Tag Event Scripts

Tag Event Overview

Scripts can be attached to Tags. When you edit a Tag, you can navigate to the Tag Events section to see a list of all of the available events. Those events are

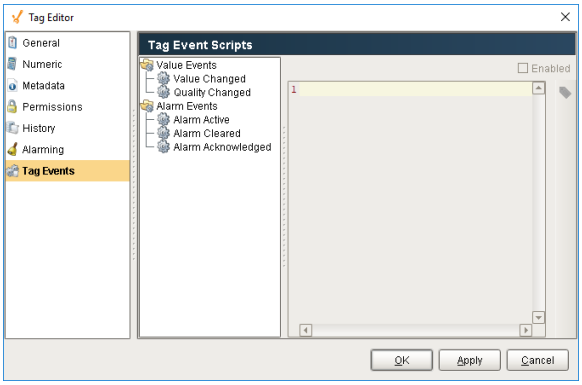
- Value Changed
- Quality Changed
- Alarm Active
- Alarm Cleared
- Alarm Acknowledged

Each event is unique and can have specialized arguments depending on the event. Also, because Tags are stored in the Gateway, all these scripts fire in the Gateway scope.

> Because these scripts are Gateway scoped, certain things like print statements will not print to the Designer console, but will print instead to the wrapper log file in Ignition's installation directory. Keep that in mind when doing any testing.

On this page

...



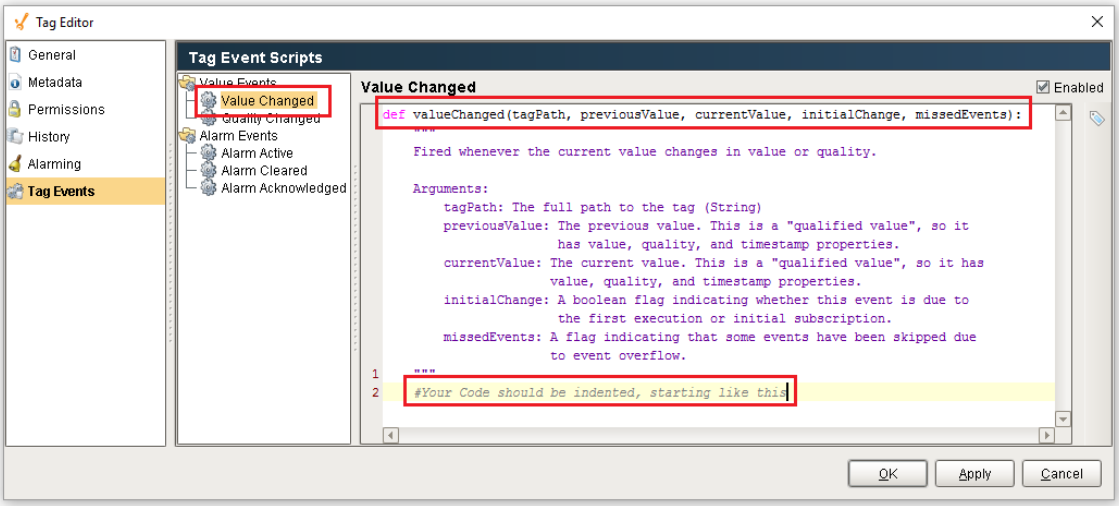
INDUCTIVE UNIVERSITY

Tag Scripts

Watch the Video

Event Scripts are Functions

Once an event has been selected, note that the top of the text area is defining a function. As a result, all code that should execute when this event occurs should be indented at least once to be included in the definition.



Event Overview

Value Changed

The Value Changed event is fired whenever the value of this particular Tag changes. This event has a variety of arguments available for use in the script:

- **String tagPath** - The full path to the Tag.
- **Object previousValue** - The previous value. This is a qualified value, so it has value, quality, and timestamp properties.
- **Object currentValue** - The current value. This is a qualified value, so it has value, quality, and timestamp properties.
- **Boolean initialChange** - A boolean flag indicating whether this event is due to the initial subscription or the first execution after a Tag update.
- **Boolean missedEvents** - A flag indicating that some events have been skipped due to an event overflow.

The currentValue and previousValue arguments are qualified values. This means that to get to the actual current value, it would look like currentValue.value.

Quality Changed

The Quality Changed event is fired whenever the quality of this particular Tag changes. Since Tags use a 'qualified value', which include a value, quality, and timestamp, this script will fire whenever any of those change. This event has a variety of arguments available for use in the script:

- **String tagPath** - The full path to the Tag.
- **Object previousValue** - The previous value. This is a qualified value, so it has value, quality, and timestamp properties.
- **Object currentValue** - The current value. This is a qualified value, so it has value, quality, and timestamp properties.
- **Boolean initialChange** - A boolean flag indicating whether this event is due to the initial subscription or the first execution after a Tag update.
- **Boolean missedEvents** - A flag indicating that some events have been skipped due to an event overflow.

Alarm Active

The Alarm Active event fires whenever an alarm becomes active on the Tag. This event has a variety of arguments available for use in the script:

- **String tagPath** - The full path to the Tag.
- **String alarmName** - The name of the alarm. This does not include the full alarm path.
- **Object alarmEvent** - The full alarm event object. The properties available to this object are:
  - id
  - source
  - name
  - priority
  - displayPath
  - displayPathOrSource (the display path if it is set, otherwise the source)
  - state (the current state, active/cleared + acked/unacked)
  - lastEventState (the last transition)
  - cleared (a boolean if the alarm is currently cleared)
  - acked (a boolean if the alarm is currently acknowledged)
  - shelved (a boolean if the alarm is currently acknowledged)
  - notes
- **String alarmPath** - The full alarm path.
- **Boolean missedEvents** - A flag indicating that some events have been skipped due to an event overflow.

Alarm Cleared

The Alarm Cleared event fires whenever an alarm becomes cleared on the Tag.This event has a variety of arguments available for use in the script:

- **String tagPath** - The full path to the Tag.
- **String alarmName** - The name of the alarm. This does not include the full alarm path.
- **Object alarmEvent** - The full alarm event object. See the Alarm Active alarmEvent object for the full list of available properties.

- **String alarmPath** - The full alarm path.
- **Boolean missedEvents** - A flag indicating that some events have been skipped due to an event overflow.

Alarm Acknowledged

The Alarm Acknowledged event fires whenever an alarm event becomes acknowledged on the Tag.This event has a variety of arguments available for use in the script:

- **String tagPath** - The full path to the Tag.
- **String alarmName** - The name of the alarm. This does not include the full alarm path.
- **Object alarmEvent** - The full alarm event object. See the Alarm Active alarmEvent object for the full list of available properties.
- **String alarmPath** - The full alarm path.
- **String ackedBy** - The full path to the user that acknowledged the alarm.
- **Boolean missedEvents** - A flag indicating that some events have been skipped due to an event overflow.

Examples

Printing all parameters

```
# This script will fetch all of the possible parameters in the tag changed script.
# Note that this will not print out to the console, but it will print to the wrapper logs located on the
gateway.

path = tagPath
prev = previousValue
cur = currentValue
init = initialChange
missed = missedEvents
print path, prev.value, cur.quality, init, missed
```

Automatic Reset

```
# This code can be used on a Value Changed script, and will automatically reset the value of the tag to
0 after it goes above 3000.
# This can be useful for counter tags.
if currentValue.value > 3000:
                system.tag.write("IntegerCounterTag", 0)
```

Copy to another Tag

```
# This code can be used on a Value Changed script, and will record the highest value of the current tag
onto another memory tag.
# This can be useful for recording the highest temperature.
highestRecordedTemp = system.tag.read("HighestTempTag").value
if currentValue.value > highestRecordedTemp:
        system.tag.write("HighestTempTag", currentValue.value)
```

Automatically reacting to an alarm

```
# This code can be used on an Alarm Active script, and can be used to automatically react to an alarm to
prevent a disaster from occurring,
# if personnel were not able to react in time.
if alarmName == "Tank Pressure Critical":
        system.tag.write("PressureReleaseValveTag", 1)
if alarmName == "Tank Pressure Too Low":
        system.tag.write("TankFillTag", 1)
```