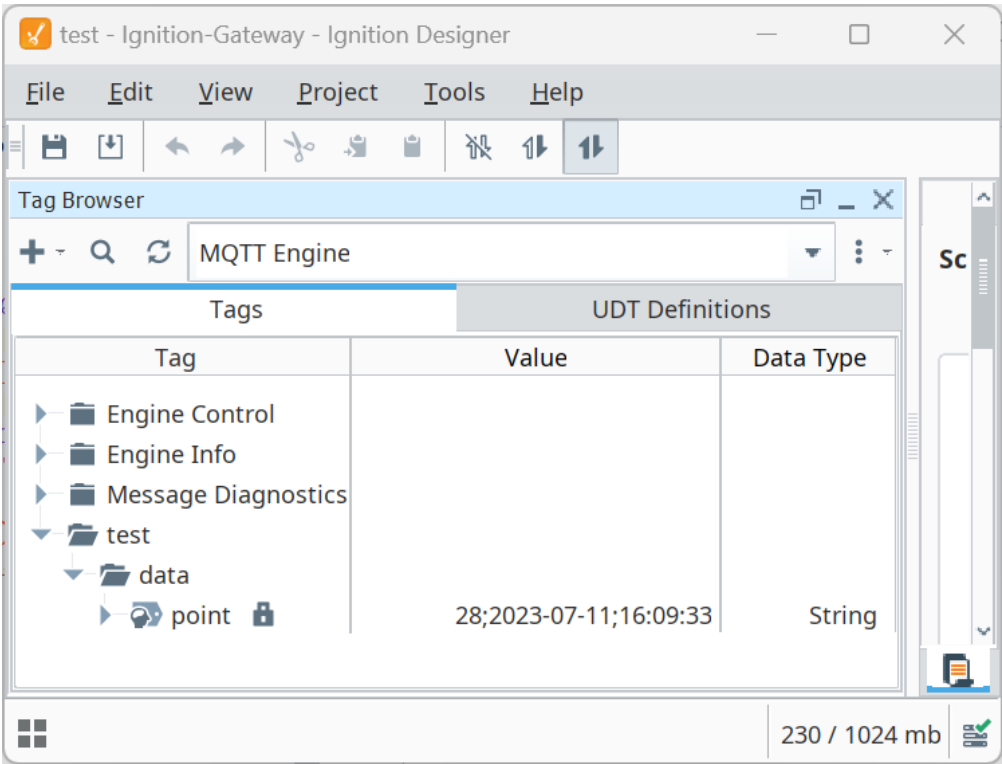
Managing Ignition timestamps for MQTT data when using custom namespaces

One common question is how to use the MQTT message's payload timestamp property rather than the time that the message arrives on the broker or received by Ignition.

Our recommended approach is to set an MQTT Engine custom namespace to parse the payload as a STRING and then use the Ignition scripting capabilities to parse out the required elements of the text string and write a Fully Qualified Value to a tag in a non managed tag provider.

Let say we have a publish received on the topic "test/data/point" with a payload "28;2023-07-11;16:09:33" where the payload includes a value and timestamp.

MQTT Engine will create a two folders "test" and "data" and a tag "point" with the value of "28;2023-07-11;16:09:33".



Now we can use an Ignition Tag Change Event Script on the tag to parse out the required elements of the text string and write a Fully Qualified Value to a tag in a non managed tag provider.

For example:

```

from com.inductiveautomation.ignition.common.model.values import BasicQualifiedValue, QualityCode
from datetime import datetime

def convertString(string):
    value = int(string.split(";")[0])
    year = int(string.split(";")[1].split("-")[0])
    month = int(string.split(";")[1].split("-")[1])
    day = int(string.split(";")[1].split("-")[2])
    hours = int(string.split(";")[2].split(":")[0])
    mins = int(string.split(";")[2].split(":")[1])
    seconds = int(string.split(";")[2].split(":")[2])

    date = system.date.getDate(year,month-1,day)
    datetime = system.date.setTime(date, hours, mins, seconds)

    return value, datetime

#parse MQTT Engine string tag
path = "[MQTT Engine]test/data/point"
value,timestamp = convertString(system.tag.readBlocking([path])[0].value)

#set quality
quality = QualityCode.Good

#set Basic Qualified Value
bqv = BasicQualifiedValue(value,quality,timestamp)

#set new tag path
newpath = "[MQTT Tags]test/data/point"

#write Basic Qualified Value
system.tag.writeBlocking(newpath,bqv)

```

