

Ignition 7.9 User Manual — Alarming

Generated 2026-10-12 from
<https://www.docs.inductiveautomation.com/docs/7.9/>

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarming

How Alarming Works

Alarming is a core feature of the Ignition platform. It provides the functionality and flexibility to configure your alarms, provides up-to-date status of alarms, store alarm history, build the logic for how, why, and when alarm notifications are delivered, manage alarm notifications for user groups, and send Email, SMS or Voice notifications. With all these features and functions in Alarming, you can easily create alarms, and design and manage your alarm notifications any way you choose!

Alarming lets you start creating alarms right out of the gate, or if you prefer, you can customize an entire alarm notification process with pipelines, schedules, rosters, and much more. Alarming features and functions translate to endless possibilities for configuring and managing your alarms and notifications. Once you review the features and functions of Alarming, you too, will soon see the endless possibilities for creating robust alarms and alarm notifications in your facility!

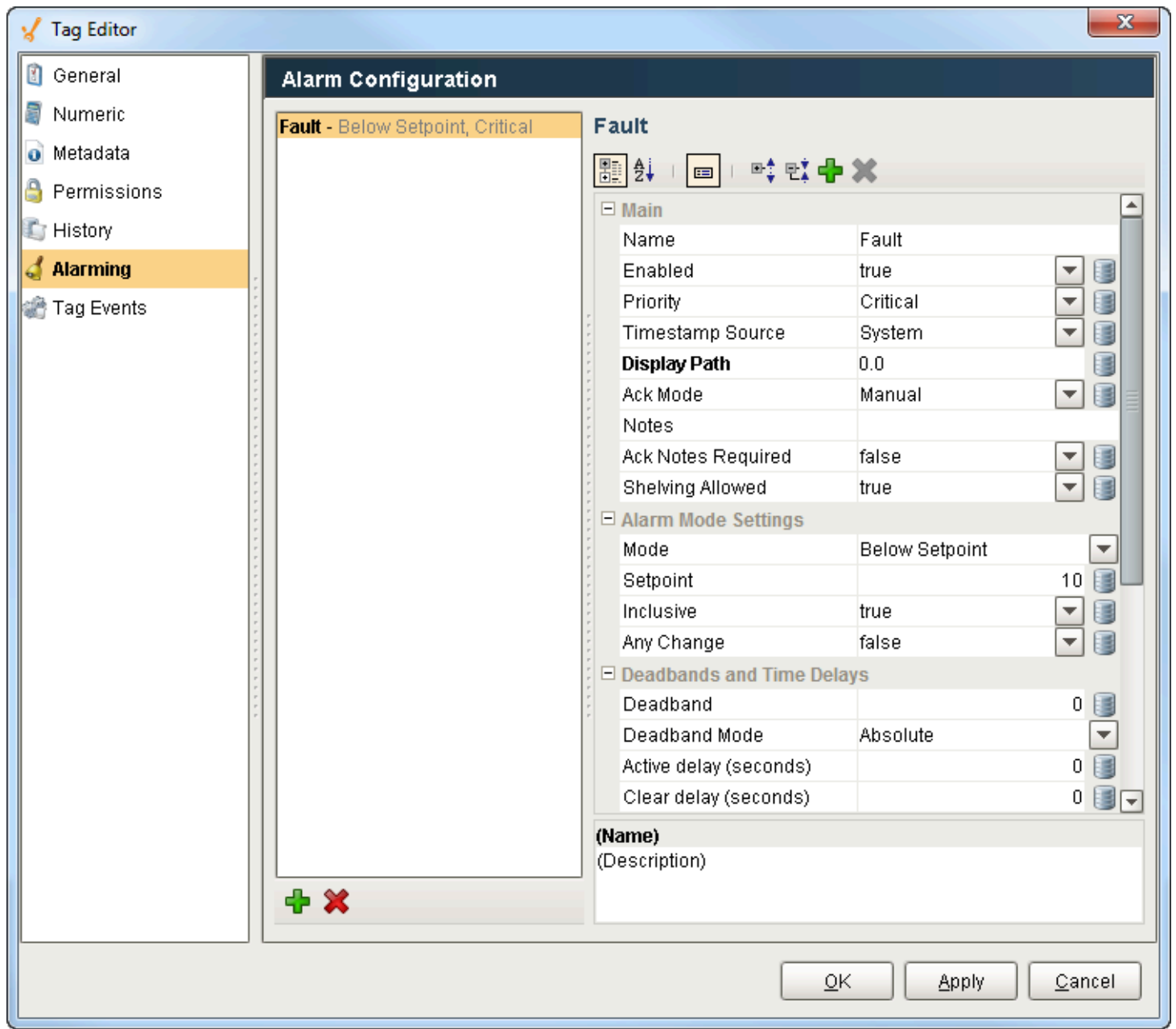
- **Alarm Types and Alarm Settings** - Available alarm types including a host of alarm settings so you can build and change the behavior of you alarm
- **Alarm Status** - Displays current status of alarms
- **Alarm Journal** - Stores historical information about alarms
- **Alarm Notification** - Lets you select the delivery channel for how alarms notifications are sent (Email, SMS, and Voice)
- **Users, Schedules, and Rosters** - Defines users, who will receive alarms, schedules and rosters
- **Alarm Notification Pipelines** - Lets you define how alarm notifications are sent out and acknowledged

Setting up Alarms

Anything is possible with alarms! Alarms have a ton of settings that you can use to create almost any type of alarm in Ignition. You can configure alarms with one alarm on a tag, multiple alarms on a tag, and even add alarms in UDTs so every instance of that tag will automatically have alarms configured when a new instance of your tag is created. You can use the alarm settings to create alarms that equal or don't equal a setpoint, above or below a setpoint, between setpoints, outside setpoints, dynamic setpoints, out of range, bad quality, etc. The conditions are limitless.

Alarms are super easy to create. All you have to do is select your tag, add an alarm, configure your alarm settings, and you're in business.

Information about configuring alarms can be found in [Alarming Properties](#) in the [Tag Intro](#) and [Setting Up Alarms](#) sections.



How to Monitor Alarm Status

Setting up the Alarm Status Table works like magic! All you have to do is drag the [Alarm Status Table](#) component onto your window, and the current alarms are displayed immediately into one view! It is highly customizable and can be configured to show active, unacknowledged, cleared, and acknowledged alarms. You can [filter on Alarm Status properties](#), show or hide alarm property information, reorganize columns, view alarm details, notes and history of an alarm, not to mention, [Acknowledge](#) or [Shelve](#) a selected alarm.

☐	Display Path	Name	Active Time	Ack'd By	Ack Time	Current State	Priority
☒	Machine D	High Alarm	3/30/16 2:18 PM			Active, Unacknowledged	Critical
☐	Machine B	High Alarm	3/21/16 8:12 AM	admin	3/25/16 4:00 PM	Active, Acknowledged	Critical
☐	Machine B	:pw Alarm	3/21/16 8:05 AM	admin	3/28/16 10:15 AM	Active, Acknowledged	Medium
☐	Machine B	Low Alarm	3/28/16 8:57 AM	admin	3/28/16 10:15 AM	Active, Acknowledged	Medium
☐	Machine B	High Alarm	3/21/16 8:05 AM			Cleared, Unacknowledged	Critical
☐	Machine B	Low Alarm	3/21/16 8:12 AM			Cleared, Unacknowledged	Medium
☐	Machine B	Low Alarm	3/22/16 11:45 AM			Cleared, Unacknowledged	Medium
☐	Machine B	Low Alarm	3/21/16 8:05 AM			Cleared, Unacknowledged	Medium
☐	Machine B	Low Alarm	3/24/16 11:58 AM			Cleared, Unacknowledged	Medium
☐	Details	Notes	Low Alarm			Cleared, Unacknowledged	Medium

Clear Pipeline

Deadband

Active delay (seconds)

Clear delay (seconds)

Timestamp Source

Ack Mode

Notes

Is Shelved?

Shelf Expiration

eventid

On Active mode

0

0

0

System

Manual

Machine High

False

a8f0d8ab-06f5-4127-a147-a66f67683645

Above Setpoint

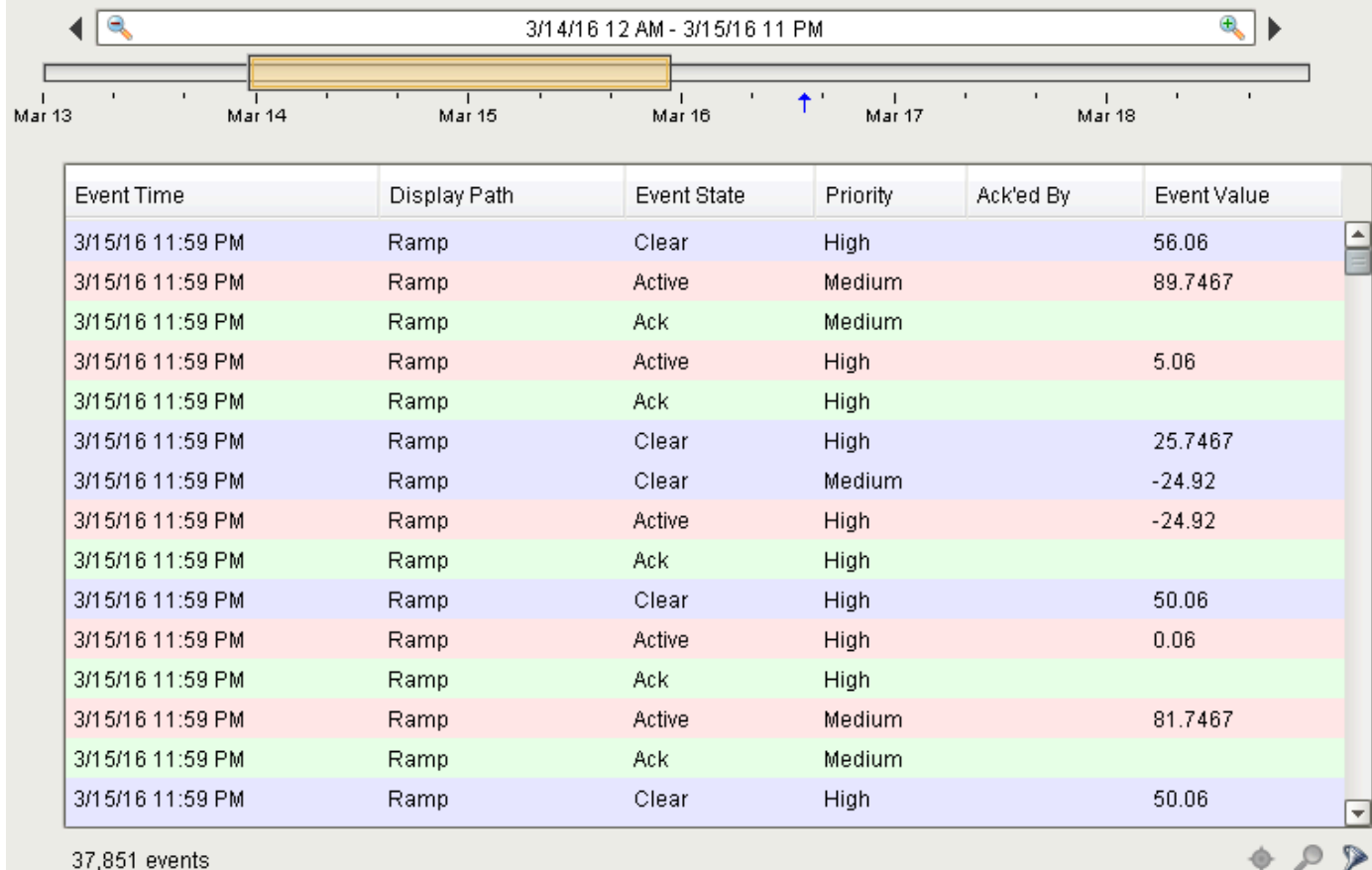
Acknowledge

Shelve

How to View Alarm History

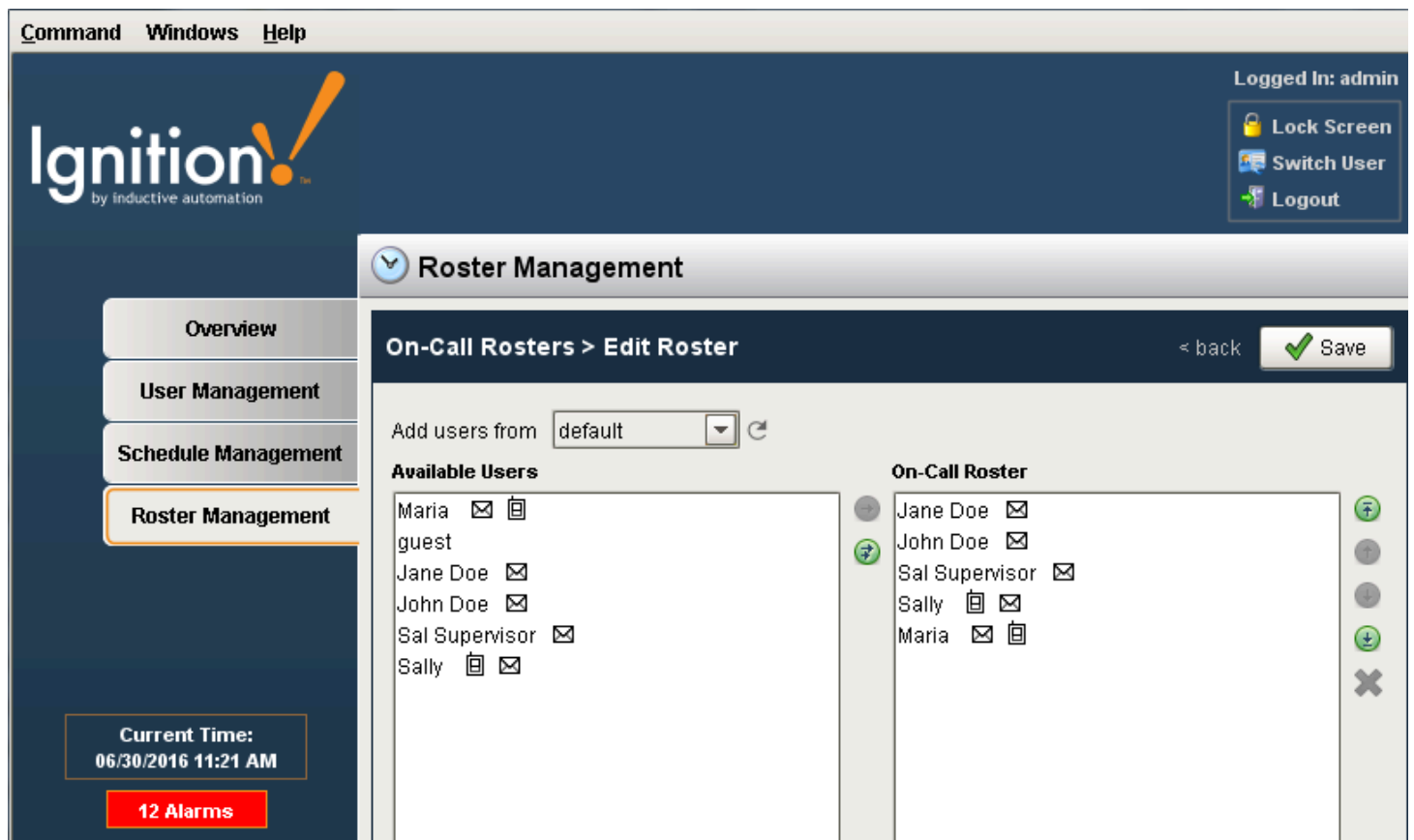
The **Alarm Journal** stores historical information about alarms in a database. It stores basic data about alarms that have occurred, such as their source and timestamp, associated data on an alarm, and the values of an alarm's properties at the time the event occurred. It captures all status changes for each alarm, as well as acknowledgement to an external SQL database of your choosing. To begin viewing alarm history, all you need to do is create an Alarm Journal Profile in the Gateway webpage.

Like the Alarm Status Table, you can **filter on alarm history properties**. The alarms are color coded so you know what each status represents. The **date range** is a very common filter type since users typically want to filter for alarm events within a specific period of time.



Alarm Notification

Alarms may also generate notifications that are delivered to users. This allows Ignition to immediately communicate events and problems to your users. The Notification system has access to Ignition's Authentication Profiles, so users can easily be added to notification rosters. Schedules can be created, which in turn allows users to receive notifications only when on-schedule, so there is no need to worry about notifying a supervisor or manager in the middle of the night.

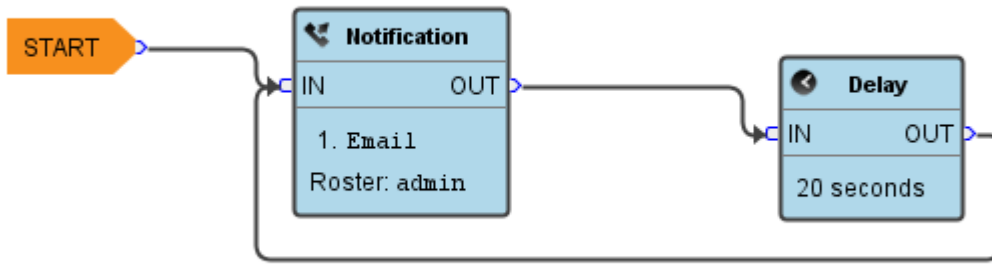


Using Alarm Notification Pipelines

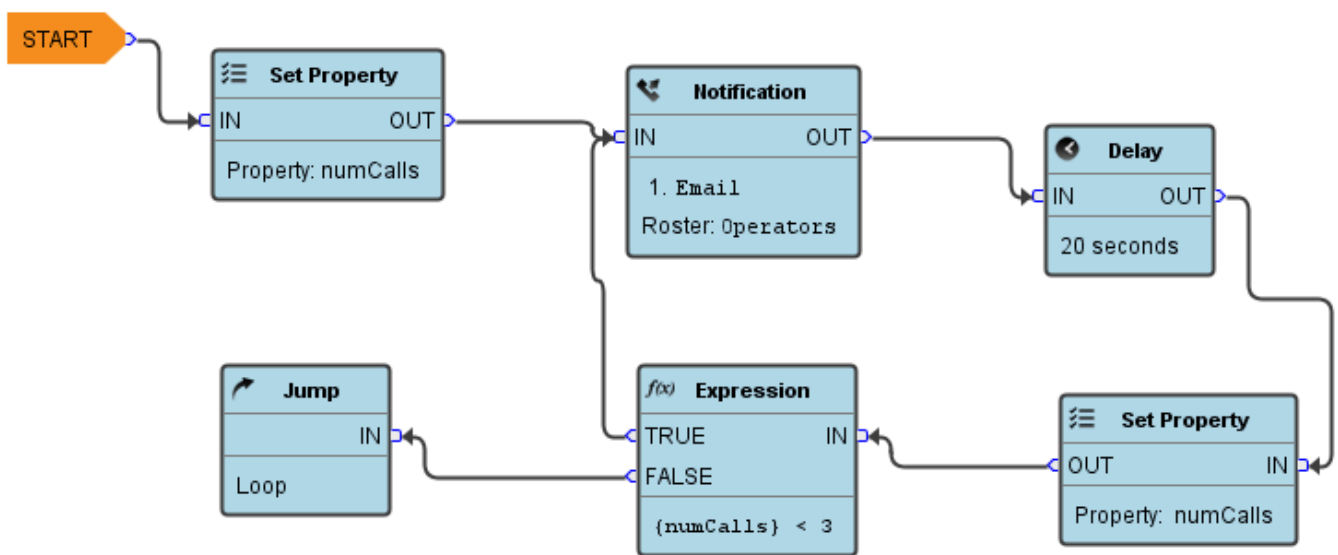
The **Alarm Notification Pipeline** feature is an innovative tool that lets you easily create routes for your alarms. By designing your own alarm notification routing, you have control of what happens when an alarm goes active, when an alarm is sent out, and who receives the alarm notification.

The alarm notification pipeline has a simple drag-and-drop interface so you can build various types of alarm logic. Its **built-in pipeline block** functionality lets you **delay** sending alarm notifications, **escalate** higher priority alarms to a different group of users, **consolidate** multiple alarms so recipients aren't flooded with messages, and **distribute** specific alarm types to different contact groups.

Alarm notification pipelines can be very simple to very complex. In this simple notification pipeline, when an alarm is triggered, the people listed in the **On-call Roster** are notified via Email. If no one acknowledges the alarm in 20 seconds, the alarm notification is routed back to the same people listed in the On-call Roster.



In this more complex alarm notification pipeline, if an operator doesn't respond to the alarm after three attempts, the pipeline jumps it to another pipeline (possibly an escalation pipeline).



To learn more about building your own pipelines, go to [Customizing Alarm Notification](#).

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Configuring Alarms



INDUCTIVE
UNIVERSITY

Configure Alarm on a Tag

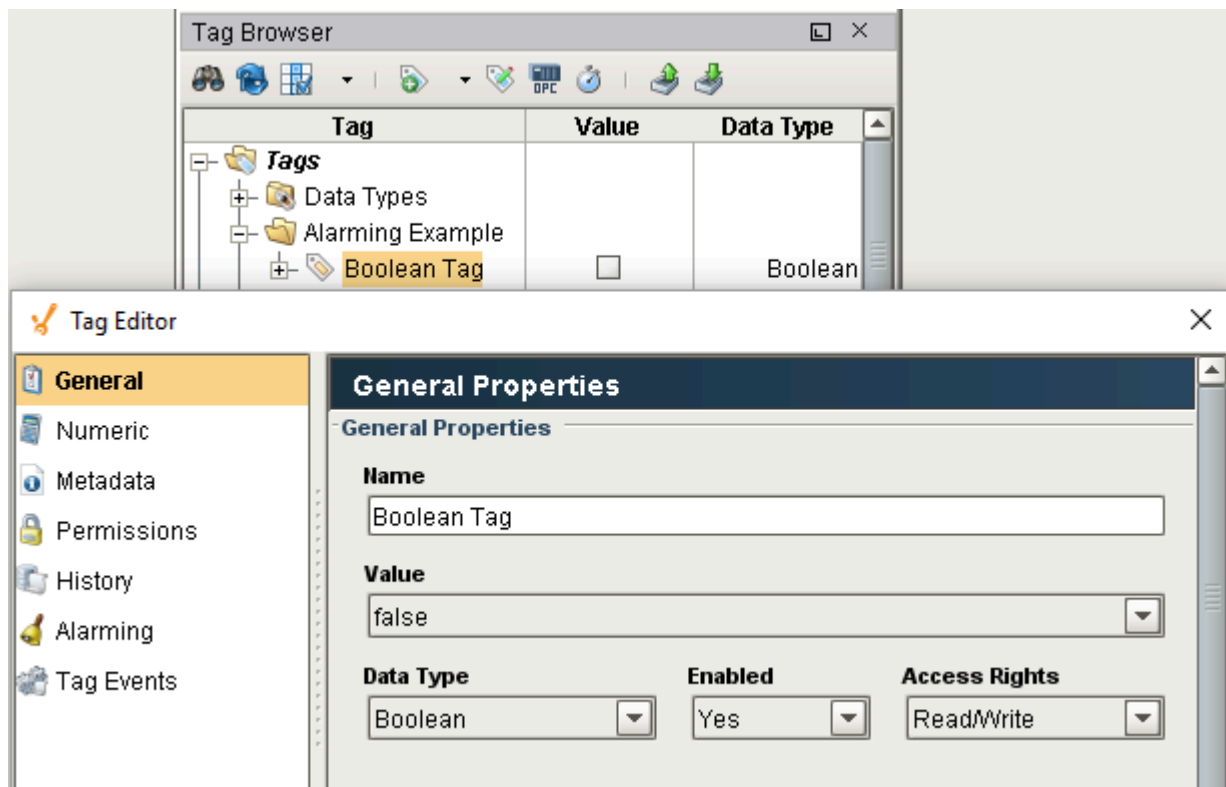
[Watch the video](#)

Alarms are defined on Tags or OPC items in [SQL Bridge \(Transaction Groups\)](#). Each alarm is a condition that will be evaluated when the value of the Tag changes. When the condition becomes true, the alarm is said to be **active**. When it becomes false, the alarm is said to be **clear**.

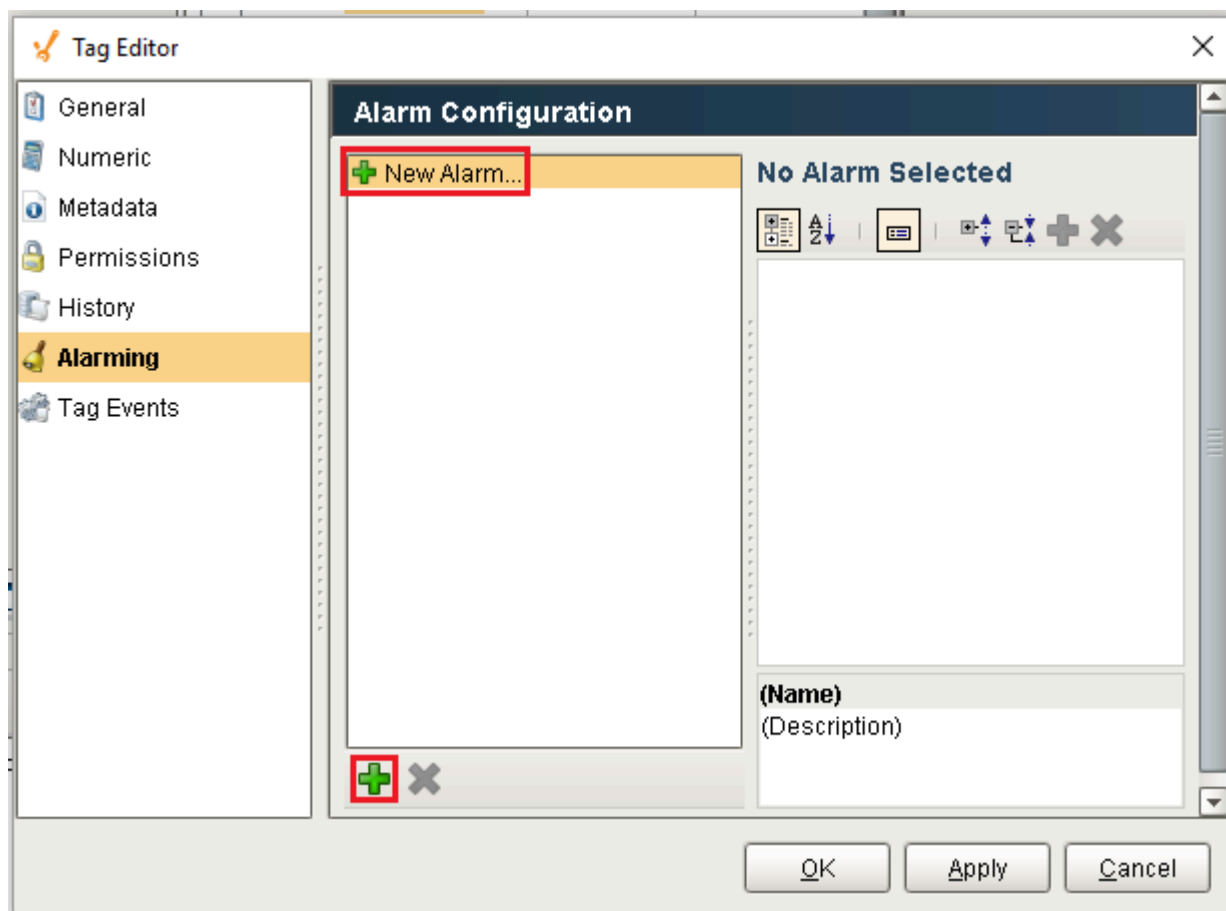
Alarms may also be **acknowledged**. This flags the alarm as such in Ignition, and the acknowledgment state of each alarm can be made visible throughout Ignition. In practice, acknowledgment allows users to 'claim' alarms, or otherwise let colleagues know that the alarm is being handled. Until an alarm has been acknowledged, it is said to be **unacknowledged**, or '**unacked**'.

Configuring an Alarm on a Tag

1. Double-click on the Tag to edit it. The Tag Editor window will appear. For this example, a Memory Tag with a Data Type of Boolean will be used



2. On the left side of the window, click on **Alarming**.
3. Either click on the green plus sign towards the top of the window, or double click the **+ New Alarm...** item to add a **New Alarm** to the Tag.



4. Enter a **Name** for the Alarm and set any other settings you wish. In the image below the **Name** was changed to "High Value" and the **Setpoint** was changed to 1. This means that when the

value on the Tag is equal to 1 (or True in the case of a boolean) then the alarm will become active.

Alarm

Main

Name	High Value
Enabled	true
Priority	Low
Timestamp Source	System
Display Path	
Ack Mode	Manual
Notes	
Ack Notes Required	false
Shelving Allowed	true

Alarm Mode Settings

Mode	Equal
Setpoint	1

5. Click **OK** to commit the new alarm.
6. Ignition will start monitoring the alarm immediately. Note the alarm icon 🚨 next to the Tag. This means that at least one alarm is configured on this Tag.

Tag	Value	Data Type
Tags		
Data Types		
Alarming Example		
Boolean Tag 🚨	1	Boolean
Area A		

Multiple Alarms on a Single Tag

Tags can have multiple alarms configured, each with unique setpoints and other configurations. The number of alarms that you can configure is virtually unlimited and will never exceed the bounds of reasonable design consideration. Adding additional alarms is just as easy as adding the first: simply click on the green plus sign + and start configuring the new alarm. This way separate alarms can monitor different setpoints, so Tags can have alarms for high and low setpoints.



**INDUCTIVE
UNIVERSITY**

Multiple Alarms on a Tag

[Watch the video](#)



TIP

Alarms can be copied and pasted with keyboard shortcuts. Select an alarm from the Tag Editor, press **Ctrl-C** to copy, and **Ctrl-V** to paste. Alarms with similar configurations can rapidly be replicated to hasten development.

Alarm Settings

Similar in concept to properties on Vision components, alarm settings, also known as alarm properties, allow you to modify the behavior of each alarm.

For example, in some situations a Tag may frequently enter and leave an alarm state, but only for a brief moment. Normally an alarm would be generated each time the alarm condition was met, but the **Active Delay** attribute will prevent any alarms from being generated until the value on the Tag has stayed in an alarm state for a set period of time.

Alarm	
Mode	Equal
Setpoint	0
Deadbands and Time Delays	
Deadband	0
Deadband Mode	Absolute
Active delay (seconds)	0
Clear delay (seconds)	0

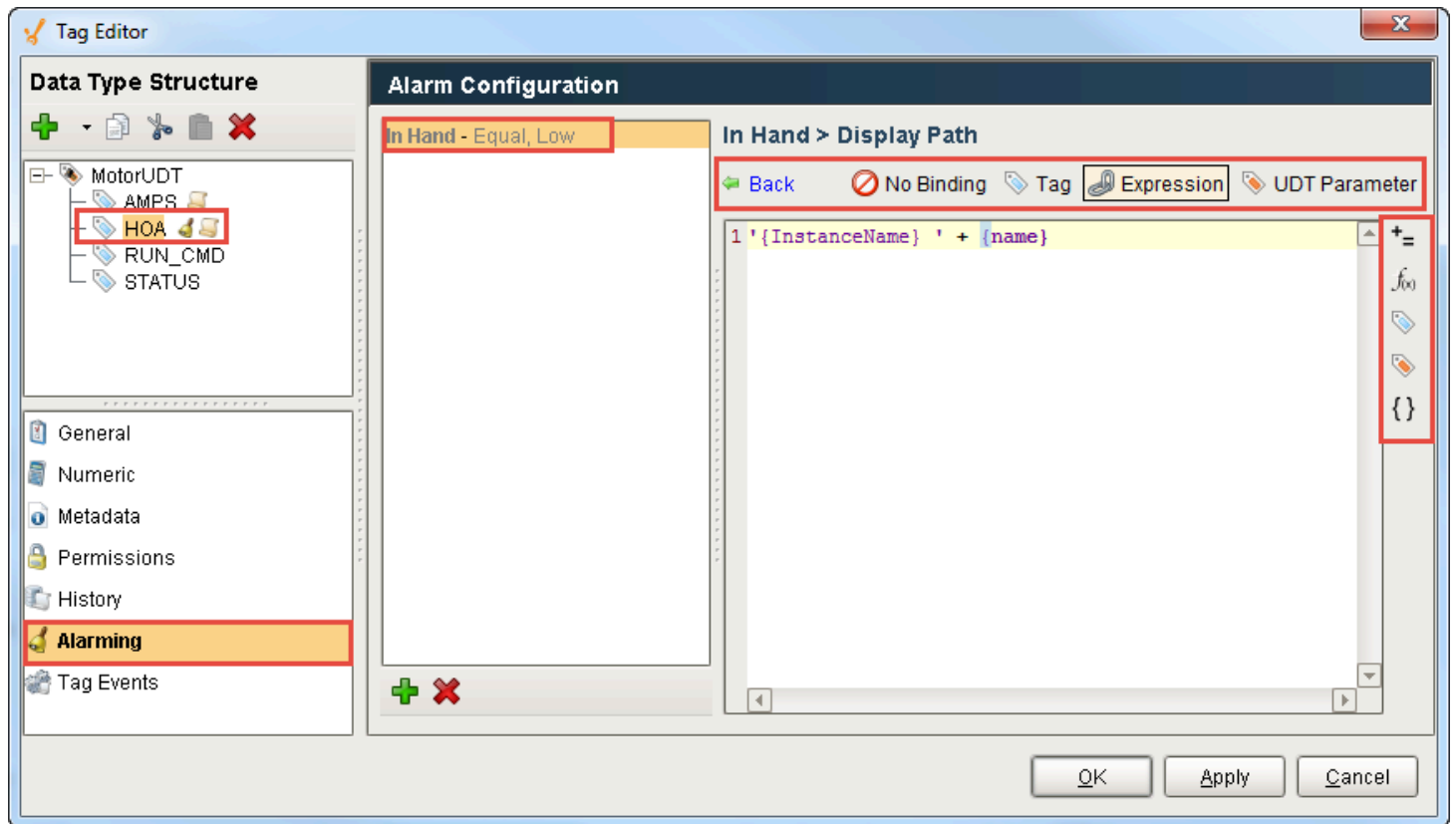
Descriptions of each attribute can be found on the [Tag Properties](#) page.

Bindings

Many alarm properties are bind-able, which means they can be bound to other Tags in the system, or expressions. For example, you might bind the enabled property to another Tag which represents whether or not your process is running, thereby disabling the alarm when production is stopped. Or, you might bind the setpoint of an alarm to a Tag that operators can manipulate, thereby letting the setpoint be changed at runtime.

To bind an alarm property, simply click on the binding icon , and the binding UI will slide in from the right. From here you can select the binding type (No Binding, Tag, Expression, or UDT Parameter, if applicable). Note that the expression can reference many useful values such as the Tag's value and

other settings of the alarm. When you've configured the binding to your liking, click on the "Back" link at the top of the screen.



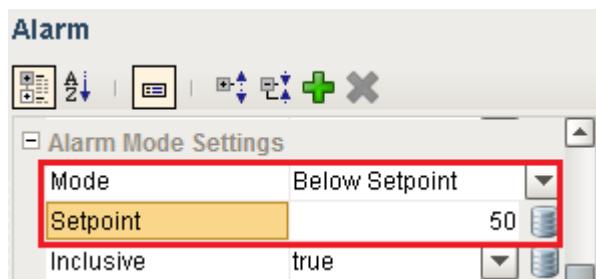
Associated Data

Alarms can be configured with custom data, known as **Associated Data**. These values can be static, driven by another tag, or evaluated with an expression. Associated Data can be viewed in **real-time**, and is recorded in the **Alarm Journal** system.

Modes and Setpoints

Each alarm is configured with one of many Modes, and usually one or more Setpoints. **Modes** determine the method in which alarm activity is evaluated, while **Setpoints** are the thresholds or limits that determine when a Tag is within the alarm state. Both properties work in together to determine when alarms become active, as well as cleared.

For example, when Mode is set to **Below Setpoint**, the Setpoint attribute is compared to the Tag's value. If the Tag value is less than the Setpoint value, the alarm becomes active. Once the value of the Tag is greater than the Setpoint value, the alarm will transition to a cleared state.



Alarming on Individual Bits

Additionally, alarms can become active by bit state. A common practice for PLC programming is to store alarm conditions as series of bits inside the PLC and expose them to the world as an integer value. Each bit is essentially a binary value. A series of 8 bits, for example, could be represented as an 8-bit integer. This integer value can be monitored by Ignition's OPC-UA server as an integer value as a Tag. This Tag can then have as many alarms as the integer has bits by monitoring the integer's bit state. Using the Bit Position Mode, multiple alarms in Ignition can be configured to monitor the state of each bit.

The following table shows binary and decimal equivalents:

Binary	Decimal
10001110	142

In this case the Tag would be an integer with a value of 142. This Tag would have eight alarms. Each one of these alarms become active in accordance with its Boolean value.

To Create the Alarms on an Integer Tag Value

1. Edit the Tag that has an integer value.
2. Create an alarm by selecting the **Alarming** tab and clicking the green plus sign + to add an alarm.
3. Complete the appropriate properties, for example, enter the **Name** of the alarm.
4. From the **Mode** section drop-down, select **Bit State** and enter a value for the **Bit Position**.
5. Repeat the same process for the remainder of the bits.

Alarms in Transaction Groups

Alarms can also be added to OPC items in Transaction Groups. This means alarms can be used without ever creating a Tag in Ignition. Simply edit an OPC item, and an Alarming section will appear

on the side of the Edit Group Tag window. From here, adding an alarm to the item is similar to adding an alarm to an Ignition Tag.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Dynamic Alarm Attributes



INDUCTIVE
UNIVERSITY

Dynamic Alarm Setpoints

[Watch the video](#)

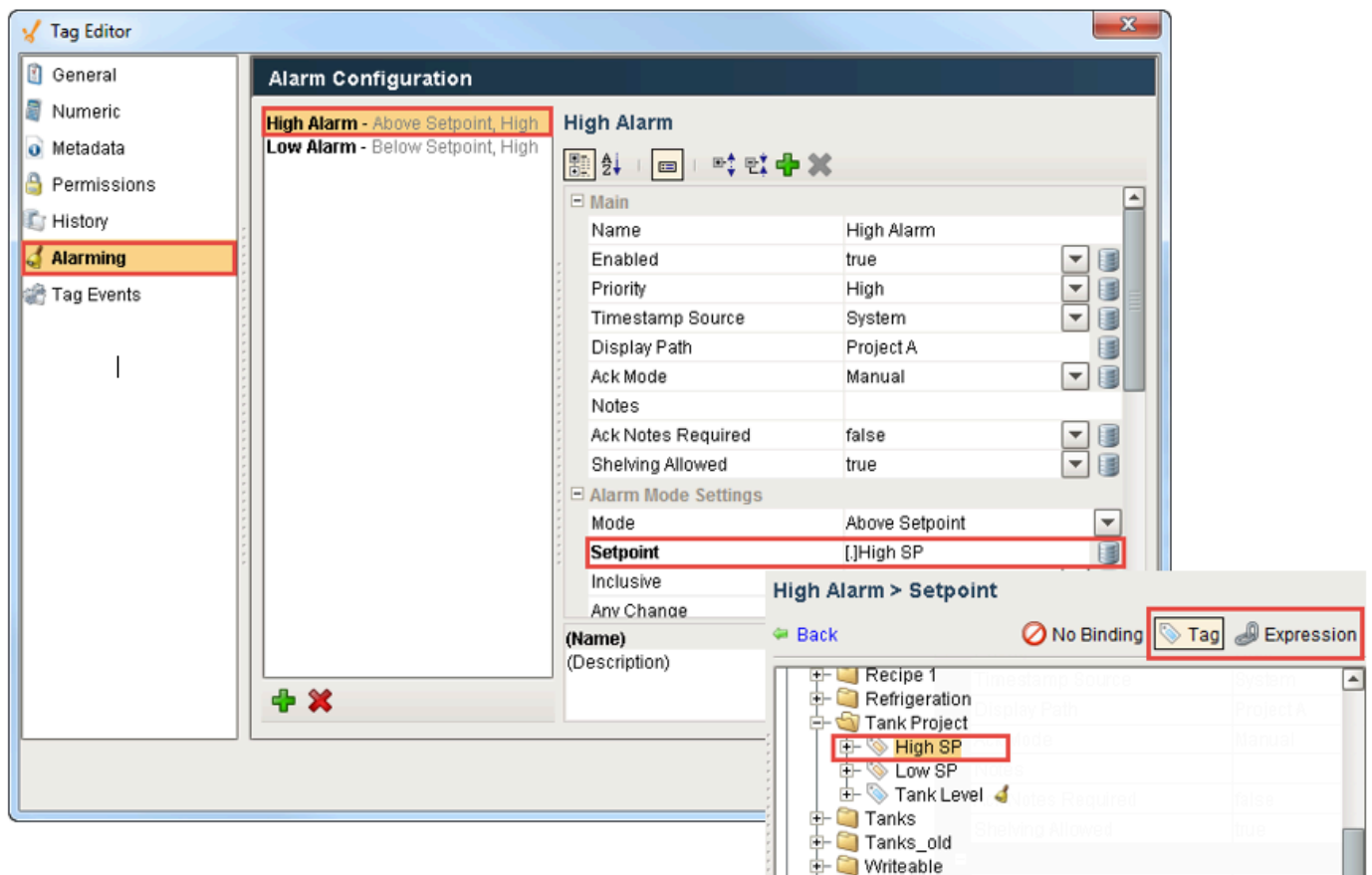
Dynamic Setpoints

You can configure alarms with dynamic setpoints inside of a Tag definition. It is similar to configuring an alarm, however, configuring an alarm with a dynamic setpoint requires additional Tags to serve as these setpoints. A good example of this is when an operator changes a high or low setpoint, it also changes how the alarm is evaluated.

To Configure Dynamic Setpoints on an Alarm

Suppose you want to alarm a tank level if it goes above or below a certain setpoint. Rather than hardcoding a value in the high and low Setpoint properties, you can bind them to either a Tag in the system or an Expression to make them dynamic. This example uses a Tag.

1. Double-click on your Tag to edit it. The **Tag Editor** window is displayed.
2. Select the **Alarming** tab and the alarm that you want to edit to associate a setpoint with. (This example uses the High Alarm).
3. Find the **Setpoint** property, and click on the binding icon next to it. Select a **Tag** or **Expression**. Bind the **Setpoint** property to a different Tag (we created a Memory Tag called High SP). Click the **Back** link in the upper left to submit your changes.



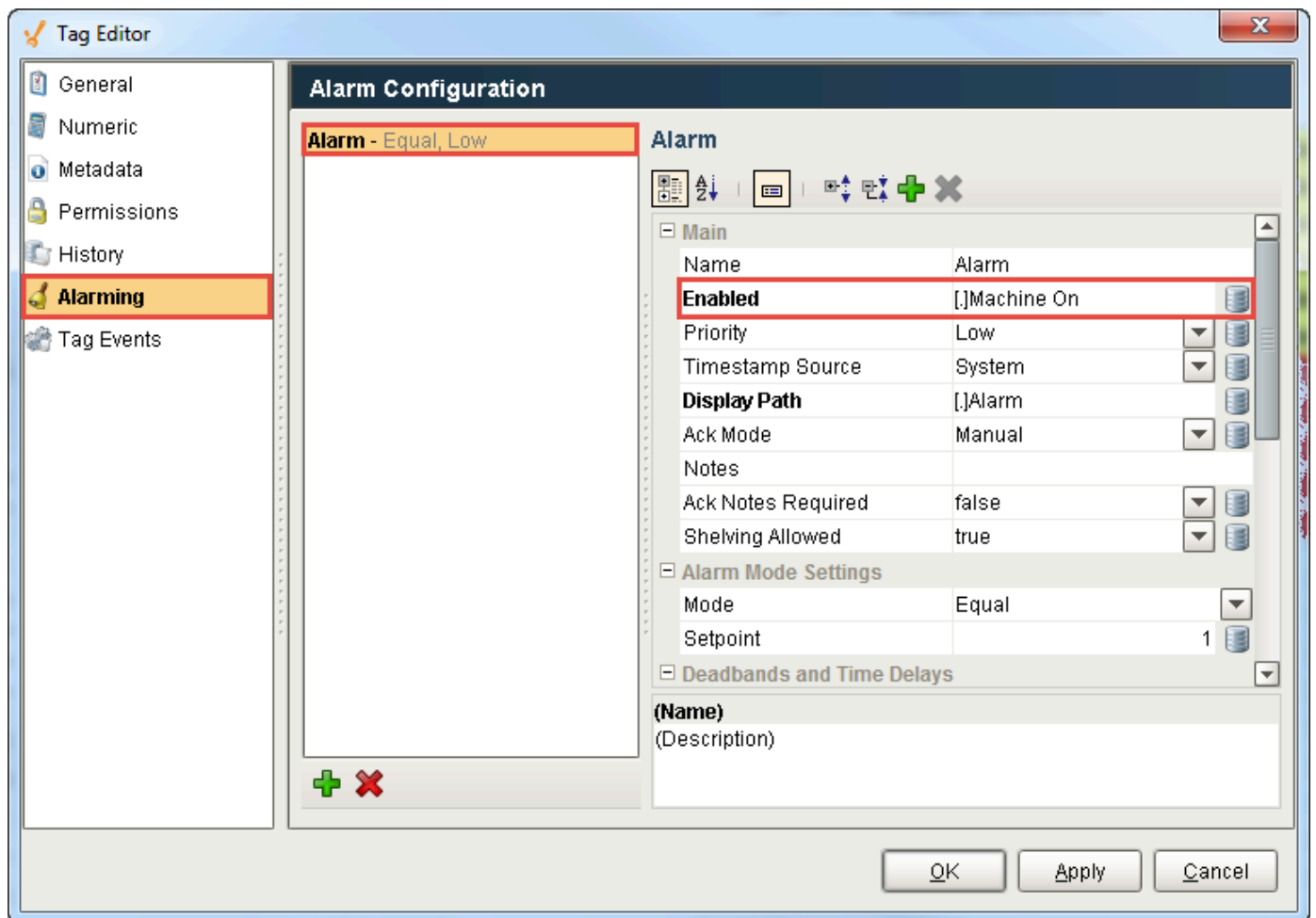
4. Repeat steps 2 and 3 for the Low Alarm, selecting Low SP for the different Tag.
5. Click **OK**. Now, the tank alarm will be evaluated based upon the high and low setpoints of each Tag.

Dynamic Enabling and Disabling

Enabling Based on Machine State

Allowing a dynamic condition to determine if an alarm is enabled or disabled is possible inside the alarm's properties; i.e. like whether or not a machine is on. This example uses two Tags; one for the Alarm, and one to determine the running state of the machine. Two conditions must be true for the alarm in the example below to become active: the value on the Tag must match the setpoint ('True' in the example below), and the Enabled property must evaluate to 'True' (driven by the value of the 'Machine On' tag).

1. Select a **Tag** that has the alarm you want to configure.
2. Access the alarm properties by double-clicking on the **Tag**, selecting the **Alarming** tab and the alarm that you want to associated with this Tag.
3. Find the **Enabled** property and bind it to a **Boolean Tag** or an **Expression** that evaluates a 'true' or 'false' condition. (This example uses a Tag called Machine On).



4. Click **OK**. Now, the alarm will only be evaluated based upon the state of the Machine On Tag.

Enabling Based on Time of Day

There are occasions when you only want alarms to be evaluated at certain times of the day. In Ignition, you can automatically enable and disable alarms for specific times of the day. This is typically achieved by binding the alarm's Enabled property to a Tag, or Expression. (This example uses an Expression).

1. From the **Tag Browser**, select the **Tag** on which you want to configure the alarm.
2. Double-click on the **Tag**, and in the Tag Editor window go to the **Alarming** tab. Click the alarm you want to place the binding on.
3. Bind the **Enabled** property to an expression that evaluates a 'true' or 'false' condition based upon the current time. Click the bind button to the right of the Enabled property, and click on **Expression** tab on the top right of the screen, and enter your expression. In the expression language, there are various functions for dates that can be used. For example, the following expression would return 'true' if the time is between the hours of 5pm and 12pm, and return 'false' if it is not.

```
timeBetween(now(0), "5:00:00 pm", "12:00:00 pm")
```

4. Click **OK**. The alarm will only be evaluated when the system clock falls between the specified time.

Tag Editor

Alarm Configuration

Alarm

Alarm - Equal, High
Alarm 1 - Equal, Low

Main

Name	Alarm
Enabled	timeBetween(now(0), "5:00:00 pm", "12:00:00 pm")
Priority	High
Timestamp Source	System
Display Path	[.]
Ack Mode	Manual
Notes	
Ack Notes Required	false
Shelving Allowed	true

Alarm Mode Settings

Mode	Equal
Setpoint	0

Deadbands and Time Delays

Enabled
Specifies whether or not this alarm is evaluated by the system.

OK **Apply** **Cancel**

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarms in UDTs



INDUCTIVE
UNIVERSITY

Configure Alarm in UDT

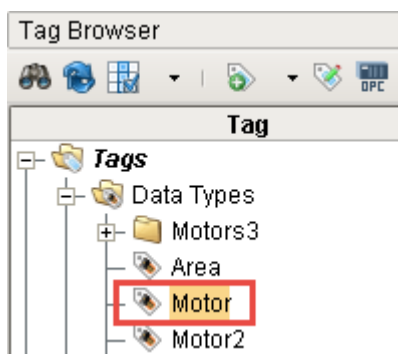
[Watch the video](#)

Configuring an Alarm on a UDT Member

The great thing about UDTs is that you configure it in one place, inside of the UDT definition, and every instance of that UDT will automatically inherit that same configuration. The same concept works for alarms on a UDT. If an alarm is configured inside a UDT, every instance of that UDT will automatically have that same alarm configuration. Even if a new instance is created, it will automatically get that same alarm configuration. Let's get started.

To Configure an Alarm on a UDT

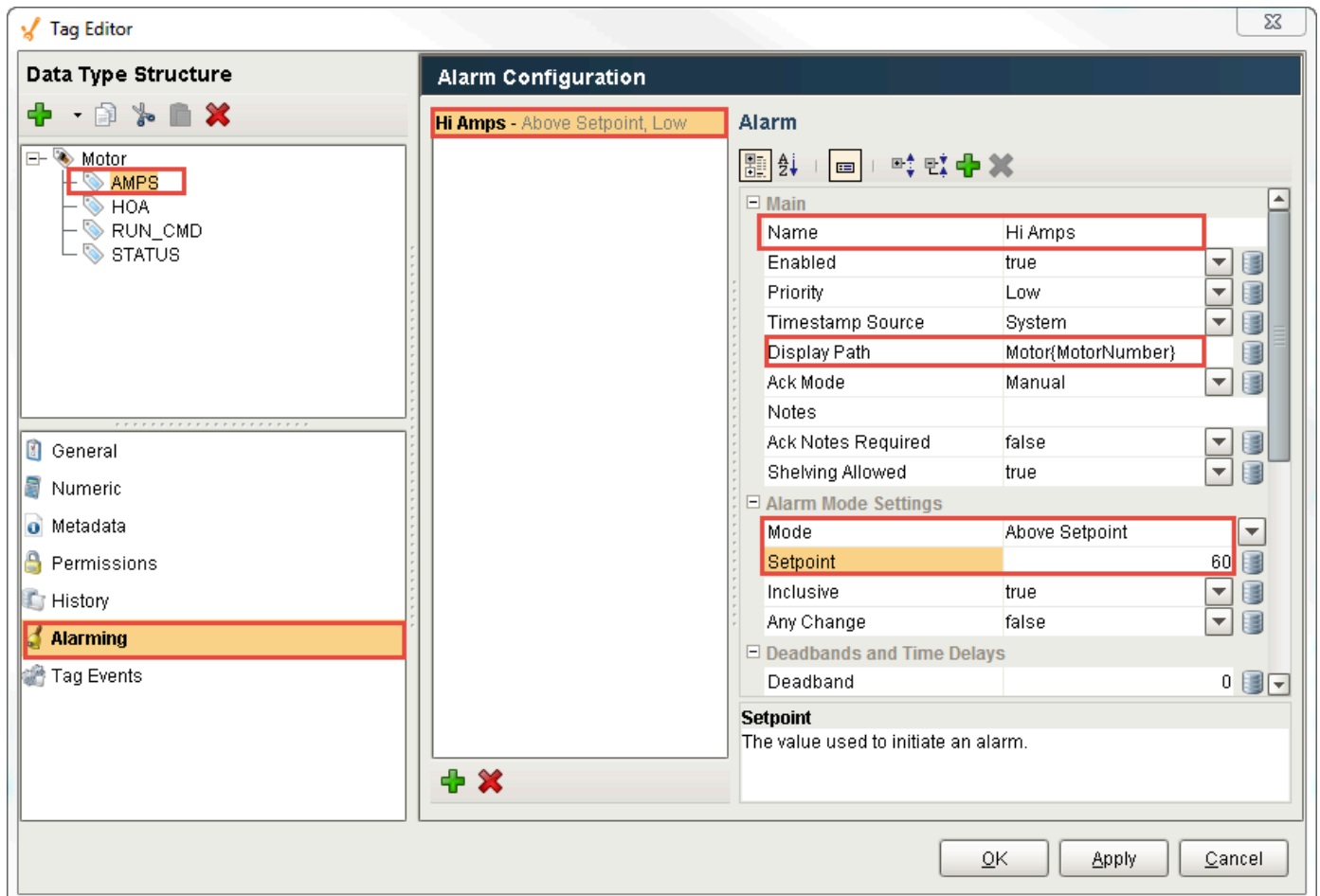
1. In your **Tag Browser**, go to the **Data Types** folder, and double click on a UDT to edit the definition. This example uses a Motor UDT.



2. Select a Tag, and click on the **Alarming** tab to add an alarm inside the UDT definition. Enter an **Alarm Name**, **Display Path**, **Mode**, and **Setpoint**. UDT configurations are set up the same way as the normal [alarm configuration on a tag](#).

DISPLAY PATH

Since you have multiple instances of a Motor, as in this example, you need to make sure that the Display Path is unique for every instance, otherwise, every instance of the UDT will have the same Display Path name, and the operator won't know which alarm to respond too. There are a couple of ways to address this: you can leave the Display Path blank, in which case, Ignition will use the full Display Path to the instance of that tag, or you can bind a property to an expression involving a UDT parameter (i.e., `Motor` + `{MotorNumber}`).



3. Click **OK**. Now, all instances of the UDT will have a similar alarm setup, but each alarm will be unique to that UDT instance.

Dynamic Setpoints in UDTs



INDUCTIVE
UNIVERSITY

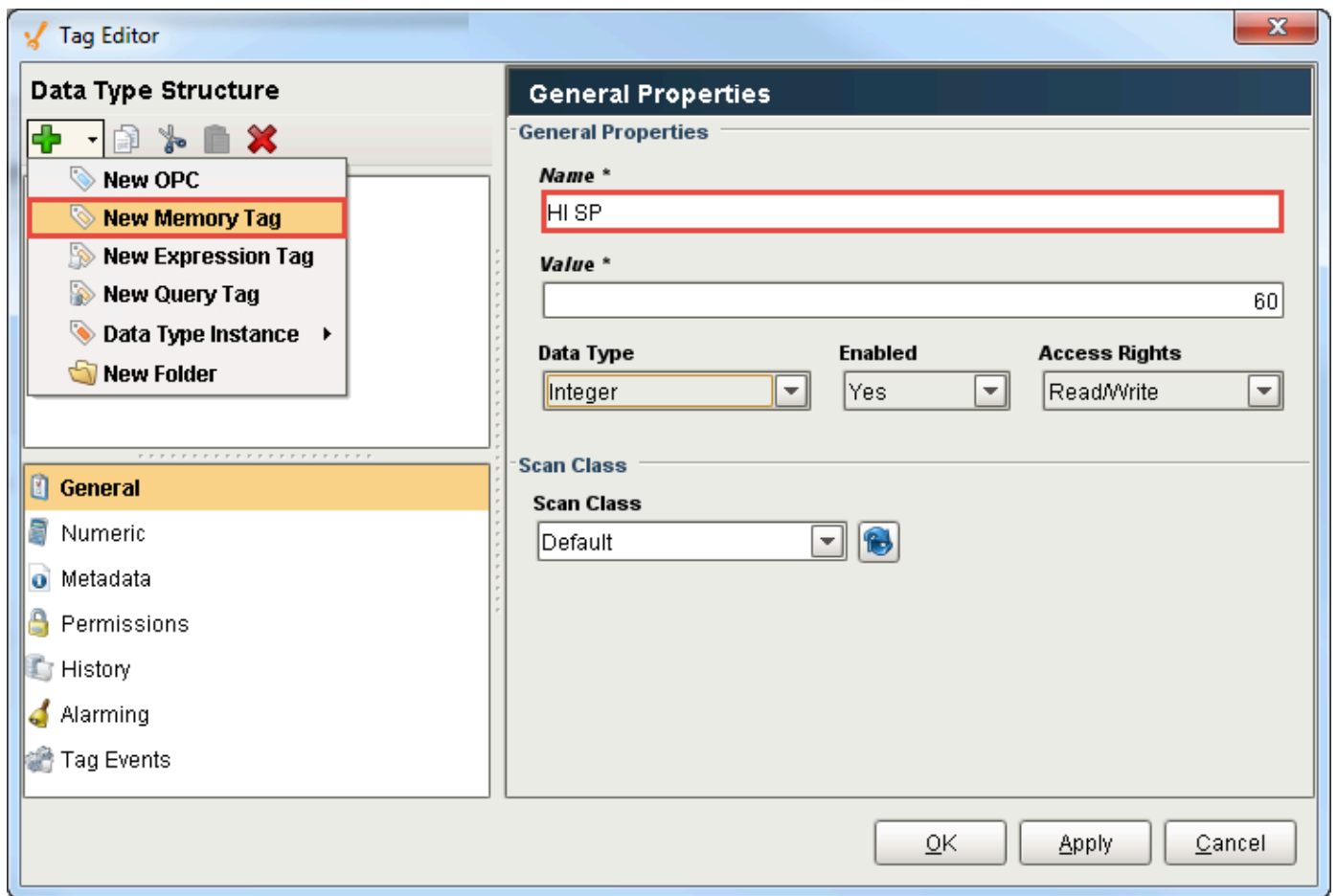
UDT Alarm Dynamic Setpoints

[Watch the video](#)

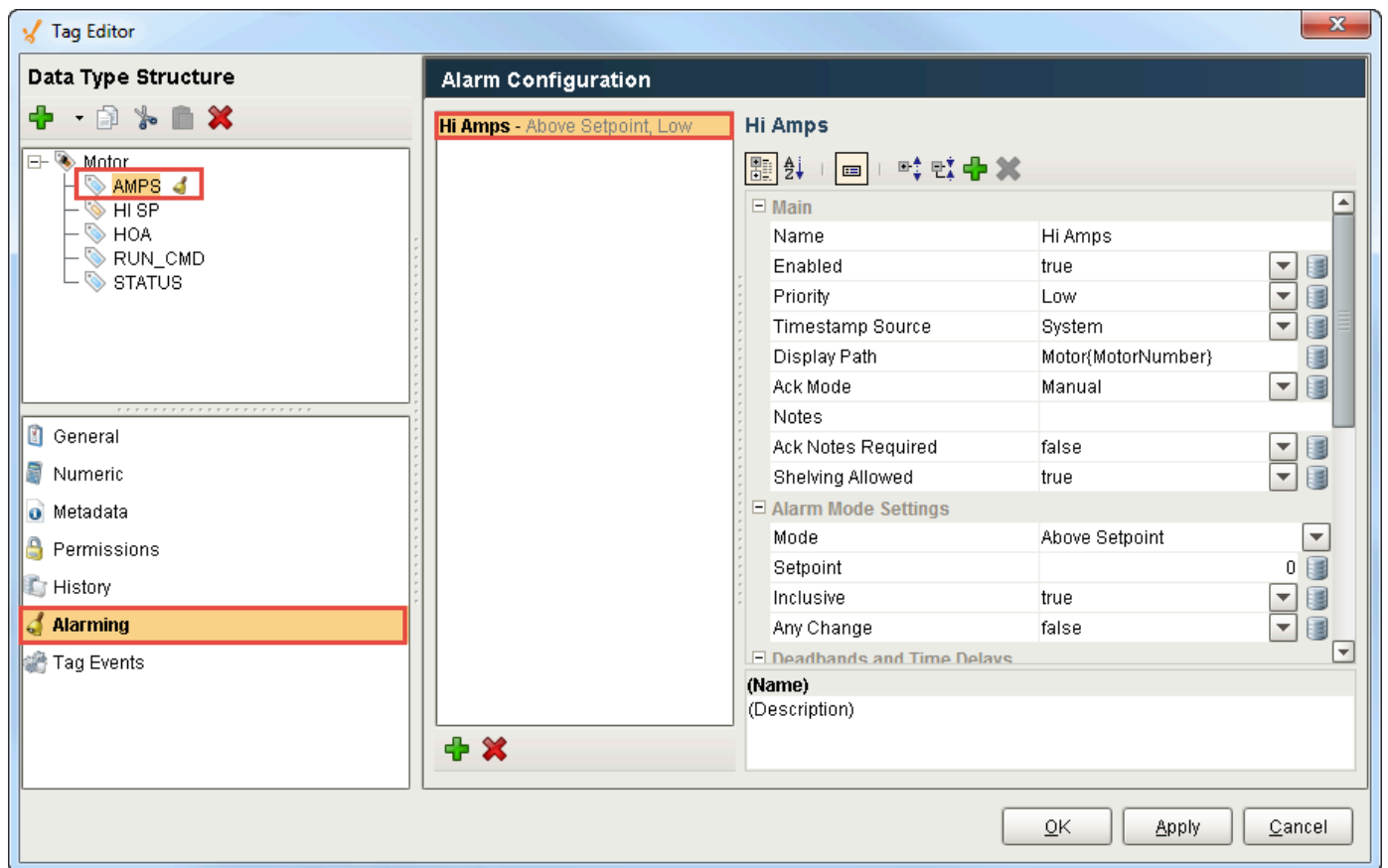
Instead of using a hardcoded setpoint, you can configure alarms with dynamic setpoints inside of a UDT definition. It is similar to configuring a UDT alarm, however, configuring a UDT alarm with dynamic setpoint requires additional Tags to serve as these setpoints in all the deployed UDTs.

To Configure Dynamic Setpoints inside a UDT

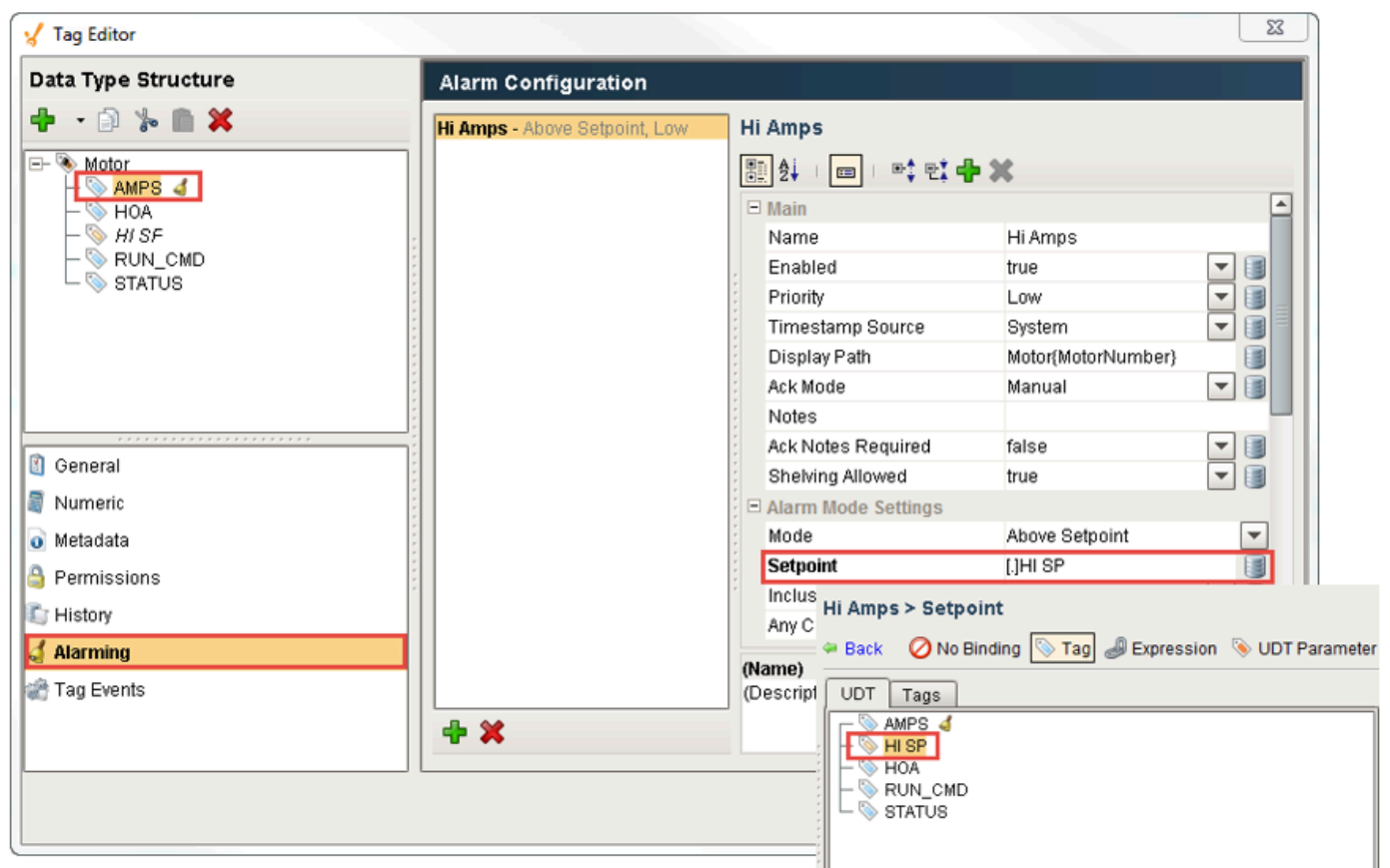
1. In the **Tag Browser**, go to your **Data Types** folder. Click on your UDT to edit the definition. Let's use the Motor UDT example again.
2. Create a **Memory Tag** inside the UDT definition to serve as the setpoint, and give it a name (i.e., HI SP).



3. Select the Tag (i.e., AMPS), and the alarm inside the UDT definition (i.e., Hi Amps) that has the alarm that you want to associate with the UDT.



4. Bind the **Setpoint** property to the new UDT Memory Tags (i.e., HI SP).



5. Click **OK**. All of the deployed UDTs will now have Memory Tags that serve as the alarm setpoints.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Associated Data



INDUCTIVE
UNIVERSITY

Alarm Associated Data

[Watch the video](#)

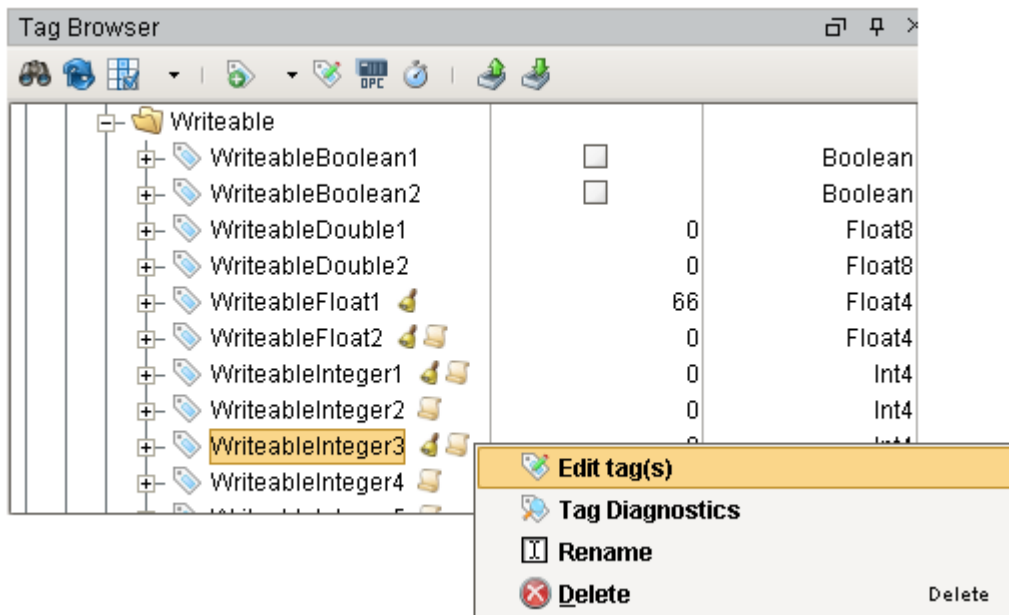
What is Alarm Associated Data

Every alarm in Ignition has alarm properties associated with it as it moves throughout the system, like active time, clear time, who acknowledged the alarm, time the alarm was acknowledged, priority, display path, and much more. You can extend the list of alarm properties by adding your own associated data or custom properties to an alarm you already have configured. Values can either be static or dynamic. Static properties are great for filtering alarms. Dynamic properties will often be bound to other Tags or an expression. Regardless of whether a value is static or dynamic, these values will be attached to the alarm event as it moves through the system, and the values will be available from the [Alarm Status](#) system, the [Alarm Journal](#) system, and the [Alarm Notification](#) system.

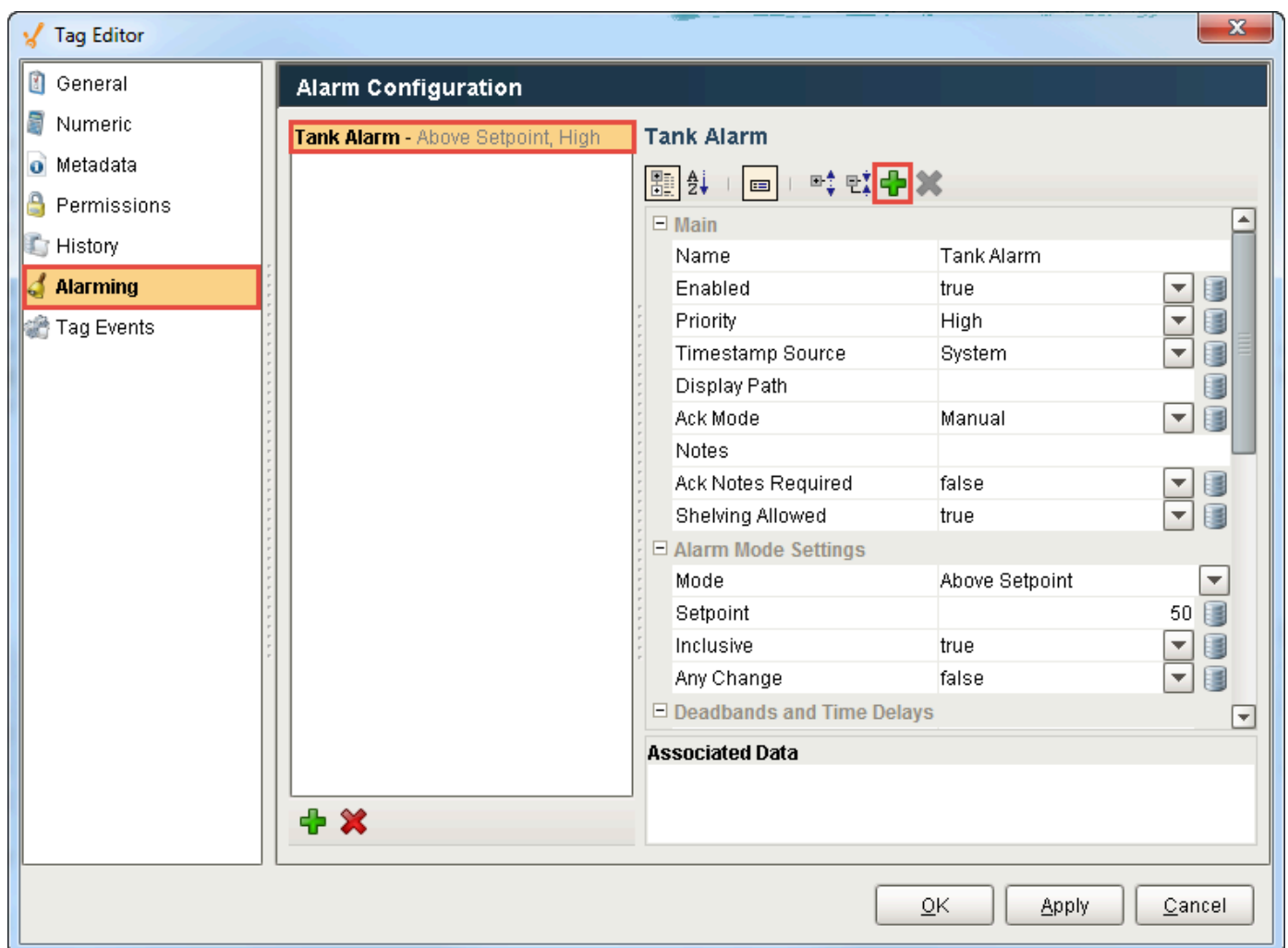
It's easy to go back to an existing alarm and add associated data. Let's give it a try!

Creating Associated Data for Both Static and Dynamic Properties

1. Right-click on your Tag and select **Edit tag(s)**. This example uses an OPC Tag, but any alarm-capable Tag will suffice.

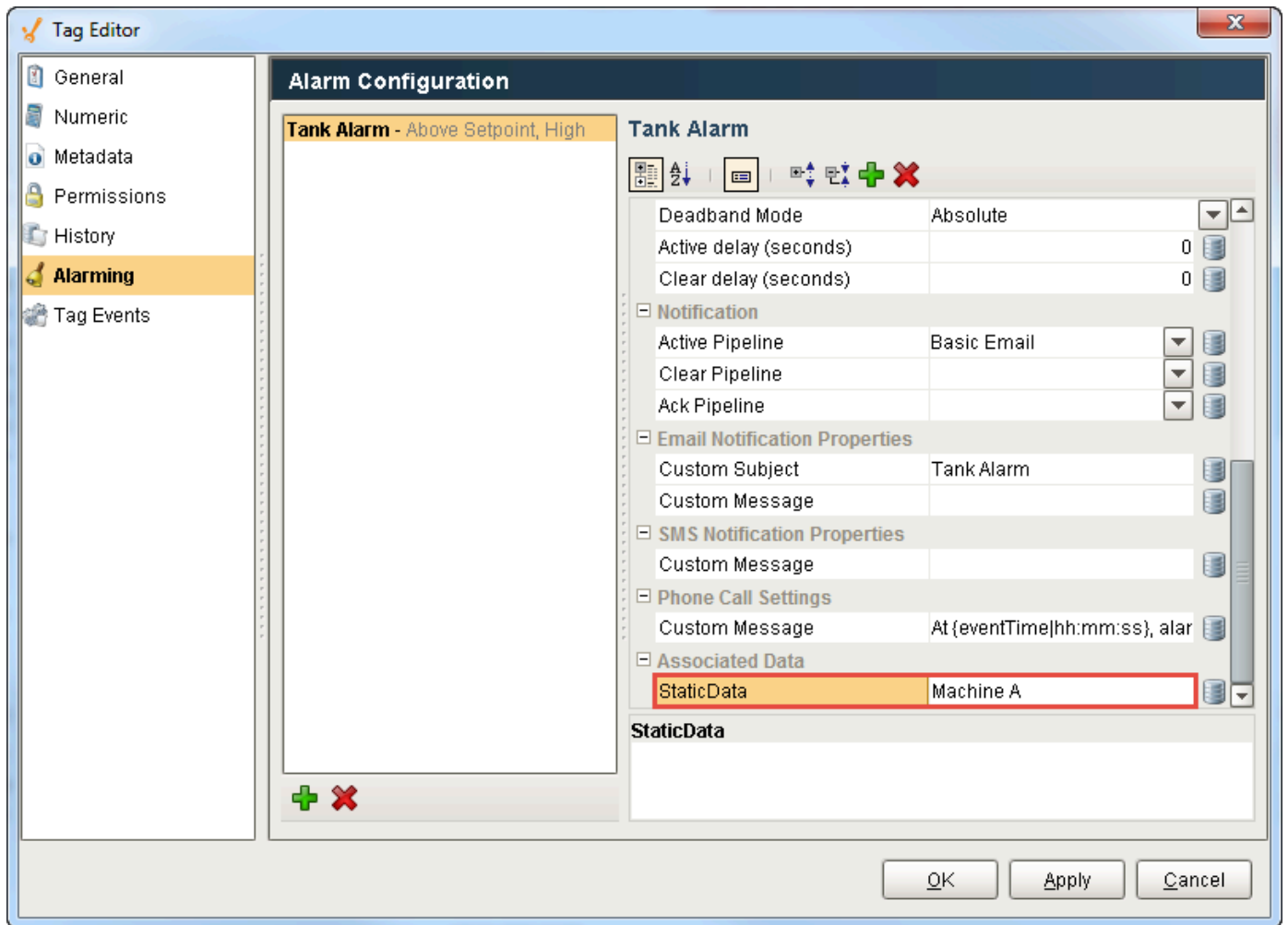


- In the **Tag Editor**, go to the **Alarming** tab, select an alarm. The Tag below already has an alarm named 'Tank Alarm', but you can **create a new alarm** for this example. Click the green plus sign + above the alarm properties to add new associated data for the alarm.

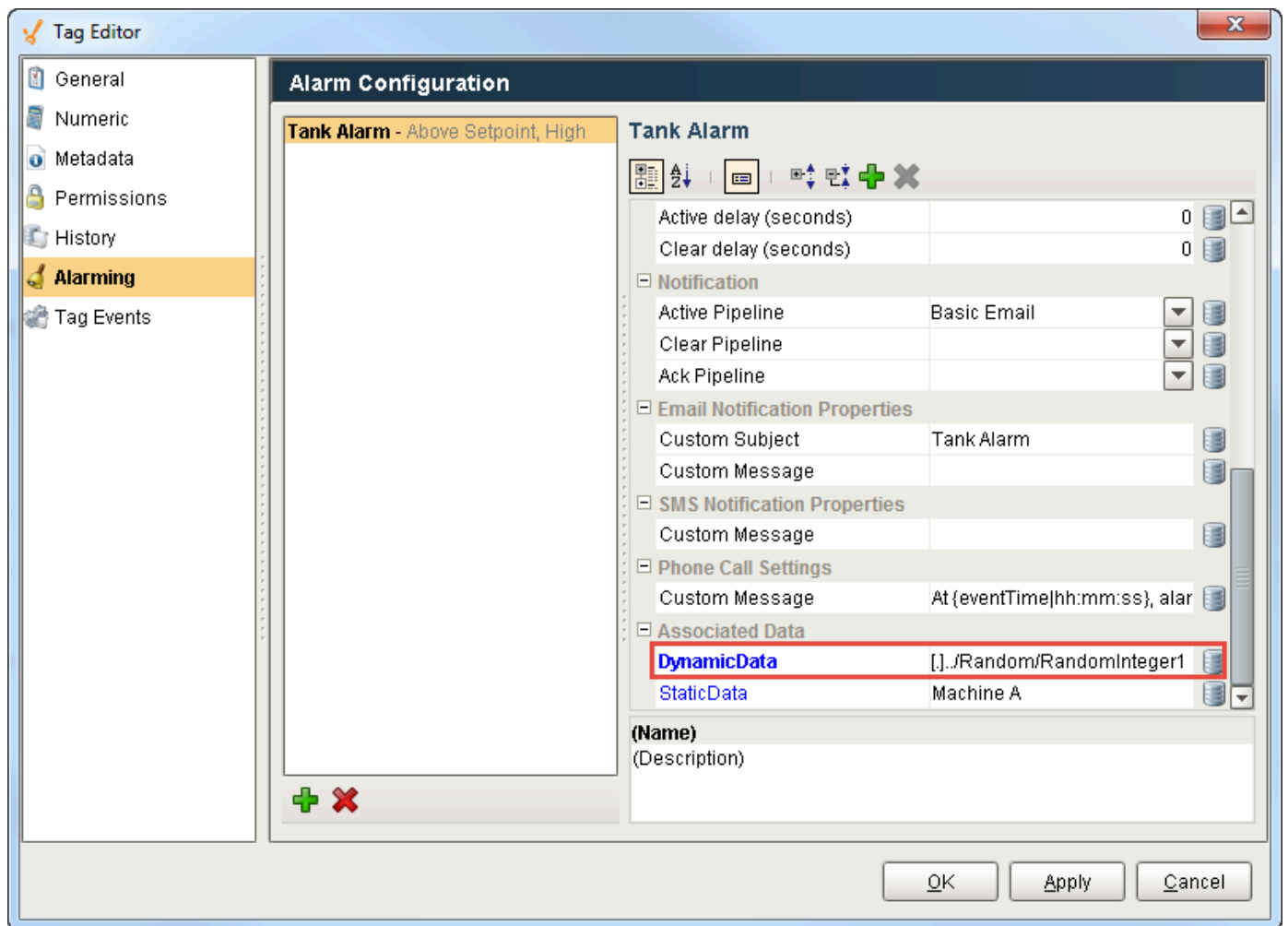


- Scroll to the bottom of the property list, and you'll see a new associated data property was added. By default, the new property name is called '**New Data**'. Double-click the property name to rename it to something more meaningful, and press enter to submit your new name. In this

example, make the associated data property a static value. Rename it to '**StaticData**' with a value of '**Machine A**'.



- Now, let's create another Associated Data property with a dynamic value. Click the green plus sign + above the alarm properties. Scroll down to the bottom of the property list and rename your '**New Data**' property to '**DynamicData**' and bind it to a Tag using the binding icon on the right side of the window (this example is using the RandomInteger1 tag from our Simulator). Click the '**Back**' button in the upper left to submit the Tag selection.



5. Click **OK**. Now, when the alarm goes active, Ignition will store the values of the associated data with the alarm. You can view the details of the alarm in the Alarm Status Component along with the new associated data properties. To view the **Details** of the alarm, check the box of the alarm you want to see, and click on the magnifying glass at the bottom of the window.

☐	Active Time	Name	Ack Time	Ack'd By	Current State	Priority
☐	10/31/16 8:32 AM	Low Alarm			Active, Unacknowledged	Critical
☐	11/3/16 8:29 AM	High Alarm			Active, Unacknowledged	Critical
☐	11/3/16 11:23 AM	Fault			Active, Unacknowledged	Critical
☐	10/31/16 8:32 AM	Low Alarm			Active, Unacknowledged	High
☒	11/3/16 12:29 PM	Tank Alarm			Active, Unacknowledged	High
☐	11/3/16 12:29 PM	Low Alarm 1			Active, Unacknowledged	High
☐	10/31/16 8:32 AM	:pw Alarm			Active, Unacknowledged	Medium
☐	10/31/16 8:32 AM	Low Alarm			Active, Unacknowledged	Medium
☐		Details	Notes		Low Alarm	Active, Unacknowledged Medium
mode setpointA Event Value voip.customMessage activePipeline DynamicData name CustomEmailSubject Event Time StaticData priority Active Setpoint 12 66 At {eventTime}hh:mm:ss, alarm named {name} became {eve... Basic Email 12 Tank Alarm Tank Alarm 11/3/16 12:30 PM Machine A High						
Acknowledge Shelve						

Alarm Grouping

Grouping alarms in Ignition is an important concept. Instead of seeing every alarm in a single view, often times, you may want to view alarms for a particular area of your plant, or for a specific set of alarms.

There are a several ways to group alarms. One way is to use a folder structure which requires you to organize your Tags into a hierarchy. Another way is to use the Display Path field in the alarm configuration. The third way, and the most recommended way, is to use associated data. It's a common design practice to associate alarm groupings on the associated data of the alarm.



INDUCTIVE
UNIVERSITY

Alarm Grouping

[Watch the video](#)

Creating an Alarm Grouping

1. Use an existing alarm that you already have configured. If you don't have one, [create a new alarm](#).

2. Right-click on your **Tag** and select **Edit tag(s)**. (This example uses the WriteableFloat1 Tag). In the **Tag Editor**, go to the **Alarming** tab, select an alarm (i.e, High Alarm), and click the green plus sign + above the alarm properties to add new associated data for the alarm.
3. Scroll to the bottom of the property list, and you'll see a new associated data property was added. In this example, make the associated data property a static value, and rename '**New Data**' to '**Group**' with a value of '**Group A**'. The associated data will have a static value that describes the Group.

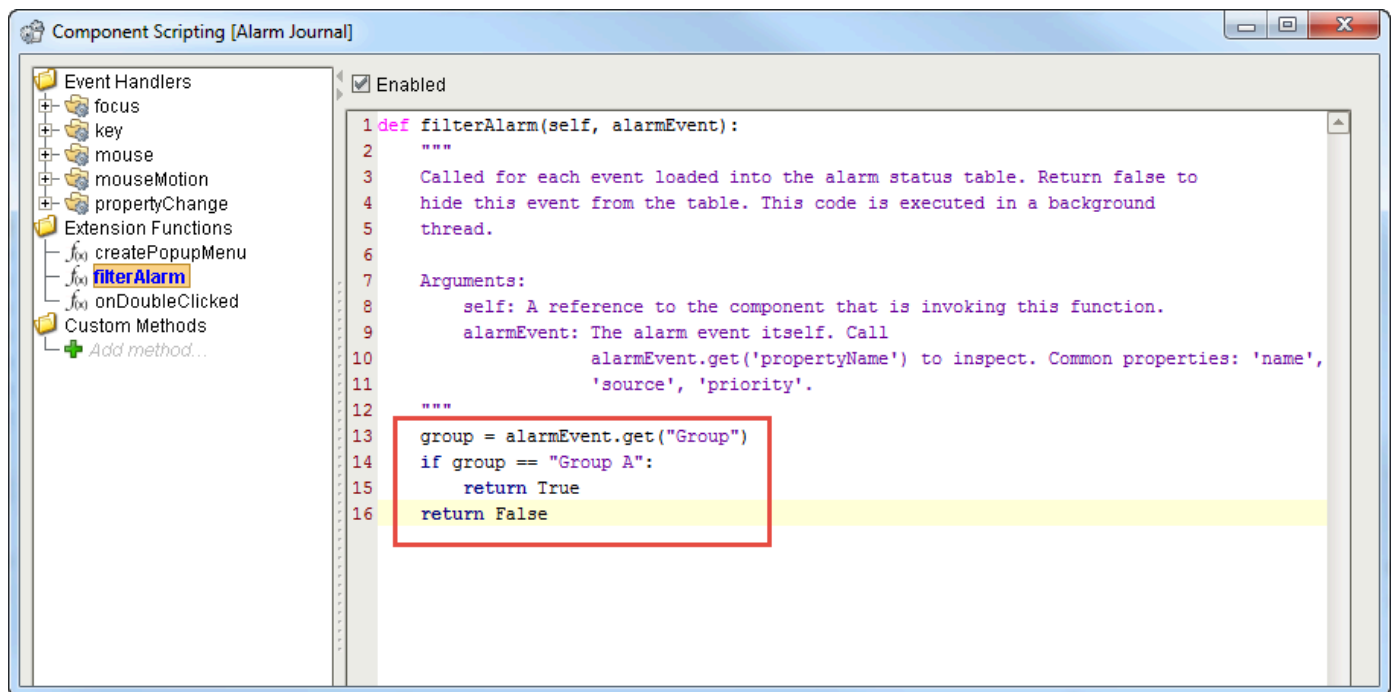


4. Click **OK**. The alarm will now always have the '**Group*8' designation associated with it. This can be used for filtering or part of your alarm pipeline notifications.
5. Before you can use the Alarm Journal to filter on your new associated data property, edit the '**filterAlarm**' Extension Function to include your new associated data property.

Select the **Alarm Journal component**, scroll down and right click on **Scripting**. Under **Extension Functions**, click on **filterAlarm**. Click on the **Enabled** checkbox to enable the script. Next, edit the script to add your new associated property and property value. Refer to the new code below to add to the filterAlarm script

filterAlarm Scripting Function

```
group = alarmEvent.get("Group")
if group == "Group A":
    return True
return False
```



6. Now, you're good to go! Open the Alarm Journal Component and filter on your associated data.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Event Properties Reference

Alarm Event properties provide a lot of information regarding an alarm event and fall into several categories:

- General Alarm properties
- Event Alarm properties
- Runtime Event properties

General Alarm properties are automatically configured with default settings when the module is first installed. Event Alarm properties are properties that occur when an event happens. Runtime Event Properties are properties that are only present while the alarm event is in memory.

Alarm Event properties are described below. If you are looking for information on the properties associated with configuring an alarm, please see the [Tag Properties](#) page.

General Alarm Properties

General alarm properties are configured during initial setup.

1. Go to the **Configure** section of the Gateway webpage, and choose **Alarming** > **General** from the menu on the left. The General Alarms Settings page will appear.
2. Set the general alarm properties shown in the following table:

Alarm Evaluation

Property	Description
Live Event Limit	Default is 5. The number of "live" events (active or unacknowledged) that can exist for a single alarm at a given time. When surpassed, older events will be acknowledged automatically by the system. This means as an alarm cycles on and off, Ignition will keep track of the last 5 times the alarm event happened until the user acknowledges them. This does not store history for those events.

Event Suppression

Property	Description
Continuous Event Detection Window (min)	Default is 10. The amount of time to store events before shutdown to prevent new duplicate events from being created on startup. This setting prevents unacknowledged active events from being generated due to reboot. If set to 0, will not be used.
Notify Initial Events	Default is false. If false, active alarms caused by the "initial state" (that is, the first value checked after being created, or after the enabled state changes) won't be sent to the notification system. This means if you add an alarm to a Tag, a notification won't be immediately sent when the new state is created.

Alarm Tag Properties

All alarm states have a list of properties that you can edit or bind when creating or editing a Tag. These are useful to know when using the `system.tag.addTag()` and `system.tag.editTag()` functions. To see the list of alarm properties that can be configured on a Tag, see the [Tag Properties](#) page.

Alarm Event Properties

An alarm event is made up of many pieces of information. The state, the value, the time, all of the configuration data for the alarm, and more are collectively known as the properties of the event. A "property" is generally a name, potentially along with a default value. There are many properties that an alarm may have, and if the property isn't present, the default value is usually assumed.

This is different than the alarm properties used during initial setup, these are the properties available when an alarm event occurs (active, clear, acknowledged).

How Properties are Used

Properties are used in a number of ways:

- Define how an alarm behaves.
- Are referenced in messages, such as the body of an Email. or SMS message.
- Are referenced in Expressions, such as in the binding of a different property, or in an Expression block.
- Are created and used in **pipelines** as temporary variables, such as to make a counter.
- Various other places.

While properties are always accessed the same (for example by using the `{propertyName}` syntax in a message or Expression, or the `getProperty()` Expression function), there are some subtle variations on where the property value comes from, and what its lifecycle will be. To understand this better, consider that an Alarm Event has the following structure:

Alarm Event

Active Event Properties

Clear Event Properties

Acknowledged Event Properties

Runtime Properties

The Active Event, Clear Event, and Acknowledge Event properties get created when the described type of event occurs. Consequently, it's possible for the same property to exist multiple times in an Alarm Event. For example, a bound configuration property or associated data is captured on both the active and clear events. When you reference a property, the alarm event will provide you with the most recent value. So, if you have a bound property called "MyData", and you reference it when the alarm is active, you may get a different result than when the reference is executed later and the alarm has cleared. The individual values are stored separately in the **Alarm Journal**, however.

Alarm Event Properties

Event Properties	Binding/Scripting Name	Description	Datatype
Name	name	The name of the Alarm	String
Enabled	enabled	Specifies whether or not this alarm is evaluated by the system. Set to False to turn off the alarm state and all associated actions.	Boolean
Priority	priority	The priority (or severity) of the alarm. Used for sorting/filtering. Numerical values are associated with each priority to make comparison easier. This property can also be referenced as a string with the following priority names. - Diagnostic (0) - Low (1) - Medium (2) - High (3) - Critical (4)	Integer or String
Display Path	displayPath	The unique path of the Alarm state. Used for display and browsing purposes.	String
Active Pipeline	activePipeline	The pipeline (if any) that will be used to process active events generated by the alarm.	String
Clear Pipeline	clearPipeline	The pipeline (if any) that will be used to clear events generated by the alarm. Used when the alarm goes into the Clear State.	String

Event Properties	Binding/Scripting Name	Description	Datatype
Active Delay	timeOnDelaySeconds	The amount of consecutive seconds that the alarm state must be True before the Tag enters this alarm state.	Double
Clear Delay	timeOffDelaySeconds	The amount of consecutive seconds that the alarm state must be False before the Tag exits this alarm state.	Double
Notes	notes	Free-form notes for the alarm state.	String

Alarm Runtime Properties

As described in the previous section, Runtime Properties are different in that they only exist while the alarm event is in memory (still "live", that is, not cleared, or not acknowledged). They do not get stored in the **Alarm Journal** by default. In addition to properties created through the **Set Property Block**, the system also has a number of defined Runtime Properties that it may use for various purposes. Though these are used internally, they are technically still regular properties, and can be accessed and modified through the normal means.

Runtime Property	Description	Datatype
IsInitialEvent	Set to "true" when the event is caused by the initial state of the alarm.	Boolean
SystemAck	Set to "true" when the alarm has been acknowledged by the system, due to an overflow of the "live event queue". Live events are alarm events that are active or not acknowledged, and are limited for each alarm by the general alarm settings.	Boolean
ShelfExpiration	When the shelf will expire for this event.	Integer
IsShelved	Is the alarm currently shelved?	Boolean
EventCanceled	If set, the event will drop out as soon as possible from the pipelines.	Boolean
EventId	The unique id (uuid) of this alarm event. Each event gets a completely unique id.	String
Source or Source Path	The qualified path to the item that generated this event. Includes the Tag Provider, Tag Path, and the name of the alarm. Example: prov:tagProviderName:/tag:folder/tagName:/alm:alarmName	String
DisplayPathOrSource	Gets the display path if defined, otherwise, returns the source.	String
State	The current overall state of the alarm. States include: • Clear and Unacked (0) • Clear and Acked (1) • Active and Unacked (2) • Active and Acked (3)	Integer
EventState	The transitional state that caused the current event. States include: • Active (0) • Clear (1)	Integer

Runtime Property	Description	Datatype
	Acknowledged (2)	
EventValue	The value associated with the current event.	Integer
AckUser	The user who acknowledged this event.	String
IsAcked	"True" if the event has been acknowledged.	Boolean
IsActive	"True" if the event is still active.	Boolean
IsClear	"True" if the event is not still active.	Boolean
ActiveTime, ClearTime, AckTime	The timestamp for each event.	Date
PipelineTransitionCount	How many transitions the event has made inside of the pipelines.	Integer

Associated Data

You can extend alarm properties by creating your own associated data. Associated data properties can be configured on any alarms that are already configured in a project.

To learn more, refer to [Alarm Associated Data](#).

The AlarmEvent Object

Some system functions, such as [system.alarm.queryStatus](#), return an AlarmEvent object which contains many methods to retrieve additional information about the individual alarm. This section details the various methods on an AlarmEvent Object.

Note, many of the functions below return some complex object, as opposed to just a standard Python datatype. In these cases, normal Python type-casting can be used to turn the object into a native Python type. For example, **getDisplayPath()** returns a **StringPath** object, which can easily be converted to a Python string with the **str()** function.

Function	Description	Returned Object
getDisplayPath	Returns the Display Path of the alarm.	StringPath
getDisplayPathOrSource	If the Display Path for the alarm is in an empty string, then returns the Source Path, otherwise, returns the Display Path	Python Unicode
getId	Returns the UUID of the alarm event.	java.util.UUID
getLastEventState	Returns the last state of the alarm. Possible return values are Active, Acknowledged, or Cleared. The last event is always returned, so if you're looking to see if an alarm is has been both Acknowledged and Cleared, use getState() instead.	AlarmStateTransition
getName	Returns the name of the alarm.	Python Unicode
getNotes	Returns the value of the Notes property. Returns a None-type object if a Note has not been configured on the Alarm.	Python Unicode or None
getPriority	Returns an AlarmPriority object representing the Priority of the alarm. Can easily be converted to an Integer with .intValue. Default is String.	AlarmPriority
getSource	Returns the Source path of the alarm.	QualifiedPath
getState	Returns the current state of the alarm: i.e., Active, Unacknowledged	AlarmState
isAked	Returns a boolean flag indicating that the alarm has been Acknowledged.	Python Boolean
isCleared	Returns a boolean flag indicating that the alarm has been Cleared.	Python Boolean

Function	Description	Returned Object
isShelved	Returns a boolean flag indicating that the alarm has been Shelved.	Python Boolean

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status

Alarm Status Table

Once alarms have been configured, most users will want to view the status of alarms from within a client. Fortunately, alarms can be visualized in several ways.

By far the most popular way to visualize alarm status is with the Alarm Status Table component. This component immediately displays the status of all alarms in the system from all Tag providers. However the component can also filter down to specific alarm criteria. For example, the component can filter by source path which means a single Tag provider, folder in a Tag path, or even UDT instances can be focused on by the component.

The default look of the component is designed to make active alarms easy to spot, but the pallet can be customize to fit your company standard.

Display Path Versus Source Path

When displaying alarms on the Alarm Status Table component, it is common to filter on either the Display Path, or the Source Path.

The **Display Path** can be customized on each alarm. The default value for an alarm's Display Path is a Tag path that leads to the name of the alarm. The Integer Tag example above has an alarm named 'Alarm', and is located on a Tag path of 'Alarming Example/Integer Tag', thus the Display path will resolve to '**Alarming Example/Integer Tag/Alarm**'. However, the Display Path can be customized when configuring the alarm. This is generally utilized to display readable messages as to what the issue is: i.e., "Tank 105 High Temp Alarm".

The **Source Path** is also a path to the alarm, but also notes the Tag provider the alarm is located in. Again using the Tag above, if the name of the Tag provider is 'default', then the source path would resolve to 'prov:**default**:/Tag:**Alarming example/Integer Tag**:/alm:**Alarm**'. Unlike the Display Path, the Source Path on an alarm can never be overridden.

Display Path	Source Path
☐ Alarming Example/Integer Tag/Alarm	prov.default:/tag:Alarming Example/Integer Tag:/alm:Alarm
☐ Alarming Example/Integer Tag/Alarm	prov.default:/tag:Alarming Example/Integer Tag:/alm:Alarm
☐ Alarming Example/Integer Tag/Alarm	prov.default:/tag:Alarming Example/Integer Tag:/alm:Alarm

Understanding the functionality of Display Paths and Source Paths allows for much flexibility when filtering alarms by Tag path.

Display the Status of a Single Alarm



INDUCTIVE
UNIVERSITY

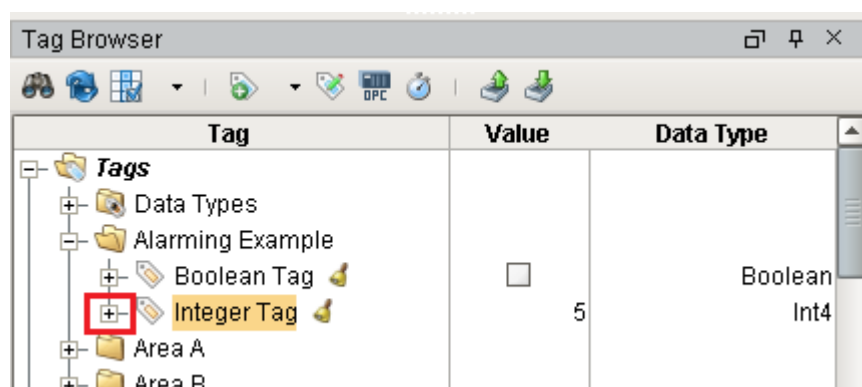
Display Status of Single Alarm

[Watch the video](#)

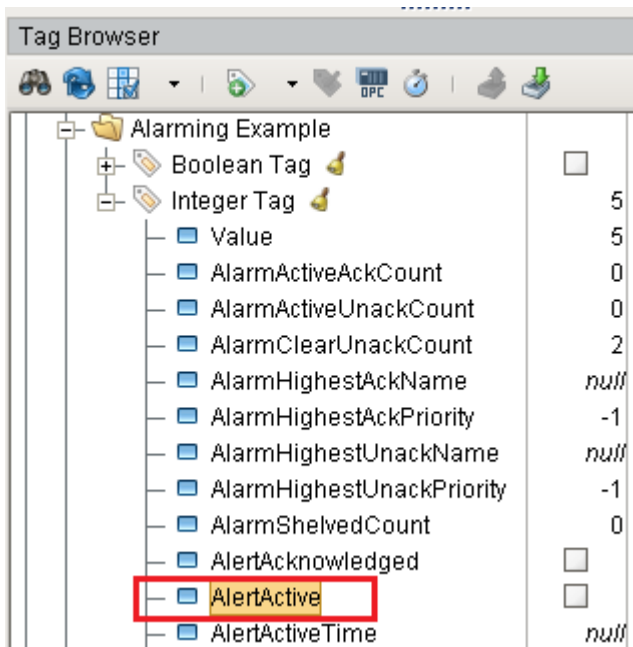
Each Tag has several **Tag properties** that track the status of all their alarms. These properties can be used in **Property Bindings** to visualize the state of alarms on a specific Tag.

Example

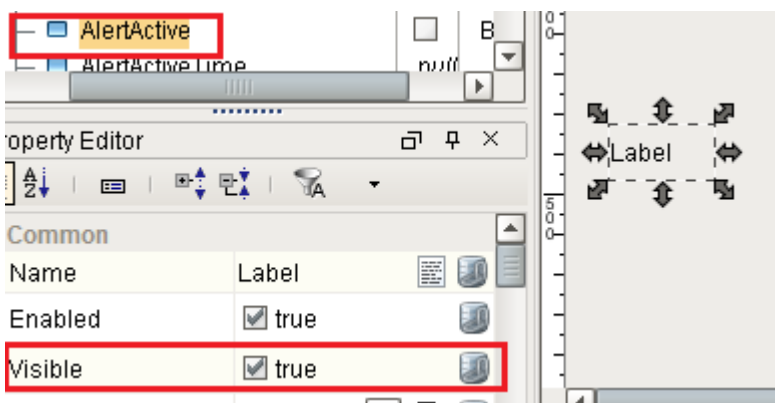
1. In the Designer, create a new **Label** component on an empty window.
2. Create a memory Tag. In this example, we named the Tag "Integer Tag" and set the datatype to Integer. The alarm is configured to become active on a value of 1.
3. Once the Tag has been created, click on the '+' icon next to the Tag in the Tag browser.



This will expand the Tag and show all of the Tag properties, including the **AlertActive** property.



4. With the **Label** component selected, drag and drop the **AlertActive** property from the Tag to the **Visible** property on the Text Field component. This will create a binding directly to that Tag property.



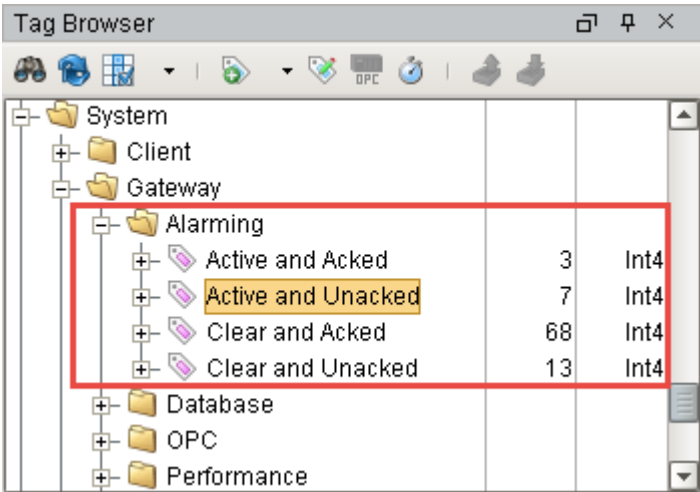
5. You'll find that the component will appear when an alarm is active, and become hidden once all alarms on the Tag are cleared.

Alarm Status Tags

Ignition provides a set of System Tags to view information about the Ignition server which includes four Tags that count the number of **alarms** in each state. A quick way to see if any alarms are currently active and get an alarm count is to add a label on the Navigation window. The four system alarm states are:

- Active and Unacknowledged
- Active and Acknowledged
- Clear and Acknowledged
- Clear and Unacknowledged

In the **Tag Browser** of the Designer (default position in middle-left side of the Designer), scroll down to the **System > Gateway > Alarming** folder. You can see all 4 of the system Tags that Ignition provides. You can also see how many alarms are currently **Active and Acked**, **Active and Unacked**, **Clear and Acked**, and **Clear and Unacked**.



These system Tags can easily be used to **visualize all alarms in the system**.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status Table Component



INDUCTIVE
UNIVERSITY

Alarm Status - Component

[Watch the video](#)

The **Alarm Status Table** is a built-in component in Ignition that displays the current status of all alarms in the system into one view. The Alarm Status Table is highly customizable and can be configured to show active, unacknowledged, cleared, and acknowledged alarms.

NOTE

Additional information on the Alarm Status Table component can be found on the [Alarm Status Table](#) page in the appendix.

Setting up the Alarm Status Table

In the **Component Palette** in the **Designer**, under Alarming, drag the **Alarm Status Table component** on to your window. You will notice that the table is already configured to show the current status of all the alarms in your system. By default, the Alarm Status Table shows all the "Active and Unacknowledged", "Active and Acknowledged", and "Cleared and Unacknowledged" alarms. It also displays the time the alarm went Active as well as the current State and Priority of the alarm.

Note: You must have alarms set up on your Tags for them to show up in the table.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Alarm Status
- Root Container
 - Alarm Status Table

Tag Browser

- Tags
 - Data Types
 - Alarms

Property Editor

Common

Name Alarm Status Table

Enabled ☒ true

Visible ☒ true

Border No Border

Behavior

Data

Appearance

Filters

Min Priority Diagnostic

Show Active and Unacked ☒ true

Show Active and Aacked ☒ true

Show Clear and Unacked ☒ true

Show Clear and Aacked ☐ false

Source Filter

Display Path Filter

Provider Filter

	Active Time	Display Path	Source Path	Current ...	Priority
☐	3/21/16 8:05 AM	Machine C	prov.default:tag:gensim/Writeable/WriteableFloat2/alm:Low ...	Active, U...	Critical
☐	3/21/16 8:05 AM	Writeable/Wri...	prov.default:tag:Writeable/WriteableInteger1/alm:Alarm	Active, U...	Critical
☐	3/21/16 8:12 AM	Machine B	prov.default:tag:3Writeable/WriteableFloat2/alm:High Alarm	Active, U...	Critical
☐	3/21/16 8:05 AM	Alarms.ProdS...	prov.default:tag:Alarms/ProdSupAttention/alm:Alarm	Active, U...	High
☐	3/21/16 8:05 AM	Machine B	prov.default:tag:Writeable/WriteableFloat1 /alm:pw Alarm	Active, U...	Medium
☐	3/21/16 8:05 AM	Machine B	prov.default:tag:gensim/Writeable/WriteableFloat2/alm:Low Alarm	Active, U...	Medium
☐	3/22/16 11:45 AM	Machine B	prov.default:tag:gensim/Writeable/WriteableFloat1 /alm:Low Ala...	Active, U...	Medium
☐	3/21/16 8:05 AM	Writeable/Writ...	prov.default:tag:Writeable/WriteableFloat1 /alm:Alarm	Active, U...	Low
☐	3/21/16 8:05 AM	Writeable/Writ...	prov.default:tag:Writeable/WriteableInteger2/alm:Alarm 1	Active, U...	Low
☐	3/21/16 8:05 AM	Writeable/Writ...	prov.default:tag:Writeable/WriteableInteger2/alm:Alarm	Active, U...	Low
☐	3/21/16 8:05 AM	Alarms.ProdS...	prov.default:tag:Alarms/ProdSupAttention/alm:Alarm 1	Active, U...	Low
☐	3/21/16 8:05 AM	Machine B	prov.default:tag:Writeable/WriteableFloat2/alm:High Alarm	Cleared, ...	Critical
☐	3/22/16 1:57 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp2/alm:Low Alarm 2	Cleared, ...	High
☐	3/22/16 1:57 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp2/alm:Low Alarm 2	Cleared, ...	High
☐	3/22/16 1:57 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp1 /alm:Low Alarm 1	Cleared, ...	High
☐	3/22/16 1:58 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp2/alm:Low Alarm 2	Cleared, ...	High
☐	3/22/16 1:58 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp1 /alm:Low Alarm 1	Cleared, ...	High
☐	3/22/16 1:58 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp1 /alm:Low Alarm 1	Cleared, ...	High
☐	3/22/16 1:58 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp2/alm:Low Alarm 2	Cleared, ...	High
☐	3/22/16 1:58 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp1 /alm:Low Alarm 1	Cleared, ...	High
☐	3/22/16 1:58 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp2/alm:Low Alarm 2	Cleared, ...	High
☐	3/22/16 1:58 PM	Ramp	prov.default:tag:gensim/Ramp/Ramp1 /alm:Low Alarm 1	Cleared, ...	High

Acknowledge Shelve

Component Palette

- Sparkline Chart
- Bar Chart
- Radar Chart
- Status Chart
- Pie Chart
- Box and Whisker Chart
- Equipment Schedule
- Gantt Chart
- Calendar
 - Calendar
 - Popup Calendar
 - Date Range
 - Day View
 - Week View
 - Month View
- Admin
 - User Management
 - Schedule Management
 - Roster Management
 - SFC Monitor
- Alarming
 - Alarm Status Table
 - Alarm Journal Table

In **Preview Mode**, right click on the header to show or hide information on the Alarm Status Table. If you want to show the name of the alarm, you can add 'Name' to the table by checking the box. You can also reorganize the columns to display how you want the columns to look by dragging the column header anywhere on the table header.

☐	Active Time	Display Path	Name ▲ 1	Source Path	Current St...	Priority
☐	3/21/16 8:05 AM	Machine B	:pw Alarm	prov.default:/tag:Writea	na...	Medium
☐	3/21/16 8:05 AM	Writeable/Writ...	Alarm	prov.default:/tag:Write	n...	Critical
☐	3/21/16 8:05 AM	Alarms/ProdS...	Alarm	prov.default:/tag:Alarm	n...	High
☐	3/21/16 8:05 AM	Writeable/Write...	Alarm	prov.default:/tag:Writea	na...	Low
☐	3/21/16 8:05 AM	Writeable/Write...	Alarm	prov.default:/tag:Writea	na...	Low
☐	3/21/16 8:05 AM	Writeable/Write...	Alarm 1	prov.default:/tag:Writea	na...	Low
☐	3/21/16 8:05 AM	Alarms/ProdSu...	Alarm 1	prov.default:/tag:Alarms	na...	Low
☐	3/21/16 8:12 AM	Machine B	High Alarm	prov.default:/tag:Write	n...	Critical
☐	3/21/16 8:05 AM	Machine B	High Alarm	prov.default:/tag:Writea	U...	Critical
☐	3/22/16 2:08 PM	Ramp	High Alarm 2	prov.default:/tag:gensi	U...	Medium
☐	3/22/16 2:08 PM	Ramp	High Alarm 2	prov.default:/tag:gensi	U...	Medium
☐	3/22/16 2:08 PM	Ramp	High Alarm 2	prov.default:/tag:gensi	U...	Medium
☐	3/22/16 2:08 PM	Ramp	High Alarm 2	prov.default:/tag:gensi	U...	Medium
☐	3/22/16 2:09 PM	Ramp	High Alarm 2	prov.default:/tag:gensi	U...	Medium
☐	3/21/16 8:05 AM	Machine C	Low Alarm	prov.default:/tag:gensi	n...	Critical
☐	3/21/16 8:05 AM	Machine B	Low Alarm	prov.default:/tag:Writea	na...	Medium
☐	3/22/16 11:45 AM	Machine B	Low Alarm	prov.default:/tag:gensi	na...	Medium
☐	3/21/16 8:05 AM	Machine B	Low Alarm	prov.default:/tag:gensi	U...	Medium
☐	3/21/16 8:12 AM	Machine B	Low Alarm	prov.default:/tag:gensi	U...	Medium
☐	3/22/16 2:08 PM	Ramp	Low Alarm 1	prov.default:/tag:gensi	U...	High
☐	3/22/16 2:08 PM	Ramp	Low Alarm 1	prov.default:/tag:gensi	U...	High
☐	3/22/16 2:08 PM	Ramp	Low Alarm 1	prov.default:/tag:gensi	U...	High

Auto Resize This Column
Auto Resize All Columns
Hide This Column "Source Path"
Show All Hidden Columns

☐ Ack Notes
☐ Ack Time
☐ Ack'ed By
☐ Active Pipeline
☐ Clear Pipeline
☐ Ack Pipeline
☒ Active Time
☐ Clear Time
☐ Event Value
☐ Deadband
☒ Display Path
☐ Acked?
☐ Active?
☐ Clear?
☒ Name
☐ Notes
☒ Priority
☒ Source Path
☒ Current State
☐ eventId
☐ Unacknowledged Duration
☐ Active Duration

Reset to Default
More...

Acknowledging and Shelving Alarms

You can click on a individual alarm and use the buttons at the bottom of the table to Acknowledge or Shelve the selected alarm. You can also use the icons to view the alarm Details, Notes, and History.

☐	Active Time	Display Path	Name	Source Path	Current State	Priority
☐	3/21/16 8:12 AM	Machine B	High Alarm	prov.default/tag:Writeable/Write...	Active, Unacknowledged	Critical
☐	3/21/16 8:05 AM	Machine B	High Alarm	prov.default/tag:Writeable/Writeab...	Cleared, Unacknowledged	Critical
☒	3/22/16 2:22 PM	Ramp	High Alarm 2	prov.default/tag:gensim/Ramp/Ra...	Active, Unacknowledged	Medium
☐	3/22/16 2:21 PM	Ramp	High Alarm 2	prov.default/tag:gensim/Ramp/Ra...	Cleared, Unacknowledged	Medium
☐	3/22/16 2:21 PM	Ramp	High Alarm 2	prov.default/tag:gensim/Ramp/Ra...	Cleared, Unacknowledged	Medium
☐	3/22/16 2:22 PM	Ramp	High Alarm 2	prov.default/tag:gensim/Ramp/Ra...	Cleared, Unacknowledged	Medium
☐	3/22/16 2:22 PM	Ramp	High Alarm 2	prov.default/tag:gensim/Ramp/Ra...	Cleared, Unacknowledged	Medium
☐	3/21/16 8:05 AM	Machine C	Low Alarm	prov.default/tag:gensim/Writeabl...	Active, Unacknowledged	Critical
☐	3/21/16 8:05 AM	Machine B	Low Alarm	prov.default/tag:Writeable/Writeab...	Active, Unacknowledged	Medium
☐	3/22/16 11:45 AM	Machine B	Low Alarm	prov.default/tag:gensim/Writeable...	Active, Unacknowledged	Medium
☐	3/21/16 8:05 AM	Machine B	Low Alarm	prov.default/tag:gensim/Writeable...	Cleared, Unacknowledged	Medium
☐		Machine B	Low Alarm	prov.default/tag:gensim/Writeable...	Cleared, Unacknowledged	Medium

☐ Details
☐ Notes

Config Properties

Source Path

prov.default/tag:gensim/Ramp/Ramp2/alm:High Alarm 2

Display Path

Ramp

Name

High Alarm 2

Mode

Above Setpoint

Priority

Medium

Current State

Cleared, Unacknowledged

Ack'd By

Active Pipeline

Clear Pipeline

Deadband

0

Active delay (seconds)

0

Clear delay (seconds)

0

Timestamp Source

System

Acknowledge

Shelve

Filter Properties

In the [Property Editor](#), there are a set of filter properties. Using the **Source Filter** and the **Display Path Filter** properties, you can filter out cleared alarms, filter for alarms in a particular Tag folder, and filter for alarms matching a certain criteria. This example filters for all alarms with 'Machine' in the Display Path.

Filtering on alarms is discussed in detail in sections, [Alarm Status General Filtering](#) and [Alarm Status Filter on Associated Data](#).

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Alarm Status
 - Root Container
 - Alarm Status Table

Tag Browser

Tag	Value	Data Type
Tags		
Data Types		
Alarms		

Property Editor

Common

Behavior

Data

Appearance

Filters

Min Priority	D diagnostic
Show Active and Unacked	☒ true
Show Active and Aacked	☒ true
Show Clear and Unacked	☒ true
Show Clear and Aacked	☐ false
Source Filter	
Display Path Filter	Machine*
Provider Filter	

Active Time

Display Path

Name

Source Path

Current State

Priority

3/21/16 8:05 AM	Machine B	Low Alarm	prov.default/tag:Writeable/Writeabl...	Active, Unacknowledged	Medium
3/21/16 8:12 AM	Machine B	High Alarm	prov.default/tag:Writeable/Writea...	Active, Unacknowledged	Critical
3/21/16 8:05 AM	Machine B	High Alarm	prov.default/tag:Writeable/Writeabl...	Cleared, Unacknowledged	Critical
3/21/16 8:05 AM	Machine C	Low Alarm	prov.default/tag:gensim/Writeabl...	Active, Unacknowledged	Critical
3/21/16 8:05 AM	Machine B	Low Alarm	prov.default/tag:Writeable/Writeabl...	Active, Unacknowledged	Medium
3/22/16 11:45 AM	Machine B	Low Alarm	prov.default/tag:gensim/Writeable/...	Active, Unacknowledged	Medium
3/21/16 8:05 AM	Machine B	Low Alarm	prov.default/tag:gensim/Writeable/...	Cleared, Unacknowledged	Medium
3/21/16 8:12 AM	Machine B	Low Alarm	prov.default/tag:gensim/Writeable/...	Cleared, Unacknowledged	Medium

Acknowledge

Shelve

The Alarm Status Table contains a host of features already built into the component which you can take advantage of to give you the current status of alarms in your system.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status - General Filtering



INDUCTIVE
UNIVERSITY

Alarm Status - General Filtering

[Watch the video](#)

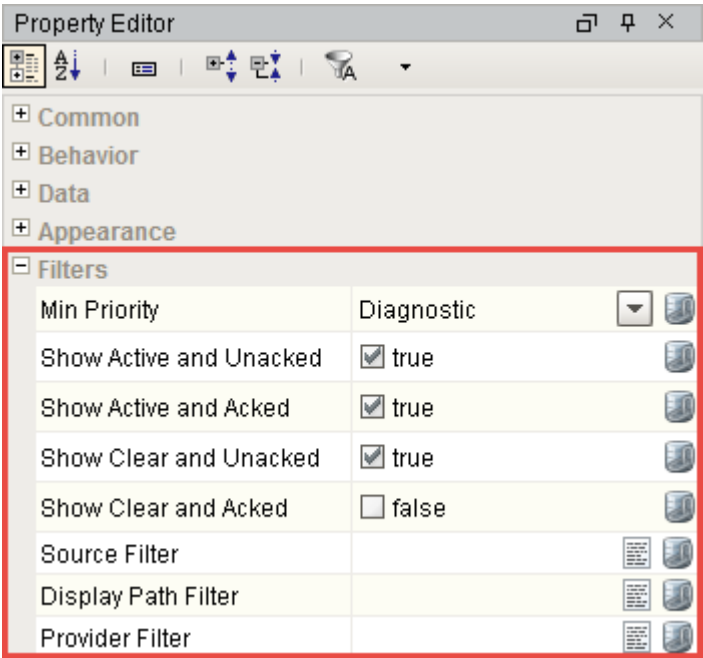
Filtering Properties

There are several different filtering properties on the Alarm Status Table component. Once you drag the Alarm Status Table component on to a window, by default it shows you all the alarms that are currently 'Active and Unacknowledged,' 'Active and Acknowledged,' and 'Cleared and Unacknowledged' with any priority.

The screenshot shows the Ignition Alarm Status Table component in a project window. The table displays a list of alarms with columns for Active Time, Display Path, Current State, and Priority. The table is filtered to show only 'Active, Unacknowledged' alarms. The 'Filters' panel on the left shows the 'Min Priority' set to 'Diagnostic' and 'Show Active and Unacked', 'Show Active and Acked', and 'Show Clear and Unacked' all checked.

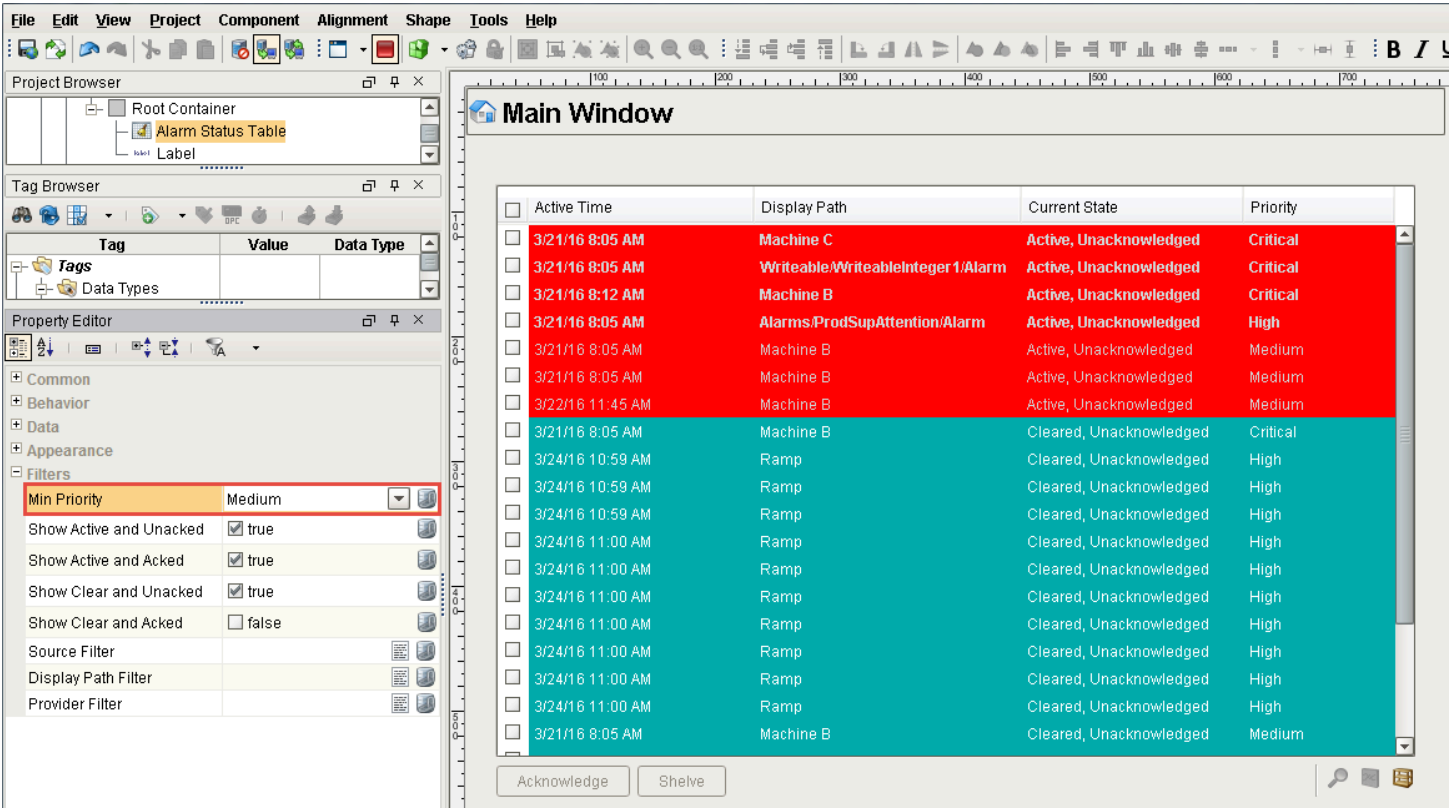
Active Time	Display Path	Current State	Priority
3/21/16 8:05 AM	Machine C	Active, Unacknowledged	Critical
3/21/16 8:05 AM	Writeable/WriteableInteger1/Alarm	Active, Unacknowledged	Critical
3/21/16 8:12 AM	Machine B	Active, Unacknowledged	Critical
3/21/16 8:05 AM	Alarms/ProdSupAttention/Alarm	Active, Unacknowledged	High
3/24/16 10:37 AM	Ramp	Active, Unacknowledged	High
3/21/16 8:05 AM	Machine B	Active, Unacknowledged	Medium
3/21/16 8:05 AM	Machine B	Active, Unacknowledged	Medium
3/22/16 11:45 AM	Machine B	Active, Unacknowledged	Medium
3/24/16 10:37 AM	Ramp	Active, Unacknowledged	Medium
3/21/16 8:05 AM	Writeable/WriteableFloat1/Alarm	Active, Unacknowledged	Low
3/21/16 8:05 AM	Writeable/WriteableInteger2/Alarm 1	Active, Unacknowledged	Low
3/21/16 8:05 AM	Writeable/WriteableInteger2/Alarm	Active, Unacknowledged	Low
3/21/16 8:05 AM	Alarms/ProdSupAttention/Alarm 1	Active, Unacknowledged	Low
3/21/16 8:05 AM	Machine B	Cleared, Unacknowledged	Critical
3/24/16 10:35 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 10:36 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 10:36 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 10:36 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 10:36 AM	Ramp	Cleared, Unacknowledged	High

You can filter the alarm list in the table to be a shorter list using the filter properties in the Property Editor instead of scrolling through every single alarm in your system. In the **Property Editor**, there is a group called **Filters** that you can configure to focus on only those alarms you want to see. You can filter on **Priority**, **State**, **Source Path**, **Display Path**, and **Tag Provider**.



Filtering on Priority

This example shows how you can filter alarms based on **Priority**. From the **Min Priority** field, select **'Medium'** from the dropdown list. You will then see only the Medium priority and higher alarms displayed which creates a smaller list of alarms for an operator to view.



In this next example, select ‘High’ from the dropdown list in the **Min Priority** field. Now, you will see all High and Critical priority alarms in the system which gives the operator an even smaller list of alarms to view.

FileEditViewProjectComponentAlignmentShapeToolsHelp

Project Browser

Root Container

Alarm Status Table

Label

Tag Browser

Tags

Data Types

Property Editor

Common

Behavior

Data

Appearance

Filters

Min Priority

High

Show Active and Unacked

true

Show Active and Aacked

true

Show Clear and Unacked

true

Show Clear and Aacked

false

Source Filter

Display Path Filter

Provider Filter

Main Window

Active Time	Display Path	Current State	Priority
3/21/16 8:05 AM	Machine C	Active, Unacknowledged	Critical
3/21/16 8:05 AM	Writeable/WriteableInteger 1/Alarm	Active, Unacknowledged	Critical
3/21/16 8:12 AM	Machine B	Active, Unacknowledged	Critical
3/21/16 8:05 AM	Alarms.ProdSupAttention/Alarm	Active, Unacknowledged	High
3/24/16 11:04 AM	Ramp	Active, Unacknowledged	High
3/21/16 8:05 AM	Machine B	Cleared, Unacknowledged	Critical
3/24/16 11:02 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 11:03 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 11:03 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 11:03 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 11:03 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 11:03 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 11:03 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 11:03 AM	Ramp	Cleared, Unacknowledged	High
3/24/16 11:03 AM	Ramp	Cleared, Unacknowledged	High

AcknowledgeShelve

Lastly, select ‘Critical’ from the dropdown list. Notice how the alarm list keeps getting shorter by using the Priority filter.

FileEditViewProjectComponentAlignmentShapeToolsHelp

Project Browser

Root Container

Alarm Status Table

Label

Tag Browser

Tags

Data Types

Property Editor

Common

Behavior

Data

Appearance

Filters

Min Priority

Critical

Show Active and Unacked

true

Show Active and Aacked

true

Show Clear and Unacked

true

Show Clear and Aacked

false

Source Filter

Display Path Filter

Provider Filter

Main Window

Active Time	Display Path	Current State	Priority
3/21/16 8:05 AM	Machine C	Active, Unacknowledged	Critical
3/21/16 8:05 AM	Writeable/WriteableInteger 1/Alarm	Active, Unacknowledged	Critical
3/21/16 8:12 AM	Machine B	Active, Unacknowledged	Critical
3/21/16 8:05 AM	Machine B	Cleared, Unacknowledged	Critical

AcknowledgeShelve

Filtering on Alarm State

The second set of filters is based on **State**. There are 4 states an alarm can be in; Active and Unacknowledged, Active and Acknowledged, Clear and Unacknowledged, and Clear and Acknowledged. Each alarm has a different color representing one of the four States.

Project Browser

Root Container

Alarm Status Table

Label

Tag Browser

Tags

Data Types

Property Editor

Common

Behavior

Data

Appearance

Filters

Min Priority

High

Show Active and Unacked

☒

true

Show Active and Acked

☒

true

Show Clear and Unacked

☒

true

Show Clear and Acked

☒

true

Source Filter

Display Path Filter

Provider Filter

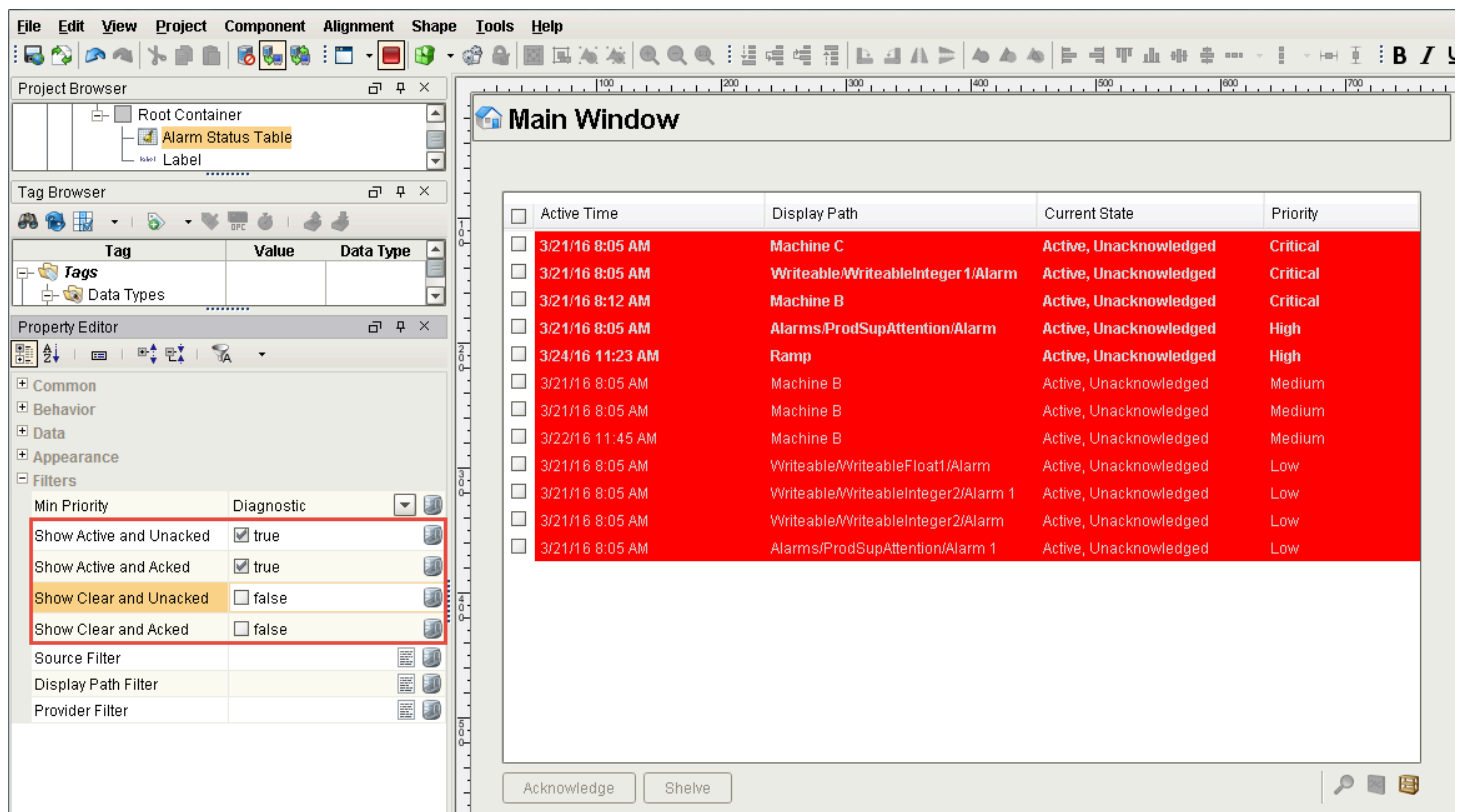
Main Window

Active Time	Display Path	Current State	Priority
☐ 3/21/16 8:05 AM	Machine C	Active, Unacknowledged	Critical
☐ 3/21/16 8:05 AM	Writeable/WriteableInteger 1/Alarm	Active, Unacknowledged	Critical
☐ 3/21/16 8:12 AM	Machine B	Active, Unacknowledged	Critical
☐ 3/21/16 8:05 AM	Alarms/ProdSupAttention/Alarm	Active, Unacknowledged	High
☐ 3/21/16 8:05 AM	Machine B	Cleared, Unacknowledged	Critical
☐ 3/24/16 11:11 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:11 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:11 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:11 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:11 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:12 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:12 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:12 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:12 AM	Ramp	Cleared, Unacknowledged	High
☐ 3/24/16 11:12 AM	Ramp	Cleared, Unacknowledged	High
☐	Writeable/WriteableBoolean1/Estop ...	Cleared, Acknowledged	Critical
☐	Machine A	Cleared, Acknowledged	Critical
☐	gensim/Writeable/WriteableBoolean...	Cleared, Acknowledged	Critical
☐	Machine B	Cleared, Acknowledged	Critical

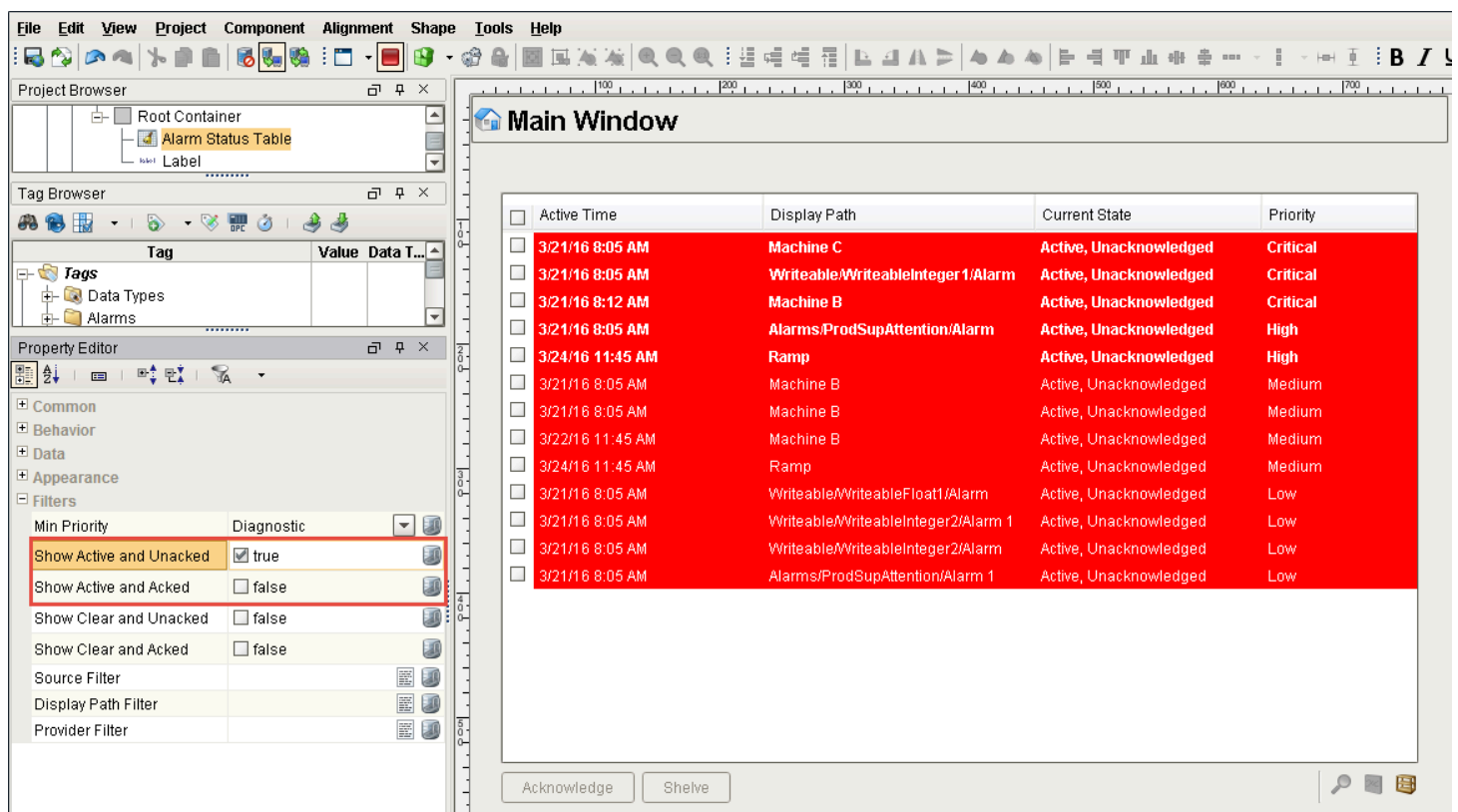
Acknowledge

Shelve

Let's make our alarm list smaller by **unchecking** both the **Clear** filters. This will eliminate all the Clear alarms from the Alarm Status Table and only show the Active alarms whether they are Acknowledged or Unacknowledged.



If you uncheck **Active and Acknowledged**, this will only show the alarms in the Alarm Status Table that are currently Active and are not Acknowledged. A good business practice is to show only the Active alarms that are either Unacknowledged or Acknowledged. It also reduces the number of alarms an operator has to scroll through.



Filtering on source Path and Display Path

Ignition also has the capability to filter alarms based on the Source or Tag Path and the Display Path.

Source Path

The **Source** is the actual **Tag path** which means you can also use Tag folders in Ignition to filter for specific alarms. For example, you may want to filter for all Ramp alarms in the 'Ramp' folder. Enter ***Ramp*** (using * as a wildcard) to look for all alarms with the Ramp Tag Path. Each of these properties allow you to restrict the results of the query to one or more paths. Multiple paths may be specified with a comma. Additionally, these properties all use the asterisk (*) as wildcard character to denote any number of leading or trailing characters, depending on placement. Some examples using Source Paths are listed below.

Example Filter	Result
prov:tagProvider:/tag:Inputs/PS_1:/alm:MyAlarm	Retrieve alarm information from the alarm at provided path: prov:tagProvider:/tag:Inputs/PS_1:/alm:MyAlarm
*PS_1:/alm:MyAlarm	Retrieves alarm information from any path that ends with PS_1:/alm:MyAlarm. Thus the following paths would be returned: prov:tagProvider:/tag:Inputs/PS_1:/alm:MyAlarm prov:tagProvider:/tag:anotherFolder/different_Path/PS_1:/alm:MyAlarm
prov:tagProvider:/tag:PS_*	Retrieves alarm information from any source path that starts with PS_ prov:tagProvider:/tag:PS_1/MyAlarm:/alm:MyAlarm prov:tagProvider:/tag:PS_2/MyAlarm:/alm:MyAlarm
MyAlarm	Retrieves any alarm information that has MyAlarm in the path.

You can see in this example, all the alarms have 'Ramp' in the Source Path.

Project Browser

- Root Container
 - Alarm Status Table

Tag Browser

- Alarms
 - Device Status
 - gensim
 - _Diagnostics_
 - Ramp

Property Editor

- Common
- Behavior
- Data
- Appearance
- Filters
 - Min Priority: Diagnostic
 - Show Active and Unacked: ☒ true
 - Show Active and Aced: ☒ true
 - Show Clear and Unacked: ☒ true
 - Show Clear and Aced: ☐ false
 - Source Filter: *Ramp*
 - Display Path Filter
 - Provider Filter

Main Window

Active Time	Source Path	Display Path	Current State	Priority	
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:High Alarm 2	Ramp	Cleared, Unack...	Medium
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:Low Alarm 2	Ramp	Cleared, Unack...	High
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:High Alarm 2	Ramp	Cleared, Unack...	Medium
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:Low Alarm 2	Ramp	Cleared, Unack...	High
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp1/alm:Low Alarm 1	Ramp	Cleared, Unack...	High
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:High Alarm 2	Ramp	Cleared, Unack...	Medium
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:Low Alarm 2	Ramp	Cleared, Unack...	High
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp1/alm:Low Alarm 1	Ramp	Cleared, Unack...	High
☐	3/24/16 1:59 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:Low Alarm 2	Ramp	Cleared, Unack...	High
☐	3/24/16 2:00 PM	prov.default/tag.gensim/Ramp/Ramp1/alm:Low Alarm 1	Ramp	Cleared, Unack...	High
☐	3/24/16 2:00 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:High Alarm 2	Ramp	Cleared, Unack...	Medium
☐	3/24/16 2:00 PM	prov.default/tag.gensim/Ramp/Ramp2/alm:Low Alarm 2	Ramp	Active, Unackn...	High
☐	3/24/16 2:00 PM	prov.default/tag.gensim/Ramp/Ramp1/alm:Low Alarm 1	Ramp	Active, Unackn...	High

Acknowledge Shelve

Display Path

Essentially, you can do the same thing using the **Display Path**. In this example, 'High Alarm' is the name that was setup in the [Alarm Configuration](#), and the Display Path is set to the name you want the operator to see. So for example, the Alarm Status Table shows all the High Alarms.

Project Browser

- Root Container
 - Alarm Status Table

Tag Browser

- Tags
 - Data Types
 - Alarms

Property Editor

- Common
- Behavior
- Data
- Appearance
- Filters
 - Min Priority: Diagnostic
 - Show Active and Unacked: ☒ true
 - Show Active and Aced: ☒ true
 - Show Clear and Unacked: ☒ true
 - Show Clear and Aced: ☐ false
 - Source Filter
 - Display Path Filter: *High Alarm*
 - Provider Filter

Main Window

Active Time	Display Path	Name	Current State	Priority	
☐	3/21/16 8:12 AM	Machine B	High Alarm	Active, Unacknowledged	Critical
☐	3/21/16 8:05 AM	Machine B	High Alarm	Cleared, Unacknowledged	Critical
☐	3/24/16 2:41 PM	Ramp	High Alarm 2	Active, Unacknowledged	Medium
☐	3/24/16 2:40 PM	Ramp	High Alarm 2	Cleared, Unacknowledged	Medium
☐	3/24/16 2:40 PM	Ramp	High Alarm 2	Cleared, Unacknowledged	Medium
☐	3/24/16 2:40 PM	Ramp	High Alarm 2	Cleared, Unacknowledged	Medium
☐	3/24/16 2:40 PM	Ramp	High Alarm 2	Cleared, Unacknowledged	Medium

Acknowledge Shelve

So, it's really easy to do some basic filtering using the alarm filter properties on the Alarm Status Table component.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status - Filter on Associated Data



INDUCTIVE
UNIVERSITY

**Alarm Status - Filter on
Associated Data**

[Watch the video](#)

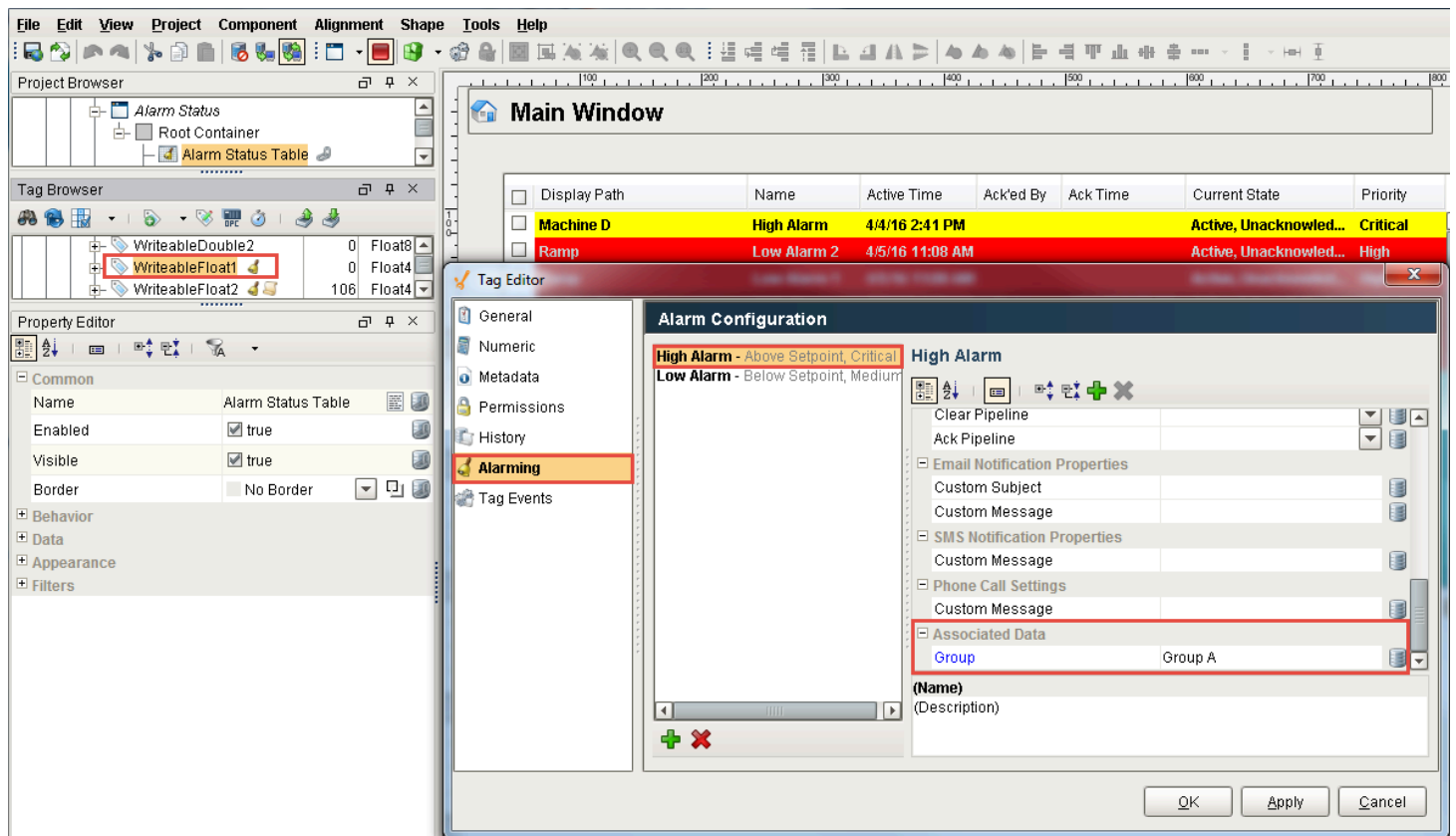
Filtering on Associated Data

Another way to filter alarms in the **Alarm Status Table** is using associated data that was added to an alarm. You can easily and quickly search, filter, and display on specific alarms based on associated data configured in an alarm.

Main Window

☐	Display Path	Name	Active Time	Ack'd By	Ack Time	Current State	Priority
☐	Machine D	High Alarm	4/4/16 2:41 PM			Active, Unacknowledged...	Critical
☐	Machine B	High Alarm	3/31/16 9:28 AM	admin	3/31/16 9:35 AM	Active, Acknowledged	Critical
☐	Writeable/WriteableInteger1/...	Alarm	3/31/16 9:28 AM	admin	3/31/16 9:36 AM	Active, Acknowledged	Critical
☐	Alarms/ProdSupAttention/Alar...	Alarm	3/31/16 9:28 AM	admin	3/31/16 9:36 AM	Active, Acknowledged	High
☐	Machine B	High Alarm	3/31/16 9:36 AM			Cleared, Unacknowledge...	Critical
☐	Machine D	High Alarm	4/4/16 2:33 PM			Cleared, Unacknowledge...	Critical
☐	Machine D	High Alarm	4/4/16 2:37 PM			Cleared, Unacknowledge...	Critical
☐	Machine D	High Alarm	4/4/16 2:39 PM			Cleared, Unacknowledge...	Critical
☐	Machine D	High Alarm	4/4/16 2:40 PM			Cleared, Unacknowledge...	Critical
☐	Ramp	Low Alarm 2	4/5/16 11:00 AM			Cleared, Unacknowledge...	High
☐	Ramp	Low Alarm 2	4/5/16 11:00 AM			Cleared, Unacknowledge...	High
☐	Ramp	Low Alarm 1	4/5/16 11:01 AM			Cleared, Unacknowledge...	High
☐	Ramp	Low Alarm 2	4/5/16 11:01 AM			Cleared, Unacknowledge...	High
☐	Ramp	Low Alarm 1	4/5/16 11:01 AM			Cleared, Unacknowledge...	High

It's a common design practice to associate alarm groupings on associated data of an alarm. In this example, a **'WriteableFloat1'** tag was used to configure a **'High Alarm'**, and associated data was added called **'Group'** with a static value called **'Group A'**.



Using a Script to Filter on Associated Data

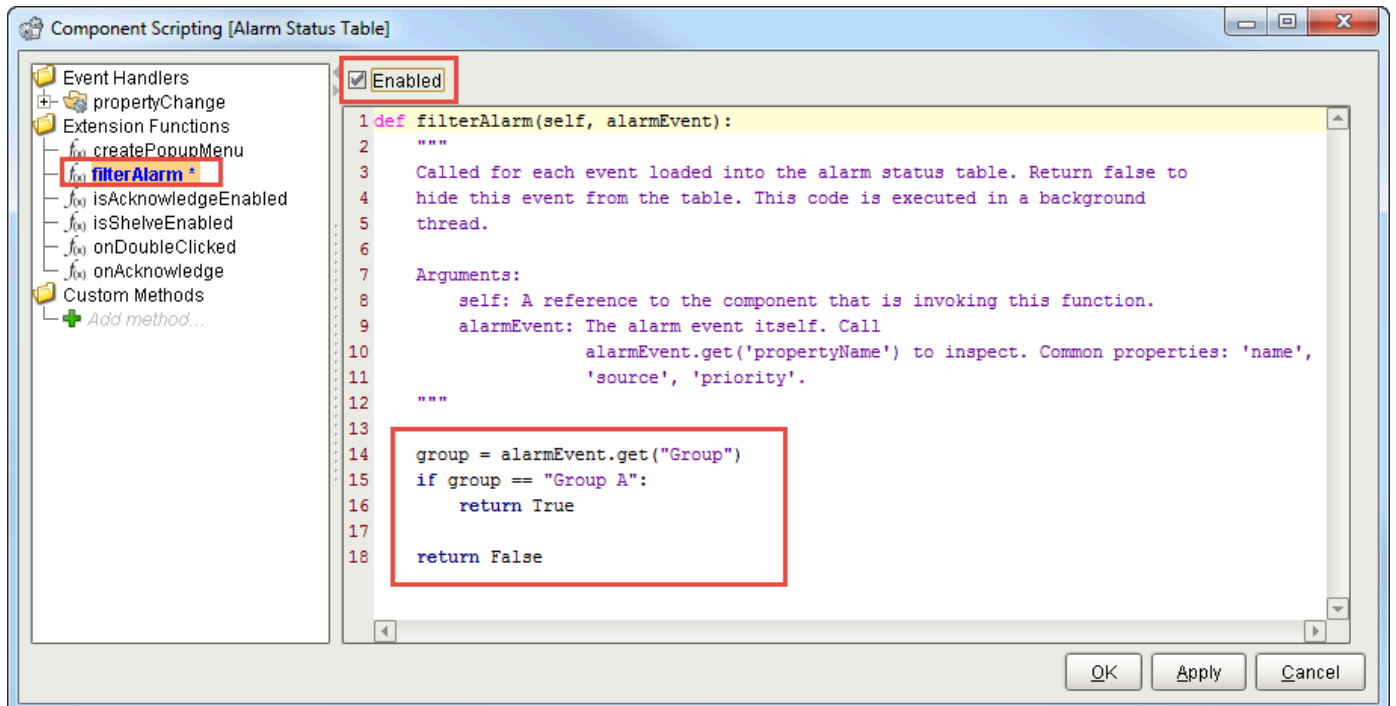
Let's create a script to filter for all the alarms in the Alarm Status Table that have the associated data called **'Group A'**. Scripting allows you to use the **'filterAlarm'** Extension Function specifically for filtering on associated data.

1. Right click on the Alarm Status Table component, and scroll down and select **Scripting**.
2. In the Component Scripting window, under **Extension Functions**, select **'filterAlarm'**.
3. Click the **Enabled** checkbox.
4. Enter the code below into the **'filterAlarm'** script. You can filter on anything you want here, but in this example, we are going to filter on **'Group A'**.

Extension Function - filterAlarm for 'Group A'

```
group = alarmEvent.get("Group")
if group == "Group A":
    return True
return False
```

Once your code is entered, it will look like this...



Click **OK**.

5. This script only displays alarms matching the associated data for '**Group A**'. For every alarm matching '**Group A**', it will return '**True**' and show the alarm in the Alarm Status Table. If the associated data does not match '**Group A**', it will return '**False**', and the alarm will not show up in the Alarm Status Table. The example below shows the alarm including all the details about the alarm.

Note: if you are seeing alarms that do not match your associated data, check your filter settings in the Property Editor. You may need to uncheck the "Show Clear and Unacked" and "Show Clear and Acked" settings depending on what you want your operators to see.

Main Window

☒	Display Path	Name	Active Time	Source Path	Current State	Priority	Event Value
☒	Machine A	High Alarm	9/13/16 4:01 PM	prov:default:tag:gensim/Writeable/Wri...	Active, Unack...	Critical	103

Details

Notes

On Active

mode	Above Setpoint
setpointA	100
Event Value	103
activePipeline	Basic Email
name	High Alarm
Event Time	9/13/16 4:01 PM
displayPath	Machine A
priority	Critical
ackNotesReqd	True
Group	Group A

Acknowledge

Shelve

NOTE

Important: If you have an error in this filtering script, it will return true for every alarm instance (and show all alarms) instead of displaying many errors to your users. You can find more information about the error in the Client Console under the Tools Menu.

Scripting gives you the ability to filter on associated data, but also gives you the freedom to filter on anything that you want.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status - Row Styles



INDUCTIVE
UNIVERSITY

Alarm Status - Row Styles

[Watch the video](#)

The Alarm Status Table comes with a particular set of colors associated with each of the alarm states. You can change these colors for each of the states by going to the Alarms Row Styles Customizer that the Alarm Status Table component provides.

By default, the Active and Unacknowledged alarms are at the very top of the table and are red and blinking, and Cleared and Unacknowledged alarms are teal and at the bottom of the table, as shown in the example below.

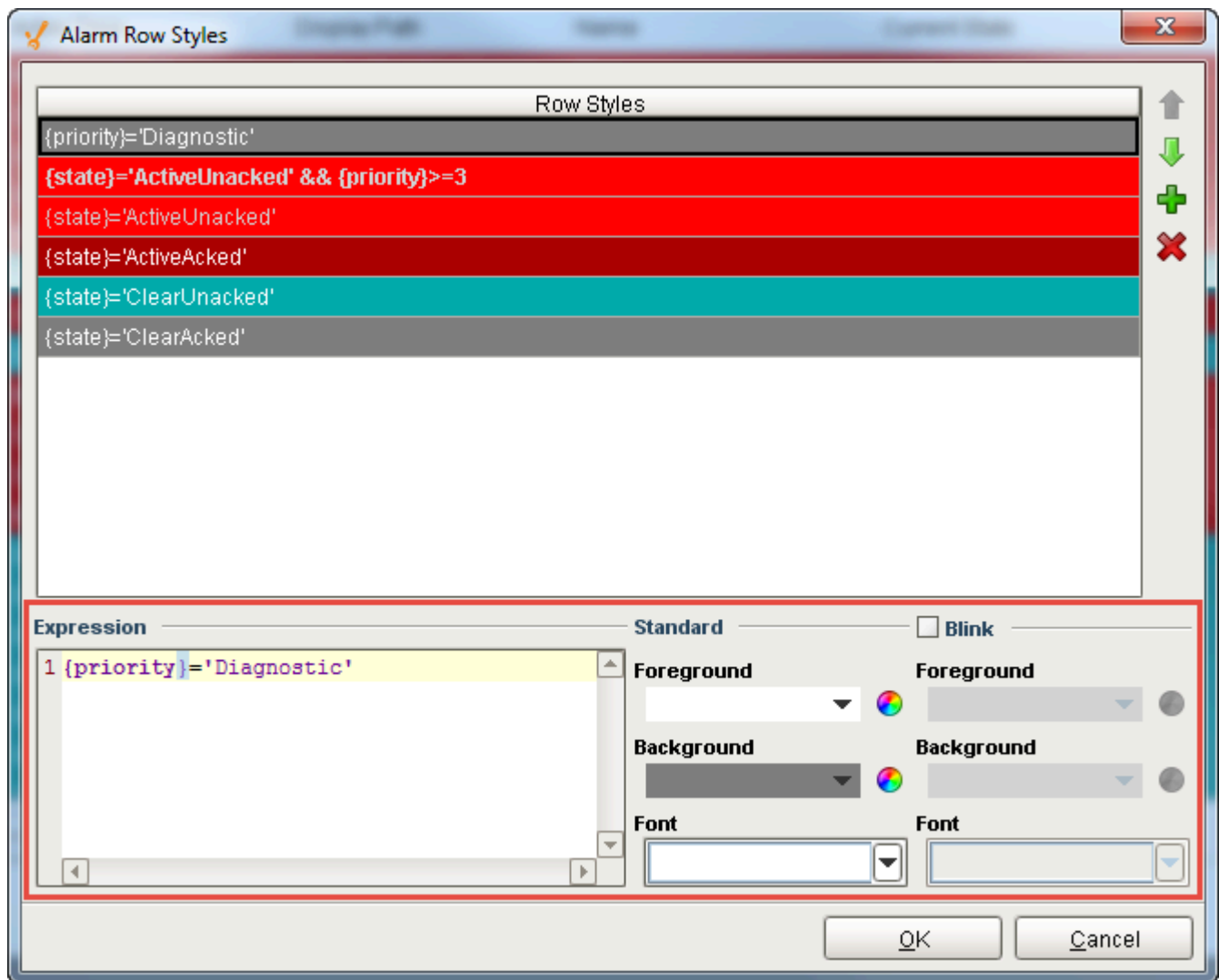
☐	Active Time	Display Path	Name	Current State	Priority
☐	9/12/16 8:17 AM	Machine B	High Alarm	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	gensim/Writeable/Write...	Fault	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Writeable/WriteableInte...	Alarm	Active, Unacknowledged	Critical
☐	9/13/16 4:01 PM	Machine D	High Alarm	Active, Unacknowledged	Critical
☐	9/13/16 4:01 PM	Machine A	High Alarm	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Alarms/ProdSupAttention...	Alarm	Active, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	Low Alarm 1	Active, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	High Alarm 2	Active, Unacknowledged	Medium
☐	9/12/16 8:17 AM	Alarms/ProdSupAttention/...	Alarm 1	Active, Unacknowledged	Low
☐	9/12/16 8:17 AM	Writeable/WriteableInteg...	Alarm 1	Active, Unacknowledged	Low
☐	9/12/16 8:17 AM	Writeable/WriteableInteg...	Alarm	Active, Unacknowledged	Low
☐	9/13/16 12:08 PM	gensim/Writeable/Writea...	High Alarm	Cleared, Unacknowledged	Critical
☐	9/13/16 12:12 PM	Machine A	High Alarm	Cleared, Unacknowledged	Critical
☐	9/13/16 2:54 PM	Machine A	High Alarm	Cleared, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Machine C	Low Alarm	Cleared, Unacknowledged	Critical
☐	9/14/16 9:47 AM	Ramp	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	Low Alarm 1	Cleared, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	Low Alarm 1	Cleared, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	Low Alarm 1	Cleared, Unacknowledged	High
☐	9/14/16 9:48 AM	Ramp	Low Alarm 1	Cleared, Unacknowledged	High

Customizing Alarm Row Styles

In the Alarm Row Styles Customizer is where you can modify an existing style, add more styles, or delete a style. Each style has an expression, a color and if you want it to blink. The expression allows you to do any evaluation you want using any parts of the alarm: Priority, State, Display Path, Active Time, Clear Time, and many more.

1. In the Designer, right click on the **Alarm Status Table** component, go to **Customizers > Alarm Row Styles**. The Alarm Row Styles window will appear displaying the default row styles.

Alarm Row Styles is an ordered list and each style has an expression. How it works is, the first style that returns **'True'** for a given alarm is the one that is going to be used, so you want to make sure the order is the correct order that you want. Click on each of the row styles to view the expression and row style.



2. Let's create another state alarm called '**ActiveUnacked**' and make it '**Critical**'. Set this new expression to return '**True**' or '**False**'. You can copy and paste the expression from the '**ActiveUnacked**' state with a **priority =3** (i.e., second style row) to create your new expression.
 - i. Click on the green plus icon .
 - ii. Enter or paste your new expression. Change the priority from '**3**' to '**4**'.

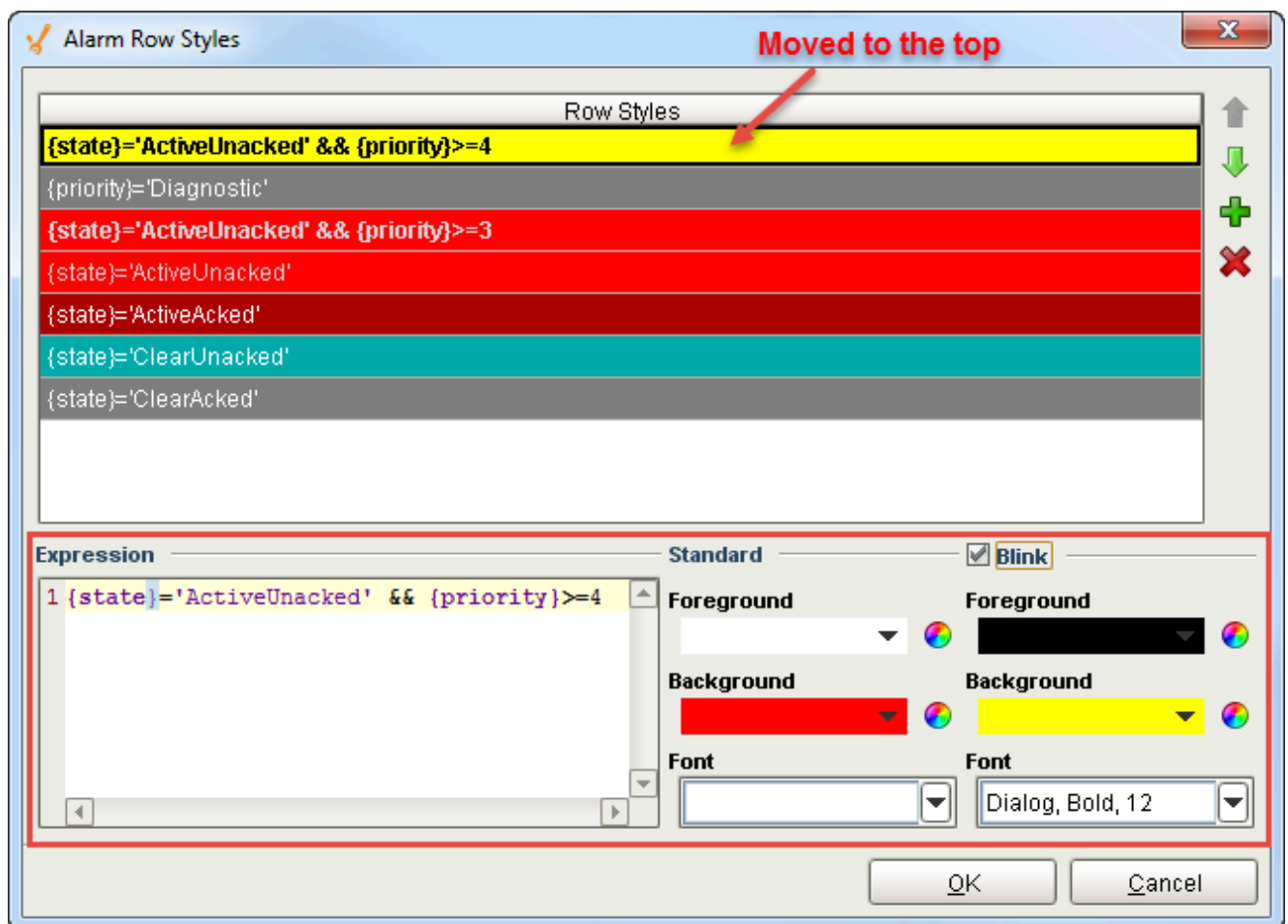
Active and Unacknowledged Critical alarm expression

```
{state}='ActiveUnacked' && {priority}>=4
```

TIP

Priorities are numbered behind the scenes, but named when you are creating them. 0 = Diagnostic, 1 = Low, 2 = Medium, 3 = High, and 4 = Critical. This allows you to use operations like '>=' (greater than or equal) when filtering on priority.

- iii. Because this is a critical alarm, you want it to catch the operator's attention, so it's a good idea to make the style standout. Make the foreground color '**white**' and the background color '**red**'. Check the **blink box** and make the foreground color '**black**' and background color '**yellow**'.
- iv. Style order is important, so if you keep this style at the bottom of the list, the alarm will never be visible to the operator because the first state to return '**True**' is **ActiveUnacked** with a **Priority** ≥ 3 . Move **ActiveUnacked** with a **Priority** ≥ 4 to the top of the row styles list, so it is evaluated before the other styles.
- v. Press **OK**.



3. The critical alarms will now blink red and yellow and will definitely catch the operator's attention.

☐	Active Time	Display Path	Name	Current State	Priority
☐	9/12/16 8:17 AM	Machine B	High Alarm	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	gensim/Writeable/Write...	Fault	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Writeable/WriteableInte...	Alarm	Active, Unacknowledged	Critical
☐	9/13/16 4:01 PM	Machine D	High Alarm	Active, Unacknowledged	Critical
☐	9/13/16 4:01 PM	Machine A	High Alarm	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Alarms/ProdSupAttentio...	Alarm	Active, Unacknowledged	High
☐	9/14/16 3:01 PM	Ramp	Low Alarm 1	Active, Unacknowledged	High
☐	9/14/16 3:01 PM	Ramp	High Alarm 2	Active, Unacknowledged	Medium
☐	9/12/16 8:17 AM	Alarms/ProdSupAttention/...	Alarm 1	Active, Unacknowledged	Low
☐	9/12/16 8:17 AM	Writeable/WriteableInteg...	Alarm 1	Active, Unacknowledged	Low
☐	9/12/16 8:17 AM	Writeable/WriteableInteg...	Alarm	Active, Unacknowledged	Low
☐	9/13/16 12:08 PM	gensim/Writeable/Writea...	High Alarm	Cleared, Unacknowledged	Critical
☐	9/13/16 12:12 PM	Machine A	High Alarm	Cleared, Unacknowledged	Critical
☐	9/13/16 2:54 PM	Machine A	High Alarm	Cleared, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Machine C	Low Alarm	Cleared, Unacknowledged	Critical
☐	9/14/16 3:00 PM	Ramp	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/14/16 3:00 PM	Ramp	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/14/16 3:01 PM	Ramp	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/14/16 3:01 PM	Ramp	Low Alarm 1	Cleared, Unacknowledged	High
☐	9/14/16 3:01 PM	Ramp	Low Alarm 1	Cleared, Unacknowledged	High
☐	9/14/16 3:01 PM	Ramp	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/14/16 3:01 PM	Ramp	Low Alarm 1	Cleared, Unacknowledged	High

Acknowledge

Shelve



To learn more about states and priorities, including all the alarm configuration properties, refer to the sections on [Alarm Event Properties Reference](#).

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status - Marquee Mode



INDUCTIVE
UNIVERSITY

Alarm Status - Marquee Mode

[Watch the video](#)

The Alarm Status Table can take up a lot of space on your Ignition client window as shown below. If you want to minimize the size of your Alarm Status Table, and not miss any alarms, you can turn on the **Marquee Mode** option.

The screenshot shows the Ignition client window with the Alarm Status Table in Marquee Mode. The table is a vertical list of alarms, with the top section (Active, Unacknowledged) highlighted in yellow and red. The bottom section (Cleared, Unacknowledged) is highlighted in teal. The table has columns for Active Time, Name, Ack Time, Ack'd By, Current State, and Priority. A red banner at the bottom left indicates '7 Active Alarms'. The interface includes a menu bar (Command, Windows, Help), a sidebar with navigation options (Overview, User Management, Schedule Management, Roster Management, Tank Details, Alarm Status, Plant A), and a top right corner with user information (Logged In: admin) and buttons (Lock Screen, Switch User, Logout).

Active Time	Name	Ack Time	Ack'd By	Current State	Priority
9/12/16 8:17 AM	Fault			Active, Unacknowledged	Critical
9/15/16 12:56 PM	High Alarm			Active, Unacknowledged	Critical
9/15/16 2:34 PM	Low Alarm 1			Active, Unacknowledged	High
9/15/16 2:34 PM	High Alarm 2			Active, Unacknowledged	Medium
9/12/16 8:17 AM	High Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	Critical
9/15/16 11:03 AM	High Alarm	9/15/16 11:57 AM	admin	Active, Acknowledged	Critical
9/12/16 8:17 AM	Alarm	9/15/16 12:55 PM	admin	Active, Acknowledged	Critical
9/12/16 8:17 AM	Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	High
9/12/16 8:17 AM	Low Alarm			Cleared, Unacknowledged	Critical
9/15/16 8:46 AM	High Alarm			Cleared, Unacknowledged	Critical
9/15/16 8:46 AM	High Alarm			Cleared, Unacknowledged	Critical
9/15/16 10:00 AM	High Alarm			Cleared, Unacknowledged	Critical
9/15/16 10:00 AM	High Alarm			Cleared, Unacknowledged	Critical
9/15/16 11:01 AM	Low Alarm			Cleared, Unacknowledged	Critical
9/15/16 11:01 AM	High Alarm			Cleared, Unacknowledged	Critical
9/15/16 11:03 AM	Low Alarm			Cleared, Unacknowledged	Critical
9/15/16 11:01 AM	High Alarm			Cleared, Unacknowledged	Critical
9/15/16 11:03 AM	High Alarm			Cleared, Unacknowledged	Critical
9/15/16 12:56 PM	High Alarm			Cleared, Unacknowledged	Critical
9/15/16 11:01 AM	Fault1			Cleared, Unacknowledged	High

In many production environments, a small Alarm Status Table is displayed at the very bottom of the Ignition client window taking up very little space. The problem is when an operator has quite a few Active alarms, they have to manually scroll through the list of alarms to see them all when the Marquee Mode property is not enabled. This can be a little tedious for operators.

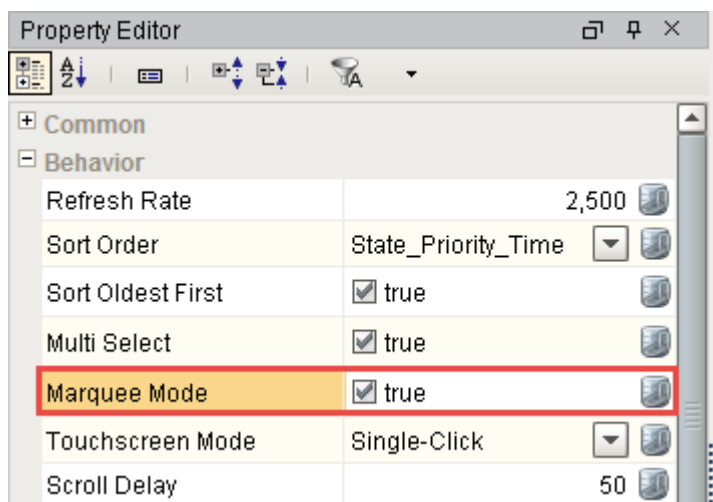
☐	Active Time	Display Path	Name	Current State	Priority
☐	9/12/16 8:17 AM	gensim/Writeable/Write...	Fault	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Writeable/WriteableInte...	Alarm	Active, Unacknowledged	Critical
☐	9/13/16 4:01 PM	Machine D	High Alarm	Active, Unacknowledged	Critical
☐	9/13/16 4:01 PM	Machine A	High Alarm	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Alarms/ProdSupAttentio...	Alarm	Active, Unacknowledged	High
☐	9/14/16 4:17 PM	Ramp	High Alarm 2	Active, Unacknowledged	Medium
☐	9/13/16 12:08 PM	gensim/Writeable/Writea...	High Alarm	Cleared, Unacknowledged	Critical
☐	9/13/16 12:12 PM	Machine A	High Alarm	Cleared, Unacknowledged	Critical

Acknowledge Shelve

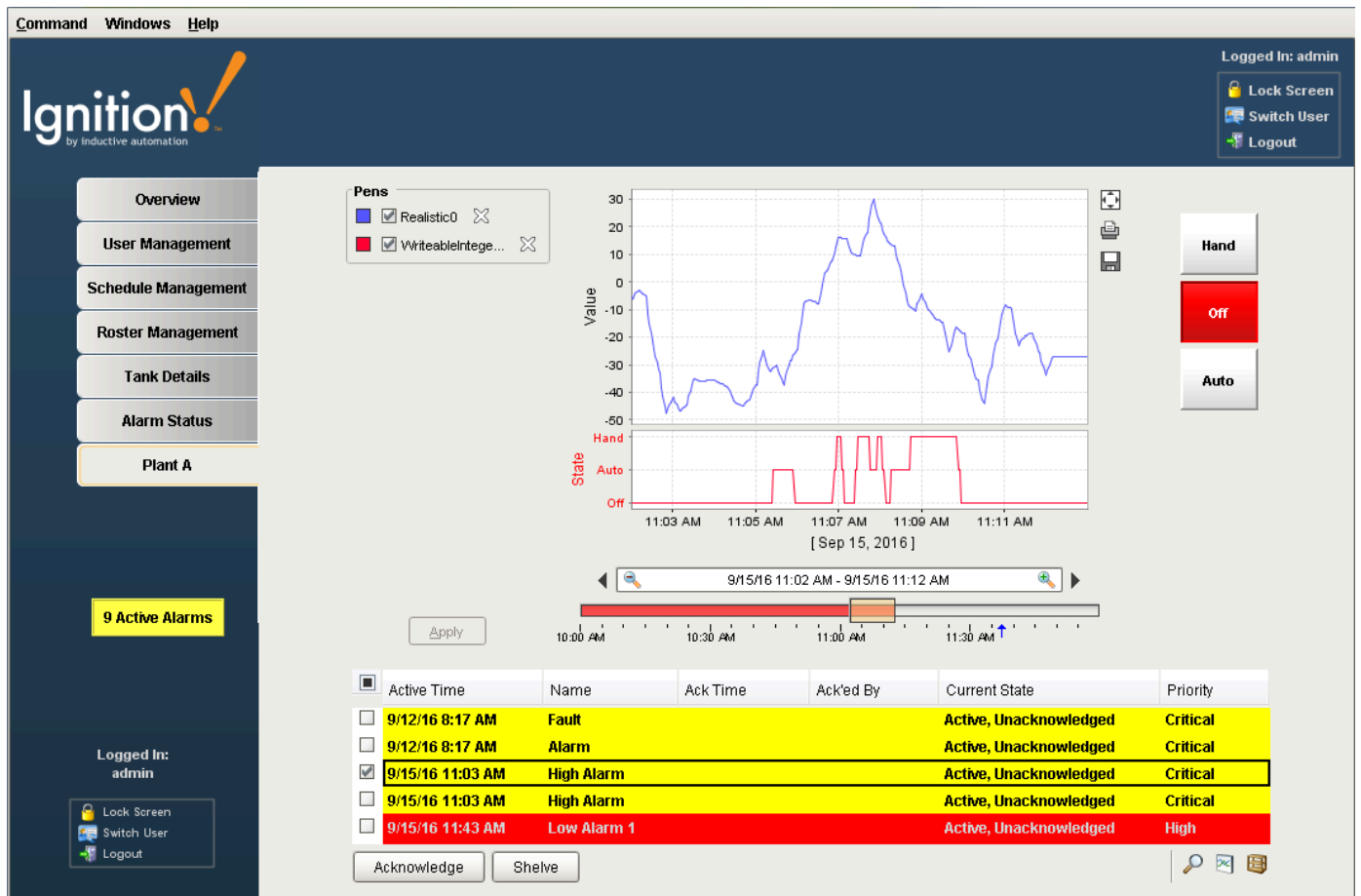
To Setup Marquee Mode

The nice feature with Marquee Mode, is it automatically scrolls through all the alarms when there is not enough vertical space on the client window to show them all at once.

1. Select the Alarm Status Table. In the **Property Editor** of the Alarm Status Table, there is a property called **Marquee Mode**. Set the property to **true**.



2. This client screen shows you a typical Alarm Status Table using the Marquee Mode located at the bottom of the window. The Alarm Status Table automatically scrolls through all alarms that are currently visible. Marquee Mode makes it very easy for operators to see what's going on without having to manually scroll through the entire list of alarms to identify all the alarms that are Active, or whatever state they are looking at.



ⓘ PAY ATTENTION TO FILTERS AND INFORM OPERATORS THAT IT'S A SCROLLING LIST OF ALARMS

When using Marquee mode, it is even more important to pay attention to the filters on the Alarm Status Table since there is limited room. Make sure your operators know that even though the Alarm Status Table may be small, when Marquee Mode is enabled that it is a scrolling list.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status - Acknowledgement



INDUCTIVE
UNIVERSITY

**Alarm Status -
Acknowledgement**

[Watch the video](#)

Using Alarm Acknowledgement

One of the most important things an operator is going to do in the client is acknowledge alarms. Alarm acknowledgement is built-in to the Alarm Status Table component. As soon as the operator selects and presses the Acknowledge button, the current state of the alarm will change, and the operator's credentials and the time the alarm was acknowledged will be recorded in the Alarm Status Table.

Ignition
by Inductive Automation

Command Windows Help

Logged In: admin

- Lock Screen
- Switch User
- Logout

Overview

User Management

Schedule Management

Roster Management

Tank Details

Alarm Status

8 Active Alarms

Logged In:
admin

- Lock Screen
- Switch User
- Logout

	Active Time	Name	Ack Time	Ack'd By	Current State	Priority
☐	9/12/16 8:17 AM	Fault			Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Alarm			Active, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Active, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Active, Unacknowledged	Critical
☒	9/12/16 8:17 AM	High Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	Critical
☒	9/12/16 8:17 AM	Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	High
☐	9/13/16 12:12 PM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/13/16 2:54 PM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐	9/13/16 4:01 PM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/13/16 4:01 PM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:45 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 9:13 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 1			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 1			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 1			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 1			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 1			Cleared, Unacknowledged	High
☐	9/15/16 9:14 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/12/16 8:17 AM	Low Alarm			Cleared, Unacknowledged	Medium
☐	9/12/16 8:17 AM	:pw Alarm			Cleared, Unacknowledged	Medium

Acknowledge Shelve [x2]

Search Filter

If you don't have the Ack'ed By or Ack Time columns in the header of the Alarm Status Table, you can simply add them in the Designer in **Preview Mode** as well as in the **Client**. Right click on the header bar, and from the dropdown list check **Acke'ed** and **Ack Time**, and any other information you want to display or remove from the table.

☐	Active Time	Name	Current State	Priority
☐	9/12/16 8:17 AM	Fault	Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	Active, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm	Active, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm	Active, Unacknowledged	Critical
☐	9/15/16 8:49 AM	Low Alarm 2	Active, Unacknowledged	High
☐	9/15/16 8:49 AM	Low Alarm 1	Active, Unacknowledged	High
☐	9/12/16 8:17 AM	High Alarm	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	Active, Acknowledged	High
☐	9/13/16 12:12 PM	High Alarm	Cleared, Unacknowledged	Critical
☐	9/13/16 2:54 PM	High Alarm	Cleared, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Low Alarm	Cleared, Unacknowledged	Critical
☐	9/13/16 4:01 PM	High Alarm	Cleared, Unacknowledged	Critical
☐	9/13/16 4:01 PM	High Alarm	Cleared, Unacknowledged	Critical
☐	9/15/16 8:45 AM	High Alarm	Cleared, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm	Cleared, Unacknowledged	Critical
☐	9/15/16 8:48 AM	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/15/16 8:48 AM	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/15/16 8:48 AM	Low Alarm 1	Cleared, Unacknowledged	High
☐	9/15/16 8:48 AM	Low Alarm 2	Cleared, Unacknowledged	High
☐	9/15/16 8:48 AM	Low Alarm 1	Cleared, Unacknowledged	High
☐	9/15/16 8:48 AM	Low Alarm 1	Cleared, Unacknowledged	High
☐	9/15/16 8:48 AM	Low Alarm 2	Cleared, Unacknowledged	High

☐ Ack Notes
☒ Ack Time
☒ Ack'ed By
☐ Active Pipeline
☐ Clear Pipeline
☐ Ack Pipeline
☒ Active Time
☐ Clear Time
☐ Event Value
☐ Deadband
☐ Display Path
☐ Acked?
☐ Active?
☐ Clear?
☒ Name
☐ Notes
☒ Priority
☐ Source Path
☒ Current State
☐ eventId

Acknowledge
Shelve

The Alarm Status Table component allows you to select an individual alarm, multiple alarms or use the select all checkbox in the header bar. To Acknowledge alarms, check all the alarms you want to acknowledge, and press the Acknowledged button. In this example, two alarms were checked, and two alarms were Acknowledged at the same time. The Alarm Status Table will record the time the alarm was acknowledged and who acknowledged the alarm (currently logged in user).

☐	Active Time	Name	Ack Time	Ack'd By	Current State	Priority
☐	9/12/16 8:17 AM	Fault			Active, Unacknowledged	Critical
☒	9/12/16 8:17 AM	Alarm			Active, Unacknowledged	Critical
☒	9/15/16 8:46 AM	High Alarm			Active, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Active, Unacknowledged	Critical
☐	9/15/16 8:46 AM	Low Alarm 1			Active, Unacknowledged	High
☐	9/15/16 8:46 AM	High Alarm 2			Active, Unacknowledged	Medium
☐	9/12/16 8:17 AM	High Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	High
☐	9/13/16 12:12 PM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/13/16 2:54 PM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/12/16 8:17 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐	9/13/16 4:01 PM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/13/16 4:01 PM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:45 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:45 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/15/16 8:46 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/15/16 8:46 AM	Low Alarm 1			Cleared, Unacknowledged	High
☐	9/15/16 8:46 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/15/16 8:46 AM	Low Alarm 1			Cleared, Unacknowledged	High
☐	9/15/16 8:46 AM	Low Alarm 2			Cleared, Unacknowledged	High
☐	9/15/16 8:46 AM	Low Alarm 1			Cleared, Unacknowledged	High

Acknowledge [x2]

Shelve [x2]



If any of the selected alarms require **Acknowledgement Notes**, a small window is displayed in which the operator is required to add notes, otherwise, the alarm cannot be acknowledged. If the operator wants to cancel the Acknowledgement Notes and close the window, use the **Expand button**  to close the Acknowledgement Notes window.

Acknowledgement Notes are setup on the alarm's configuration settings. To setup Acknowledgement Notes, go to your alarm configuration settings and set '**Ack Notes Required**' to '**true**'. For more information refer to [Configuring Alarms](#).

☐	Active Time	Name	Ack Time	Ack'd By	Current State	Priority
☐	9/12/16 8:17 AM	Fault			Active, Unacknowledged	Critical
☒	9/15/16 12:56 PM	High Alarm			Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	High Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	Critical
☐	9/15/16 11:03 AM	High Alarm	9/15/16 11:57 AM	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 12:55 ...	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	High
☐	9/12/16 8:17 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 10:00 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	Acknowledgement	High Alarm			Cleared, Unacknowledged	Critical

Calibrated equipment. Issued resolved.

Acknowledge

Security for Alarm Acknowledgement

If you want to restrict who can use the Acknowledge button, there is the **Show Ack Button** in the Property Editor that can be set to **'False'**. By setting the **Show Ack Button** to **'false'**, this hides the Acknowledge button on the Alarm Status Table.

In order for operators to acknowledge alarms, the correct permission must be assigned. Permissions can be assigned using the Show Ack Button. Click the **Show Ack Button binding icon** to open the property binding window. This example shows how to set permissions to acknowledge alarms for people in the **Operator Role**. You can setup permissions for any role, user and user source in your system.

1. In the Designer, select the Alarm Status Table component, and under **Property Binding Type**, select **Expression**.
2. Click the **Function icon** and scroll down to **Users**, and select **'hasRole'**. This enters the function name.
3. Edit the expression to read: **hasRole("Operator")**

4. Click **OK**. If you currently have the "Operator" role, you will notice that the **Show Ack Button** property is now 'true', otherwise it will be 'false'.

The screenshot displays the Ignition software interface. On the left, the **Property Editor** shows the **Show Ack Button** property set to **true**. The **Main Window** displays a table with alarm data. A **Property Binding** dialog is open, showing the **Expression** `1 hasRole("Operator")` and the **Users** category in the **Advanced** section.

Project Browser

- Alarm Status
 - Root Container
 - Alarm Status Table

Tag Browser

- WritableFloat1 (0, Float4)
- WritableFloat2 (106, Float4)
- WritableInteger1 (0, Int4)

Property Editor

Property	Value
Number Format	#,##0.###
Date Format	
Table Background	255,255,255
Selection Color	0,0,0
Selection Thickness	2
Flash Interval	500
Show Table Header	☒ true
Show Header Popup	☒ true
Table Font	Dialog, Plain, 12
Table Header Font	Dialog, Plain, 12
Row Height	22
Show Footer	☒ true
Show Ack Button	☒ true
Show Shelf Button	☒ true
Show Details Button	☒ true
Show Chart Button	☒ true
Show Manage Shelf Button	☒ true
Row Styles	Dataset [7R x 8C]
Notes Area Location	Hidden

Main Window

Display Path	Name	Active Time	Ack'd By	Ack Time	Current State	Priority
Machine D	High Alarm	4/4/16 2:41 PM			Active, Unacknowledged	Critical
Ramp	Low Alarm 1	4/5/16 8:48 AM			Active, Unacknowledged	High
		31/16 9:35 AM			Active, Acknowledged	Critical
		31/16 9:36 AM			Active, Acknowledged	Critical
		31/16 9:36 AM			Active, Acknowledged	High
					Cleared, Unacknowledged	Critical
					Cleared, Unacknowledged	Critical
					Cleared, Unacknowledged	Critical

Property Binding: Root Container.Alarm Status Table

Binding Types

- Tag
 - Tag
 - Indirect Tag
- Property
 - Property
 - Expression
- Database
 - DB Browse
 - SQL Query
- No Binding

Expression

```
1 hasRole("Operator")
```

Options

- ☐ Overlay Opt-Out

Advanced

- Aggregates
- Alarming
- Colors
- Date and Time
- Logic
- Math
- Strings
- Translation
- Type Casting
- Users
 - hasRole(role, [username], [usersource]) returns Boolean

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status - Shelving



INDUCTIVE
UNIVERSITY

Alarm Status - Shelving

[Watch the video](#)

Another important feature of the **Alarm Status Table** is the capability to shelve **alarms**. Shelving alarms allows you to temporarily silence an alarm for a fixed period of time. This feature is extremely handy when an alarm is already active and you want to temporarily suppress the alarm while you're working on the issue. The Alarm Status Table component will not send any notifications while the alarm is shelved, and will be temporarily dropped from the Alarm Status list so operators don't get confused and think it's active. When the shelved time period is up and if the alarm is still active, it will return into the Alarm Status list.

ACTIVE ALARMS MUST BE PRESENT IN THE SYSTEM TO SHELVE AN ALARM

Active alarms must be present in the Alarm Status Table before you can shelve an alarm.

How to Shelve an Alarm

To shelve an alarm, select one alarm or multiple alarms, then press the **Shelve button**. The **Shelving tab** will automatically open so you can set a duration to silence an alarm. You can set a duration from 5 minutes to 4 hours in which to shelve selected alarm(s). Select an alarm, choose **30 minutes**, and press **Apply**. You can test out your shelved alarm in the Designer in **Preview Mode**.

☐	Active Time	Name	Ack Time	Ack'd By	Current State	Priority
☐	9/12/16 8:17 AM	Fault			Active, Unacknowledged	Critical
☒	9/15/16 12:56 PM	High Alarm			Active, Unacknowledged	Critical
☐	9/15/16 4:12 PM	High Alarm			Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	High Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 12:55 ...	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	High
☐	9/12/16 8:17 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 10:00 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 10:00 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 11:01 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 11:01 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 11:03 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 11:01 AM	High Alarm			Cleared, Unacknowledged	Critical

Shelving

Shelve the selected alarm(s) for:

5 minutes

5 minutes

15 minutes

30 minutes

1 hours

2 hours

4 hours

Apply

Shelve

To view all the alarms that are currently shelved, go to the bottom right of the Alarm Status Table and click on the **Shelved Alarms icon** . A Shelved Alarm tab will open and show all of the shelved alarms, as well as when the shelved time period expires. To close the Shelved Alarm tab, click either the **Shelved Alarm icon** or the **Expand** button .

After the amount of time expires on a shelved alarm, it will be evaluated, and if it is still active, it will automatically return to the Alarm Status list. If the alarm transitions to a cleared state during the time shelved period (i.e., tank water level is low and when the water level returns to the setpoint), the alarm will show up as 'Cleared' in the Alarm Status list instead of 'Active'.

☐	Active Time	Name	Ack Time	Ack'd By	Current State	Priority
☐	9/12/16 8:17 AM	Fault			Active, Unacknowledged	Critical
☐	9/15/16 4:12 PM	High Alarm			Active, Unacknowledged	Critical
☐	9/12/16 8:17 AM	High Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 12:55 ...	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	High
☐	9/12/16 8:17 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 8:46 AM	High Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 10:00 AM	High Alarm			Cleared, Unacknowledged	Critical
☐		Low Alarm			Cleared, Unacknowledged	Critical

Shelved Alarms

Path	Shelved By	Expires
prov:default/tag:gensim/Ramp/Ramp2/alm:Low Alarm 2	admin	27 minutes, 20 seconds
prov:default/tag:gensim/Writeable/WriteableFloat1/alm:High Alarm	admin	2 minutes, 3 seconds

[Unshelve](#)

To unshelve an alarm, **select the alarm(s)** under the **Shelved Alarms** tab, click the **Unshelved** link, and the alarm will return back to the Alarm Status list.

☐	Active Time	Name	Ack Time	Ack'd By	Current State	Priority
☐	9/12/16 8:17 AM	Fault			Active, Unacknowledged	Critical
☐	9/16/16 9:56 AM	Low Alarm 1			Active, Unacknowledged	High
☐	9/16/16 9:56 AM	High Alarm 2			Active, Unacknowledged	Medium
☐	9/12/16 8:17 AM	High Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 12:55 ...	admin	Active, Acknowledged	Critical
☐	9/12/16 8:17 AM	Alarm	9/15/16 8:24 AM	admin	Active, Acknowledged	High
☐	9/12/16 8:17 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐	9/15/16 11:01 AM	Low Alarm			Cleared, Unacknowledged	Critical
☐		Low Alarm			Cleared, Unacknowledged	Critical

Shelved Alarms

Path	Shelved By	Expires
prov:default/tag:gensim/Writeable/WriteableFloat2/alm:High Alarm	admin	14 minutes, 6 seconds
prov:default/tag:gensim/Writeable/WriteableFloat1/alm:High Alarm	admin	14 minutes, 6 seconds

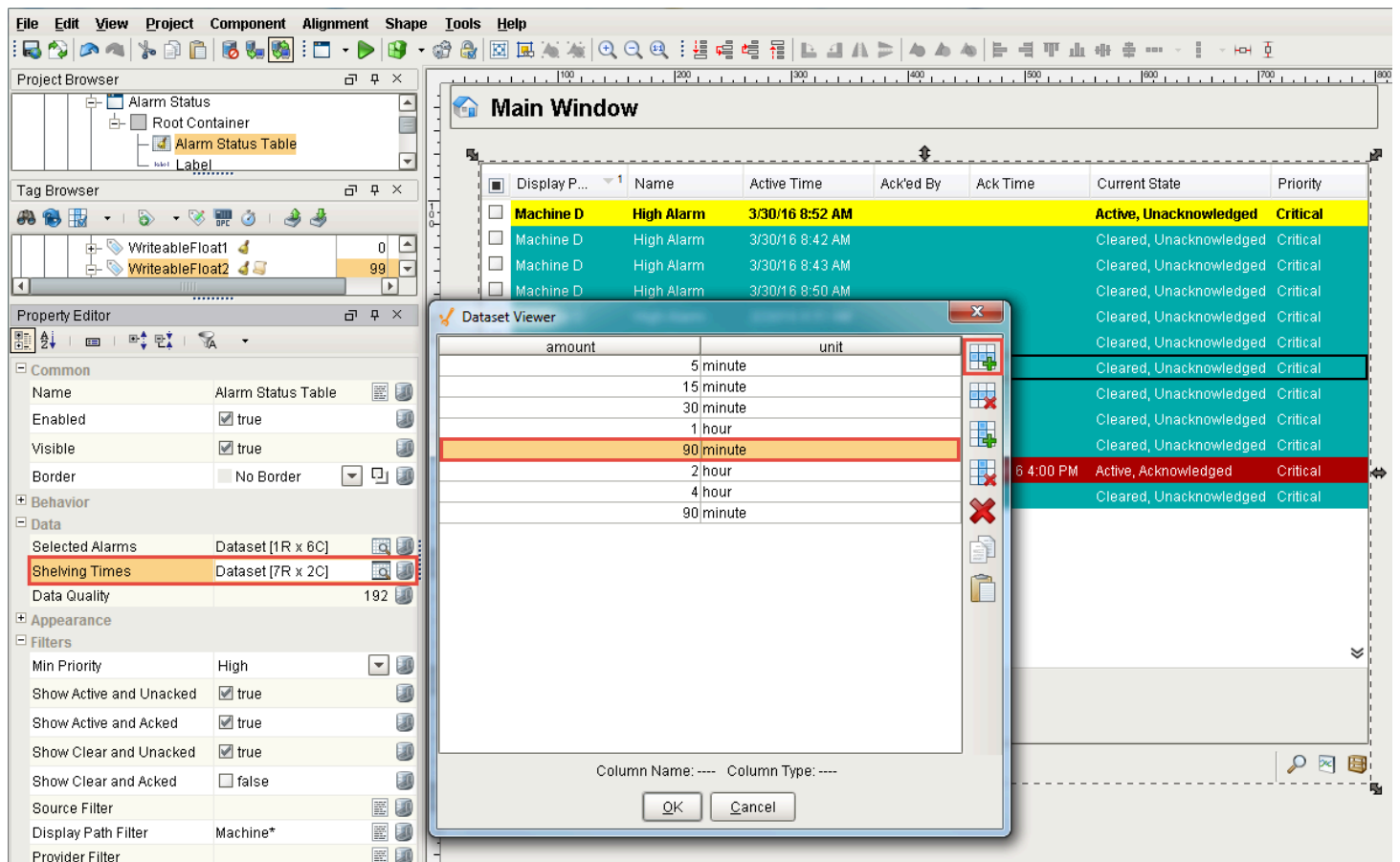
[Unshelve](#)

Custom Shelving Duration

The shelving duration values on the Alarm Status Table component can be customized. By default, the component has several shelving preset times, but you can add or remove times based on your requirements. One thing to note with modifying the shelving times, is that there is no option to set a shelving time indefinitely. Shelving times are meant to be temporary, not long term.

In the Property Editor, there is a property called **Shelving Times**. This is where you can add, remove, or change the shelving times for alarms.

1. In the Designer, select the Alarm Status Table component, and click on the **DatasetViewer** icon to open the Dataset Viewer.
2. Click the green plus row icon  to insert a new row. Enter the hour and the unit of time.
3. Press **OK**. In this example, you can see that **90 minutes** was added to the list of shelving times.



To verify your new time was entered correctly, select the alarm(s) you want to shelve and click the Shelve button. Choose the shelve time from the dropdown list and press **Apply** to temporarily silence your alarm(s). You can see how easy it is to modify your shelved times.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Alarm Status
 - Root Container
 - Alarm Status Table
 - Label

Tag Browser

- WriteableFloat1
- WriteableFloat2

Property Editor

Common

Name Alarm Status Table

Enabled ☒ true

Visible ☒ true

Border No Border

Behavior

Data

Selected Alarms Dataset [2R x 6C]

Shelving Times Dataset [7R x 2C]

Data Quality 192

Appearance

Filters

Min Priority Medium

Show Active and Unacked ☒ true

Show Active and Aacked ☒ true

Show Clear and Unacked ☒ true

Show Clear and Aacked ☐ false

Source Filter

Display Path Filter *Machine*

Provider Filter

Main Window

Display P...	Name	Active Time	Ack'ed By	Ack Time	Current State	Priority
☒	Machine D	High Alarm	3/30/16 8:52 AM		Active, Unacknowledged	Critical
☐	Machine D	High Alarm	3/30/16 8:42 AM		Cleared, Unacknowledged	Critical
☐	Machine D	High Alarm	3/30/16 8:43 AM		Cleared, Unacknowledged	Critical
☐	Machine D	High Alarm	3/30/16 8:50 AM		Cleared, Unacknowledged	Critical
☐	Machine D	High Alarm	3/30/16 8:51 AM		Cleared, Unacknowledged	Critical
☐	Machine C	Low Alarm	3/28/16 2:38 PM		Cleared, Unacknowledged	Critical
☐	Machine C	Low Alarm	3/28/16 2:52 PM		Cleared, Unacknowledged	Critical
☐	Machine C	Low Alarm	3/30/16 8:21 AM		Cleared, Unacknowledged	Critical
☐	Machine C	Low Alarm	3/30/16 8:42 AM		Cleared, Unacknowledged	Critical
☐	Machine C	Low Alarm	3/30/16 8:43 AM		Cleared, Unacknowledged	Critical
☐	Machine B	High Alarm	3/21/16 8:12 AM	admin 3/25/16 4:00 PM	Active, Acknowledged	Critical
☐	Machine B	:pw Alarm	3/21/16 8:05 AM	admin 3/28/16 10:15 AM	Active, Acknowledged	Medium
☐	Machine B	Low Alarm	3/28/16 8:57 AM	admin 3/28/16 10:15 AM	Active, Acknowledged	Medium
☒	Machine B	High Alarm	3/21/16 8:05 AM		Cleared, Unacknowledged	Critical
☐	Machine B	Low Alarm	3/21/16 8:12 AM		Cleared, Unacknowledged	Medium
☐	Machine B	Low Alarm	3/22/16 11:45 AM		Cleared, Unacknowledged	Medium

Shelving

Shelve the selected alarm(s) for:

5 minutes Apply

5 minutes
15 minutes
30 minutes
1 hours
90 minutes
2 hours
4 hours

Shelve [x2]

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Status - Tag History and Alarm History



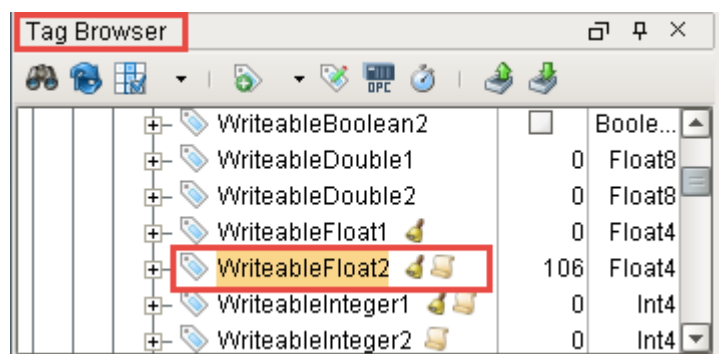
INDUCTIVE
UNIVERSITY

Alarm Status - Tag History and Alarm History

[Watch the video](#)

The **Alarm Status Table** component has a unique feature built into it that lets you view the actual trend of a tag's value and an alarm's history for any alarm that is currently active. It shows you exactly where the alarms occurred, when it was cleared, and when it was acknowledged.

For the trend feature to work, you must have your tag history setup so it's logging to the **Tag Historian** as well as an **Alarm Journal Profile** configured so that the log history gets logged to a database. As long as both the Tag Historian and Alarm Journal are configured, you will see the trend for any given alarm that is currently active or in the Alarm Status Table list.



In the Alarm Status Table, select any **alarm** and click on the **Trend icon**  located on the bottom right of the table. The trend chart shows you the trend of the selected alarm's value over time. You can look at the last 5 minutes, 30 minutes, 1 hour, 8 hours or up to 24 hours to view the exact history of that tag's value, as well as all the alarms that occurred over time. The yellow triangles denote when the alarm was active, the green circles identify when the alarm was cleared, and a vertical black bar

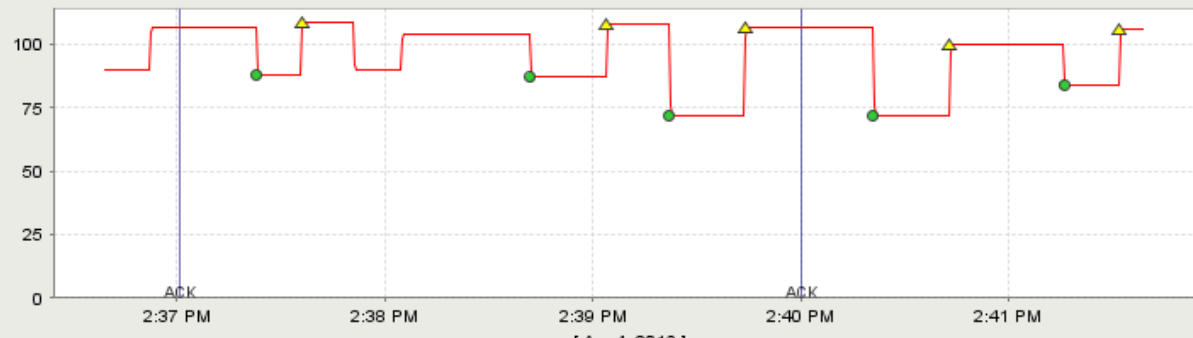
shows when the alarm was acknowledged. It's all built into the Alarm Status Table component and you don't have to go to a separate window to view an alarm trend or alarm history.

 Main Window

☐	Display Path	Name	Active Time	Ack'd By	Ack Time	Current State	Priority
☒	Machine D	High Alarm	4/4/16 2:41 PM			Active, Unacknowledg...	Critical
☐	Ramp	Low Alarm 1	4/4/16 2:41 PM			Active, Unacknowledg...	High
☐	Machine B	ipw Alarm	4/1/16 9:56 AM			Active, Unacknowledged	Medium
☐	Machine B	Low Alarm	4/1/16 9:56 AM			Active, Unacknowledged	Medium
☐	Writeable/WriteableInteger2/A...	Alarm 1	3/31/16 9:28 AM			Active, Unacknowledged	Low
☐	Writeable/WriteableInteger2/A...	Alarm	3/31/16 9:28 AM			Active, Unacknowledged	Low
☐	Alarms/ProdSupAttention/Alar...	Alarm 1	3/31/16 9:28 AM			Active, Unacknowledged	Low
☐	Writeable/WriteableFloat1/Ala...	Alarm	4/1/16 9:56 AM			Active, Unacknowledged	Low
☐	Machine B	High Alarm	3/31/16 9:28 AM	admin	3/31/16 9:35 AM	Active, Acknowledged	Critical
☐	Writeable/WriteableInteger1/A...	Alarm	3/31/16 9:28 AM	admin	3/31/16 9:36 AM	Active, Acknowledged	Critical
☐	Alarms/ProdSupAttention/Alarm	Alarm	3/31/16 9:28 AM	admin	3/31/16 9:36 AM	Active, Acknowledged	High

Value Trend

5m 30m 1h 8h 24h



Acknowledge Shelve

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Using Alarm Status Tags



INDUCTIVE
UNIVERSITY

Using Alarm Status Tags

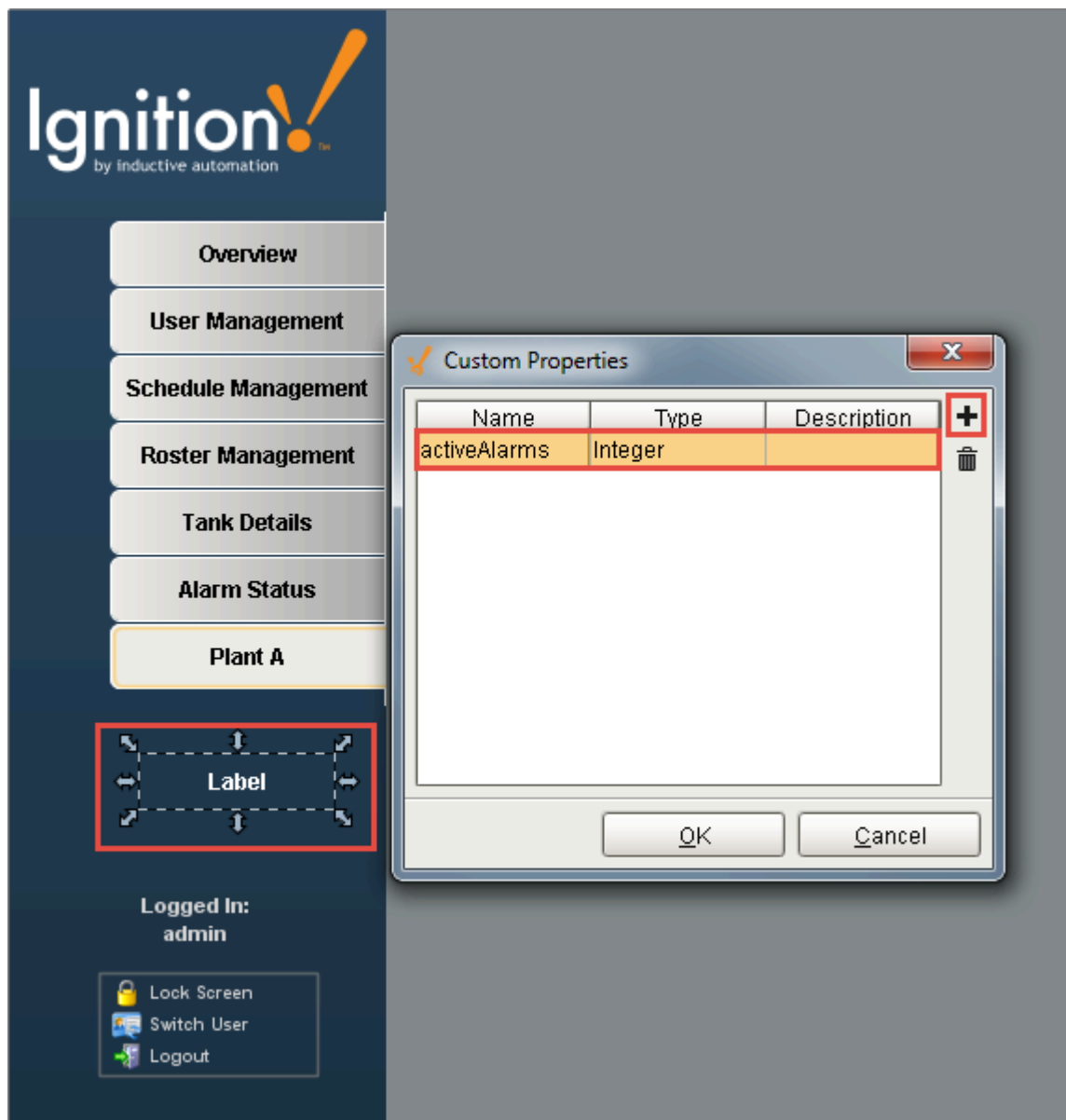
[Watch the video](#)

Ignition displays four **Gateway System Tags** that give you the number of alarms that are active and acknowledged, active and unacknowledged, cleared and acknowledged, and cleared and unacknowledged. A quick way to see if any alarms are active and to get a active alarm count is to add a Label component on a [navigation window](#).

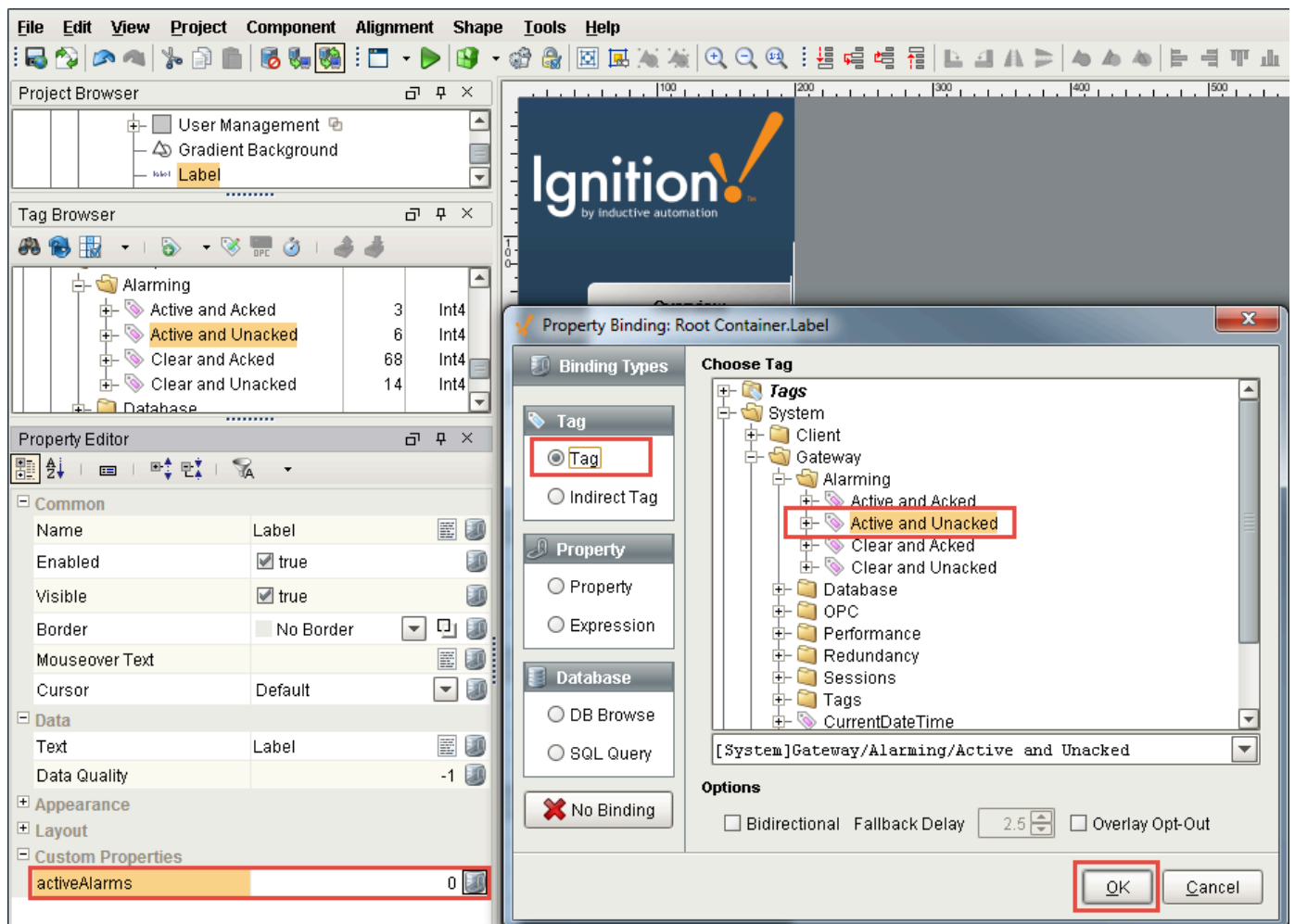
Add a Label with Alarm Status

Let's setup an Active Alarm on the Navigation window using a **Label** component for the **Active and UnAked** tag showing how many alarms are currently active and unacknowledged.

1. In Designer, open your **Navigation window**, **drag in a Label component** and place it under the tab strip.
2. Link the Label component to the Active and Unacked alarm system tag. Right click on the **Label** component, and select on **Customizers > Custom Properties**.
3. Create a new property and name it '**activeAlarms**' with an '**Integer**' data type.
4. Click **OK**.



5. In the **Property Editor** under **Custom Properties**, click the binding icon for the 'activeAlarms' property. The property binding window will open.
6. Next, bind the new custom property 'activeAlarms' to the 'Active and Unacked tag'. In the **Tag Browser**, under **Binding Types**, select the **Tag** radio button, and choose the **System > Gateway > Alarming > Active and Unacked Tag**.
7. Click **OK**.

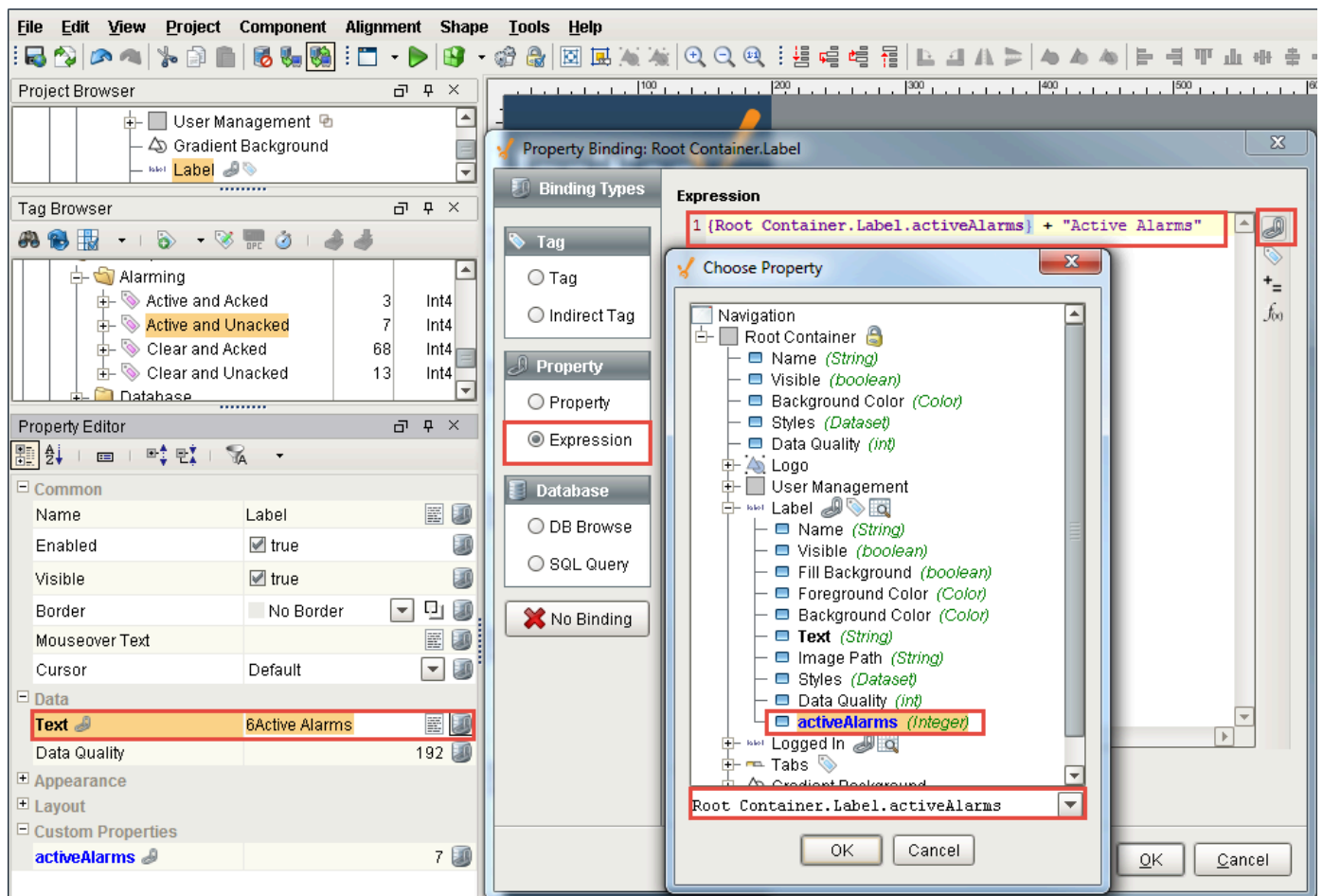


- Now we need to display the number of alarms on the Label that are active. Select the **Label**, and in the **Property Editor**, click the binding icon  for **Text** property.

Under **Binding Types**, select the **Expression** radio button and click **Link** icon  in the upper right corner of the window. Bind the **Text** property of the label and combine the expression with the literal string value '**Active Alarms**' as shown below. Click **OK** twice.

Combine Expression with Literal String Value

```
{Root Container.Label.activeAlarms} + " Active Alarms"
```



9. The last thing to do is setup a style to make the Active Alarms label stand out on the Navigation window. Let's change the color to something bright and make it blink. Right click on the **Label** and select **Customizers > Style Customizer**.
10. For the **Driving Property**, select 'activeAlarm', and for the **Styled Properties**, select **Background Color**, **Foreground Color**, **Border**, and **Visible**, and click on the green arrow pointing to the right  and add them to the **Used Properties** list.
11. Under the **Styles** section, click on the green plus icon  at the bottom 2 times to add 2 **Values**. Click the **Expand buttons**  to open and customize the style for the Active Alarm Label. Set the following values for **Styles**:

Value 0

Background Color: **Red**
 Border: **Etched (Raised)**
 Foreground Color: **White**
 Visible: **check the box**

Value 1

Click the **Animate** box. Use the green plus icon to add 2 **animation steps**. Change the values to:

Step 1 Duration: 1000

Background Color: **Red**

Border: **Etched (Raised)**

Foreground Color: **White**

Visible: **check the box**

Step 2 Duration: 1000

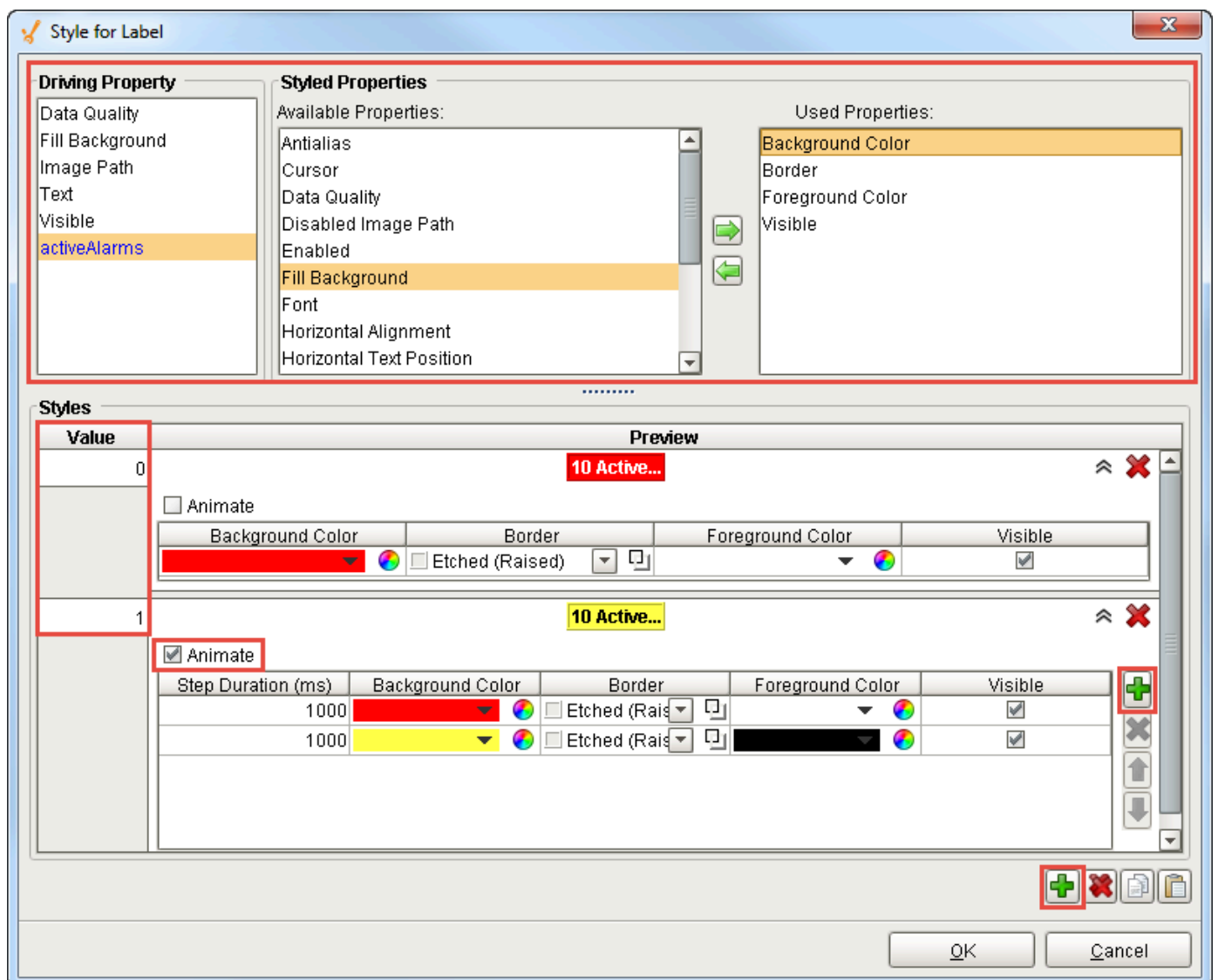
Background Color: **Red**

Border: **Etched (Raised)**

Foreground Color: **White**

Visible: **check the box**

Click **OK**.



Now, you have a label on the Navigation window that blinks when there is an active alarm and provides an active alarm count.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- User Management
- Gradient Background
- Label

Tag Browser

Tag	Value	Data Type
Tags		
Data Types		
Alarms		

Property Editor

Common

Name	Label
Enabled	☒ true
Visible	☒ true
Border	☐ Etched (Raised)
Mouseover Text	
Cursor	Default

Data

Text	7 Active Alarms
Data Quality	192

Appearance

Font	Dialog, Bold, 12
Fill Background	☒ true
Foreground Color	0,0,0
Background Color	255,255,71
Image Path	

Ignition!
by inductive automation

Overview

User Management

Schedule Management

Roster Management

7 Active Alarms

Logged In:
admin

Lock Screen

Switch User

Logout

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Notification

Alarm notification is the act of sending a message to a group of people when an alarm becomes active or cleared. In Ignition, this functionality is enabled by having the Alarm Notification Module installed which provides alarm pipelines and Email notification. The three methods or **Notification Profile Types** of sending alarms is Email, SMS and Voice Notifications.

Configuring Alarm Notification

Alarm Notifications are the culmination of several sub-systems in Ignition. Before alarms can send out any notifications, the following items must be configured:

- **Notification Profile**
- **On-Call Roster**
- **Notification Pipeline**

Notification Profile

The Notification Profile determines what type of notification (Email, SMS, or Voice) is sent out, and contains the necessary information for Ignition to send out the notification. Each alarm notification type can be configured for both one-way and two-way notifications. Details on each type of Notification Profile can be found on the **Notification Profile Types** page.

On-Call Roster

An On-Call Roster is a collection of users that are notified when an alarm occurs. These users can be taken from multiple **User Sources**. When an alarm is triggered, it is sent to a designated On-Call Roster where it evaluates the users schedules, and only notifies those users that have an active schedule. Users that are off-schedule will not be notified. More Information can be found on the **User, Schedule, and Roster Management** page.

Notification Pipelines

Notification Pipelines define the logic of what should happen between an alarm becoming active (or clear), and people being notified. Pipelines are configured in the Designer, but you'll need to configure at least one alarm notification profile and at least one roster in the Gateway before you can configure a valid pipeline.

More Details on configuring Pipelines can be found on the [Alarm Notification Pipelines](#) page.

Notification Testing

Before assigning Pipelines to alarms, the notification system can inject test alarms to help troubleshoot any problems. A notification will be sent out to any On-Call Rosters in the selected pipeline, so it is recommended to either enable test mode on any Notification blocks, or create a testing-only roster.

To test alarms, follow these simple steps:

1. Go to the Gateway webpage and select the **Configure** section.
2. From the menu on the left, select **Alarming > Notification**. The Alarm Notification Profiles page will open.
3. Simply click the **Test Pipelines and Notification Profiles...** link. The link will navigate you to a testing page.

The screenshot shows the Ignition Gateway web interface. The top navigation bar includes 'Controller', 'USER MANUAL', 'SUPPORT', 'admin', 'Sign Out', and a 'Launch Designer' button. The main navigation menu on the left has sections for SYSTEM, NETWORKING, and SECURITY. The 'CONFIGURE' section is active, and the 'Alarm Notification Profiles' page is displayed. A green banner at the top of the page indicates 'License Incomplete' with a timer at 0:00:00 and buttons for 'Reset Trial' and 'View Modules'. A success message states 'Successfully deleted Alarm Notification Profile "Ignition-ubuntu_EMail"'. Below this is a table of active profiles:

Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	delete edit
SMS		true	SMS Notification	Running	delete edit

At the bottom of the page, there are two links: 'Create new Alarm Notification Profile...' and 'Test Pipelines and Notification Profiles...'. The latter link is highlighted with a red rectangle.

4. Select your configured Pipeline, and click the **Submit** button.

Pipeline/Notification Test Page

 Use this page to inject a fake alarm event into a pipeline to test out your pipeline and notification profile settings.

Pipeline

Escalate

Display Path

TestPage/TEST_ALARM

Priority

Diagnostic

Submit

5. Once the system has been tested successfully, the last step is to assign a pipeline to your alarms. This can be done by [editing your Tags in the Designer](#). Go to your **Tag Browser** and double click on your tag. The **Tag Editor** will open, and click on **Alarming**. Select an alarm and go to the Notification section, and choose a pipeline (i.e., Escalate) to assign to your alarm. Click **OK**.

Tag Editor

General

Numeric

Metadata

Permissions

History

Alarming

Tag Events

Alarm Configuration

Fault1 - Equal, High

High Alarm - Above Setpoint, Critical

Low Alarm - Below Setpoint, Medium

Fault1

Deadband

0

Deadband Mode

Absolute

Active delay (seconds)

0

Clear delay (seconds)

0

Notification

Active Pipeline

Escalate

Clear Pipeline

Ack Pipeline

Email Notification Properties

Custom Subject

Active Pipeline

The pipeline that will be used to process active events generated by this alarm.

OK

Apply

Cancel

Now, when an alarm occurs, an alarm notification will be sent to the users in the On-Call Roster whose schedule is active.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Notification Profile Types

Once you have alarms configured in Ignition, you can notify people when alarms occur. Ignition provides three Notification Profile Types: Email, SMS or text message, and Voice through a phone call in Ignition. All three of these notification profiles can be configured to be two-way which can be acknowledged via email, or by text message, or by pressing 1 on the phone. Here is a brief description of each Notification Profile Type:

- **Email Notification Profile** - simply sends an email containing the alarm notification message. The two-way sends an email containing links that allow the user to acknowledge the alarm.
- **SMS Notification Profile** - sends a text message to users notifying them of an alarm, and a link to acknowledge the alarm.
- **Voice Notification Profile** - delivers alarm notifications to users via telephone. Acknowledgement is provided as a voice instruction as part of the call.

Details on how to create and configure each of these profiles are found on the following sub-pages.

Module Requirements

Each Notification Profile requires the **Alarm Notification Module**. This is because Notification Pipelines are managed and defined by the Alarm Notification Module. The **SMS Notification Module** and **Voice Notification Module** are similar in concept to browser plug-ins in that they extend the functionality of another piece of software. The SMS Notification and Voice Notification are both separate modules, and can be downloaded from the [Ignition Downloads webpage](#). To learn more about installing these modules, refer to [Installing or Upgrading a Module](#).

Additional Requirements

Each Alarm Notification Profile needs a method to deliver the message. These methods are not included with the Alarm Notification Module, and some may require additional hardware to be purchased.

- **Email Notification** - The Gateway will need network access to an SMTP server.

- **SMS Notification** - A cellular modem with an active SIM card. Alternatively, a Twilio account may be used instead of the physical cellular modem.
- **Voice Notification** - Any SIP compatible phone system.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Email Notification Profile



INDUCTIVE
UNIVERSITY

Email Notification Profile

[Watch the video](#)

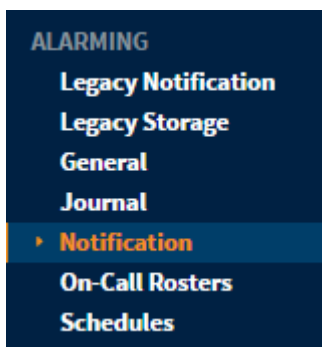
Alarm notification is the act of sending an email message to a group of people when an alarm becomes active or cleared. In Ignition, this functionality is enabled by having the Alarm Notification Module installed which provides alarm pipelines and email notification.

Once you have your Email Notification Profile created in Ignition, you can notify people when alarms occur via Email. If the two-way setting is enabled, operators can also acknowledge the alarm by clicking the link in the body of the email.

Let's get started setting up an Email Notification Profile.

How to Create an Email Notification Profile

1. Go to the **Configure** section of the Gateway webpage.
2. Select **Alarming** > **Notification** from the menu on the left side. The Alarm Notification Profiles page will be displayed.



3. Click on the **Create new Alarm Notification Profile** link.

Alarm Notification Profiles

Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	delete edit
SMS		true	SMS Notification	Running	delete edit

→ [Create new Alarm Notification Profile...](#)

→ [Test Pipelines and Notification Profiles...](#)

4. A list of available alarm notification profiles appear. If any of the profiles are not displayed, it means the required module is missing, or faulted. (To verify that the Alarm Notification Module is installed and running, click on **Modules** from the menu bar on the left side). Choose **Email Notification**, and click **Next**.

Add Alarm Notification Profile Step 1: Choose Type

☒ **Email Notification**

Send alarm notifications via email.

☐ **Remote Gateway Notification**

Enables alarm notification using notification profiles of a remote gateway. Will also expose the pipelines of the remote gateway directly to tags.

☐ **SMS Notification**

Send alarm notifications via SMS using your Airlink Raven XE.

☐ **VOIP Voice Notification**

Telephone notification using VOIP, compatible with most SIP based telephony systems.

[Next >](#)

5. Configure the settings for the Email Notification Profile, and then click **Create New Alarm Notification Profile**.
- Enter a **Name**: Email 1
 - Click the **Enabled** box to enable Email notifications.
 - Enter the **Hostname**. This is the hostname of the **SMTP server** that will be responsible for sending the emails.
 - Enter the **Port**. The default is **Port 25**, but your mail server may be different.
 - Enable **SSL/TLS** if you are using encryption.
 - Enter your **Username** and **Password** if they are required.

- Click the **Two-way Enable** box if you want to make your Email Notification Profile two-way so that the operator can acknowledge the email. There is a link to acknowledge the alarm in the email.

i NOTE

You must have a connection back to the Ignition Gateway. So, if you are working from home and not on a VPN, you will either have to get on a VPN or expose your Ignition server on the internet. You can do this with port forwarding, talk to your IT department about it.

- **Auditing:** You have the option of attaching emails to an Audit profile so that Ignition can store all the events associated with the Email Notification Profile. (I.e., who is receiving alarm and who is acknowledging an alarm).

New Alarm Notification Profile

Main	
Name	Email 1
Description	
Enabled	☒ (default: true)

Email Settings	
Use SMTP Profile?	☐ If selected, this notification profile will use one of the gateway defined SMTP profiles. Otherwise, it will use the settings defined here. (default: false)
SMTP Profile	- none - If Use SMTP Profile is selected, alarm notifications will be emailed using this profile.
Hostname	mail@inductiveautomation.com Hostname of the SMTP server to send email through. Only used when Use SMTP Profile is false.
Port	25 Port SMTP service is running on. Only used when Use SMTP Profile is false. (default: 25)
Enable SSL/TLS	☐ Connect using SSL/TLS. Only used when Use SMTP Profile is false. (default: false)
Username	user@inductiveautomation.com Only used when Use SMTP Profile is false.
Password	
Password	 Re-type password for verification.

Two-way Settings

Two-way Enabled

☒
(default: false)

Gateway

Address and port this gateway is reachable at. Will be used in notification emails. Example: 1.2.3.4:8088

Send HTTPS Link

☐
(default: false)

Auditing

Audit Profile

- none -

If an audit profile is selected, events such as emails and acknowledgements will be stored to the audit system. Note that alarm acknowledgements are also stored to the alarm journal.

☐ Show advanced properties

Create New Alarm Notification Profile

6. Once you create your Email Notification Profile, it will appear in the Alarm Notification Profiles list, and it will be running. You can edit the profile at any time to change any of the settings, and create as many Email Notification Profiles as you need.

Alarm Notification Profiles

✓ Successfully created new Alarm Notification Profile "Email 1"

Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	deleteedit
Email 1		true	Email Notification	Running	deleteedit
SMS		true	SMS Notification	Running	deleteedit

**TIP**

Once your Notification profile is created, you can use it in an [Alarm Pipeline](#).

Email Notification Profile Settings

Email Settings

These settings specify the SMTP server that should deliver the emails. Multiple Ignition Gateways may be configured to use the same SMTP server.

Property Name	Description
Use SMTP Profile?	If selected, this notification profile will use one of the Gateway-defined SMTP profiles, Otherwise, it will use the settings defined here. The setting should NOT be enabled unless an SMTP Profile has been configured elsewhere on the Gateway. Enabling this property causes the profile to ignore the Hostname and Port properties listed below.
SMTP Profile	If the Use SMTP Profile property is enabled, alarm notifications will be emailed using this profile.
Hostname	Hostname of the SMTP server to send email through. Only used when Use SMTP Profile is false.
Port	Port SMTP service is running on. Only used when Use SMTP Profile is false.
Enable SSL/TLS	Connect Using SSL/TLS. Only used when Use SMTP Profile is false.
Username	The username Ignition should use when communicating with the SMTP server. This is only required if the SMTP server expects authentication. Only used when Use SMTP Profile is false.
Password	The password Ignition will use to authenticate against the SMTP server. This is only required if the SMTP server expects authentication.

Two Way Settings

These settings allow users to acknowledge alarms directly from the Email Notification. When enabled, a link is included in the email that will redirect to a page on the Gateway.

Property Name	Description
Two-way Enabled	Enables remote alarm acknowledgement. If disabled, users will not be able to acknowledge alarms from the email.
Gateway	Network address and port the Gateway is reachable at. Will be used in notification emails.
Send HTTPS Link	Specifies if the link should use an HTTPS link for SSL.

POP3 Two-way Settings

New in 7.9.1 These settings allows Notifications to be acknowledged from when retrieved from a POP3 mailserver. Multiple Ignition Gateways may be configured to use the same POP3 server.

Property Name	Description
Pop3 Two-way Enabled	Enable two-way notification via POP3.
Enable SSL/TLS	Enables SSL/TLS between the Gateway and the mailserver.
Delete On Acknowledge	If enabled, will delete messages from the inbox after acknowledging an alarm.
Hostname	The hostname of the mailserver.
Port	The port the mailserver is running on.
Username	The username the Gateway will use when authenticating against the mailserver.
Password	The password the Gateway will use when authenticating against the mailserver.
Custom Message	Allows an opportunity to provide an additional message to the notification. This custom message is in addition to the Custom Messages configured on the Alarm and the Notification Blocks in the Alarm Pipeline.
Polling Interval	The interval (in milliseconds) that the Gateway will check for new notifications.

Auditing

Property Name	Description
Audit Profile	If an audit profile is selected, events such as emails and acknowledgements will be stored to the audit system. Note that alarm acknowledgements are also stored to the alarm journal.

Advanced

Property Name	Description
SMTP Timeout	Timeout (in milliseconds) to use when connecting to, reading from, and writing to the SMTP server.
Debug Mode Enabled	Enable email session debugging. Information is printed to standard output (wrapper.log).
STARTTLS Enabled	Enable use of the STARTTLS command, allowing the connection to be upgraded to an SSL or TLS connection if supported by the server. This is not necessary for connections that are already SSL/TLS. Ignored when Use SMTP Profile is checked.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Simple One-way Email Notification Profile

New in 7.9.2

The Simple One-way Email Notification Profile is another type of Email Notification Profile that can be setup to notify people when alarms occur. This One-way Email Notification Profile is just as the name suggests, one way! It listens for active alarms within a given priority range, and when the alarm criteria is met, it sends alarm notifications via email to a specified **On-Call Roster**.

Although it's similar to the Email Notification Profile, it does not utilize **Alarm Pipelines** and does not include a link in the Email message to allow recipients to acknowledge the alarm.

NOTE

The Simple One-way Email Notification requires the Alarm Notification Module to be installed.

Simple One-way Email Notification Profile Properties

Main

Property Name	Description
Name	Name of the profile.
Description	Brief description of the profile. Optional.
Enabled	Whether the Simple One-way Email Notification is enabled.

Email Settings

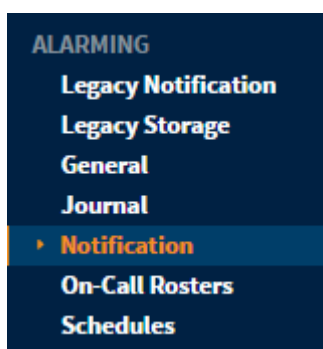
Property Name	Description
SMTP Profile	The SMTP Profile that will be used to distribute the notifications.
On-Call Roster	The Roster of users that will be notified when an alarm meets the criteria set in this profile.

Alarm Criteria

Property Name	Description
Minimum Priority	For a Notification to be sent from this Notification Profile, its priority must be equal to or higher than the priority specified on this property. (Diagnostic, Low, Medium, High, Critical)
Maximum Priority	For a Notification to be sent from this Notification Profile, its priority must be equal to or lower than the priority specified on this property. (Diagnostic, Low, Medium, High, Critical)

Creating a Simple One-way Email Notification Profile

1. Go to the **Configure** section of the Gateway webpage.
2. Under **Alarming**, select **Notification** from the menu on the left side.



3. A list of Alarm Notification Profiles will appear. Click on the **Create new Alarm notification Profile** link at the bottom of the page.

Alarm Notification Profiles

Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	delete edit
Email 1		true	Email Notification	Running	delete edit
SMS		true	SMS Notification	Running	delete edit

[→ Create new Alarm Notification Profile...](#)

[→ Test Pipelines and Notification Profiles...](#)

4. A list of all the alarm notification profile types will appear. Choose **Simple One-way Email Notification**, and click **Next**.

NOTE

If any of the profiles are not shown, it means the required module is missing or faulted. To verify the Alarm Notification Module is installed and running, click on **Modules** from the menu bar on the left.

Add Alarm Notification Profile Step 1: Choose Type

☐ **Email Notification**

Send alarm notifications via email.

☐ **Remote Gateway Notification**

Enables alarm notification using notification profiles of a remote gateway. Will also expose the pipelines of the remote gateway directly to tags.

☒ **Simple One-way Email Notification**

Send alarm notifications via email to users in a user source. Listens for active alarms within a given priority range.

☐ **SMS Notification**

Send alarm notifications via SMS using your Airlink device.

[Next >](#)

5. Configure the profile settings for the Simple One-way email Notification Profile, and then click **Create New Alarm Notification Profile**.

- **Name:** OneWay Email
- **SMTP:** SMTP (Please see the [Email Settings](#) page for more information on creating an SMTP profile.)
- **On-call Roster:** Manager (For more information, please see the On-Call Rosters page.)

- **Minimum Priority:** High
- **Maximum Priority:** Critical

This notification profile example below is configured to send email notifications to the users in the **Managers** On-Call Roster when the alarm criteria is **High** or **Critical**.

New Alarm Notification Profile

Main	
Name	OneWay Email
Description	
Enabled	☒ (default: true)

Email Settings	
SMTP Profile	SMTP
On-Call Roster	Managers

Alarm Criteria	
Minimum Priority	High (default: Low)
Maximum Priority	Critical (default: Critical)

Create New Alarm Notification Profile

6. Once created, the Simple One-way Email Notification Profile will appear in your Alarm Notification Profiles list, and it will be running. You can edit the profile to change the On-Call Roster and/or alarm criteria at any time. You can create as many Simple One-way Email Notification Profiles as you need.

Alarm Notification Profiles

✓ Successfully updated Alarm Notification Profile "Email 1"

Name	Description	Enabled	Type	Status		
Email		true	Email Notification	Running	delete	edit
Email 1		true	Email Notification	Running	delete	edit
OneWay Email		true	Simple One-way Email Notification	Running	delete	edit
SMS		true	SMS Notification	Running	delete	edit

Using the Simple One-way Email Notification Profile

Once you have the notification profile setup, that's it! There is no other setup required, as this notification profile type can't be used in an alarm pipeline. Instead, any alarms that fall within the set priority range will automatically be sent out to the roster listed in the notification profile. It is important to be aware that there is no way to filter what alarms get sent using this notification profile other than the priority, so if alarms are sent to the same roster through an alarm pipeline, users may see duplicate alarms.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

SMS Notification Profile



INDUCTIVE
UNIVERSITY

SMS Notification Profile

[Watch the video](#)

SMS Notification Module

The **SMS Notification Module** allows you to deliver SMS alarm notifications via a cellular modem configured with a SIM card belonging to an active cellular account. If enabled, recipients of these messages can reply with a special code in order to acknowledge the alarm.

NOTE

The SMS module is dependent on the Alarm Notification Module. You must have both modules installed to send alarms.

Device Configuration

Inductive Automation officially supports three Airlink Devices: the RV50, the Raven XE, and the LS300. Most devices from most manufacturers can be used, but some configuration settings might be different. You can do the basic configuration for the Airlink modem by importing a template settings file provided by Inductive Automation.



**Knowledge
Base**

KBA: Airlink LS300

AirLink LS300 Setup Guide

[Link to Knowledge Base Article](#)

Click for more information about [The AirLink® LS300](#).



Knowledge
Base

KBA: Airlink RV50

AirLink RV50 Setup Guide

[Link to Knowledge Base Article](#)

Click for more information about [The AirLink® RV50](#).

Multiple Systems with One Modem

You can use one SMS modem with multiple Ignition systems for one-way (outgoing) messaging. If you want to use two-way messaging (incoming and outgoing), then only one Ignition system can receive those incoming responses.

Create an SMS Notification Profile

1. Go to the **Configure** section of the Gateway.
2. Choose **Alarming > Notification** from the menu on the left to open the **Alarm Notification Profiles** page.
3. Find the blue arrow and click on **Create new Alarm Notification Profile**.
4. Select the **SMS Notification** profile type and click **Next**.

Add Alarm Notification Profile Step 1: Choose Type

☐ **Email Notification**
Send alarm notifications via email.

☐ **Remote Gateway Notification**
Enables alarm notification using notification profiles of a remote gateway. Will also expose the pipelines of the remote gateway directly to tags.

☒ **SMS Notification**
Send alarm notifications via SMS using your Airlink Raven XE.

☐ **VOIP Voice Notification**
Telephone notification using VOIP, compatible with most SIP based telephony systems.

Next >

5. Set the following settings concerning your device, and then click on **Create New Alarm Notification Profile**.

- **Name:** SMS
- **Description:** Optional
- **Enabled Box:** Click to Enable SMS notifications.
- **Airlink Host Address:** The IP address of the Airlink modem.
- **Send Port:** The default port is 17341, which is configured in the device.
- **Receive Port:** The default port is 17342, which is the port used by Ignition when two-way messaging is enabled. The port must not already be used by the host system, and must not be blocked by a firewall.
- **Two-way Enabled:** If enabled, the message recipients will receive a text message notifying them of an alarm, and a link to acknowledge the alarm. This is communicated to Ignition via UDP data sent from the modem. Therefore, the Airlink modem must be configured with the IP address of the system.
- **Numeric Only Ack Code:** If this is true, the ack code sent with the SMS message will be a numeric code instead of an alphanumeric code.
- **Audit Profile:** If an audit profile is selected, events such as SMS messages and acknowledgements will be stored to the audit system.

Once all the settings are entered, click the **Create New Alarm Notification Profile** button.

New Alarm Notification Profile

Main	
Name	SMS
Description	
Enabled	☒ (default: true)

SMS Settings	
Airlink Host Address	XX.XX.X.XXX
Send Port	17341 (default: 17,341)
Receive Port	17342 (default: 17,342)
Two-way Enabled	☒ (default: false)
Numeric Only Ack Code	☐ (default: false)

Auditing	
Audit Profile	- none - If an audit profile is selected, events such as SMS messages and acknowledgements will be stored to the audit system. Note that alarm acknowledgements are also stored to the alarm journal.

Create New Alarm Notification Profile

6. Once you have the SMS Notification profile set up, you can add phone numbers to your contacts using the **SMS** Contact type.

Twilio Module

SMS notifications can also be sent with the **Twilio** module. This does not require a cellular modem, but the Gateway must have Internet access, and a Twilio account with SMS capabilities must be created (www.twilio.com). Once created, the account must be defined in Ignition.

NOTE

The Twilio module is dependent on the Alarm Notification module. You must have both modules installed to send alarms.

Defining a Twilio Account

1. Go to the **Configure** section of the Gateway.
2. Choose **Twilio > Accounts** from the menu on the left.
3. Click the **Create new Twilio Account** link.
4. Provide a **Name** for the account as well as the **Twilio Account Sid**, and **Twilio Auth Token**.
 - Both the Account SID and Auth Token can be retrieved from the Account Settings menu in Twilio.
5. If desired, provide the **Public Hostname**. This is the hostname or IP address that inbound requests will be forwarded. If the users should be able to acknowledge alarms by responding to the SMS messages, then this property should list a publicly reachable hostname or IP address.
6. Click the **Create New Twilio Account** button.

Once the account has been defined, a Twilio SMS notification profile must also be created.

Creating a Twilio SMS Notification Profile

1. Go to the **Configure** section of the Gateway.
2. Choose **Alarming > Notification** from the menu on the left to open the **Alarm Notification Profiles** page.
3. Find the blue arrow and click on **Create new Alarm Notification Profile**.
4. Select the **Twilio SMS Notification** profile type, and click **Next**.
5. Provide a name for the profile, and select a **Twilio Account** from the dropdown.
 - **Acknowledgement Allowed** enables users to respond to notifications directly from the SMS message; also known as **Two-Way notification**. If enabled, the **Public Hostname** property in the Twilio Account must be configured (see step 5 in "To define a Twilio account" above).
6. Click **Create New Alarm Notification Profile**.



TIP

Once your Notification profile is created, you can use it in an [Alarm Pipeline](#).

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Voice Notification Profile



INDUCTIVE
UNIVERSITY

Voice Notification Profile

[Watch the video](#)

NOTE

For the **Voice Notification Module** to work, a compatible Voice Module must also be installed.

The Voice Notification Module adds the ability to deliver alarm notifications to users via telephone, using most SIP compatible phone system, excluding those requiring Cnonce. Messages are constructed in text and are delivered through a high quality text-to-speech engine. The engine supports multiple voices and languages.

Core Features

Deliver voice calls through any SIP compatible phone system. No dedicated hardware required. Messages generated by high-quality text to speech, and not a canned set of prerecorded files.

- Supports multiple languages concurrently, based on user preference.
- Allows users to acknowledge events.
- Supports requiring a personal identification number for additional security.
- Ties into the audit log to audit call events, successful message delivery, and user acknowledgements.
- Supports message consolidation.

Voice Notification Configuration

About SIP and VOIP

The Session Initiation Protocol, is a highly popular specification for implementing Voice Over IP (VOIP) based phone systems. The protocol itself, as the name suggests, is responsible for initiating communication sessions, and then other protocols such as SDP and RTP are used to actually transfer voice data. Ignition has these protocols built in, SIP is a peer-to-peer protocol, where one side talks directly to the other. However, it is possible to have Gateways that repeat and route data between the two parties. Sometimes phone calls on VOIP networks stay purely in software, but often a Gateway will transition the call to a traditional phone line.

By leveraging SIP, Ignition can call physical phones, soft phones, be worked into more complex PBX schemes, while avoiding the high cost of traditional, dedicated voice cards. To get started, though, you'll need some sort of SIP Gateway. Asterisk is a popular, open source, system that is used by thousands of companies worldwide. If you simply want to connect to a phone line, there are several low cost devices that runs Asterisk.

Gateway Configuration

To get started with voice notification, add a new profile by going to **Configure > Alarming > Notification** in the Ignition Gateway. You are only required to specify the host address of the SIP gateway, though depending on the Gateway, you may be required to enter a username and password.

There are additional settings that dictate how calls are managed, such as timeouts for answering, and the maximum amount of time that a call can take. Additionally, you can choose to link the notification profile to an audit profile in order to record important call lifecycle events.

After saving the profile, you should see the status update from “Unknown” to “Registered”, indicating that the Gateway has successfully registered with the SIP Gateway. If you see an error, verify that the settings are correct, and that the username and password are correct. The system log console can also be useful in determining what is wrong.

NOTE

If you receive errors indicating that an “invalid parameter” has been used, try setting the local and public bind interface settings under the advanced options. These should be set to the IP address of the network card that is being used to communicate with the SIP Gateway. On some systems, especially Linux hosts, the default empty values result in this error.

Using Skype

If you don't have an existing VOIP system in place, using a hosted service is the quickest way to get going with the voice notification module. Skype offers SIP based service through their Skype Connect product. To get started, you must have a business account, which provides you access to the Skype Manager. From there, you can create a new Skype Connect channel. For more information, visit <http://manager.skype.com>

Once you have created a Skype Connect channel, configuration in the Gateway is similar to that of any other PBX system. The host will be "sip.skype.com", and the user name and password will be those provided by skype during the registration process. Note: while skype allows to you specify how many "channels" may be used, Ignition currently only supports one channel at a time.

Configuring Messages

The message played to the user during the phone calls is defined in the call script. The script dictates the overall structure of the call, defining the phrases and options, and the possible responses. The messages for each alarm are built off of a message template that can reference properties in the alarm. The script can be edited by selecting "manage scripts" next to the voice notification profile. Note: Although the link appears next to a particular profile, the scripts are shared across all voice notification profiles.

The role of each phrase in the script is explained on the settings page. Some parts of the script, such as the phrase requesting the user's PIN number, will only be used if certain settings are configured on the notification block in the pipeline (in this case, the setting to require a PIN). As previously mentioned, the alarm message (for both active and clear) can reference any property of the alarm. The default message looks like this:

```
At {eventTime|hh:mm:ss}, alarm named {name} became {eventState} with a value of {eventValue}
```

In this case, the message refers to the alarm name, the eventState and eventValue (note: eventState is different than state. Event state is just the transition that triggered the alarm, such as "active", whereas "state" is the full current state, such as "active, unacknowledged"), and the eventTime. Notice that the event time is formatted to only use the time, and not include the current date.

Scripts can be created for different languages. When the system attempts to deliver a notification, it will look to see if the target user has a preferred language. If so, and the language has both a script defined, and a compatible voice installed, the user will be notified in that language.

Alarms also allow you to define a custom message, relevant to that particular state. If a custom message is defined, it will be used instead of the default message in the script.

The Call Lifecycle

A call is initiated when an alarm event enters the notification profile block in a pipeline. When this occurs, the target users are collected, based on the defined call roster, and the current schedule. Only users who have phone contact details defined will be selected for phone notification.

The voice system can only call one number at a time, and so it takes the first contact off of the queue and initiates the call. The user is given up to the “answer timeout” to answer. After picking up, the user will be asked to enter their pin number, or press any number to continue. The call is not considered “answered” until this action occurs, so the message will be repeated until the answer timeout expires. By acting in this way, the system is able to confirm that a human has actually answered, and the call will then be audited as successful.

Once past the initial challenge, the user will then hear the alarm messages. After each message, they will be asked to either acknowledge, ignore, or repeat the message. Selecting “acknowledge” will cause the alarm to be acknowledged in the alarming system, likely causing it to drop out of the alarm pipeline (dependant on pipeline settings). Ignoring the alarm indicates that the user has heard the message, but cannot or does not want to acknowledge the alarm.

Once the call has completed, the notification system will check the alarm events against the pipeline fallout conditions, and move on to the next call. The system will cycle through all alarms and all contacts (and all phone numbers for each contact) until everyone has been notified, or the pipeline settings have caused all events to drop out.

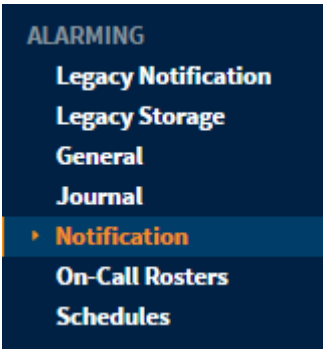
Pipeline Configuration

The voice notification system is accessed through the Notification Block in the Designer's pipeline configuration, like all other notification methods. As with other methods, you select a call roster, and can optionally turn on consolidation. There are only two options which can be set on the notification block:

Option Name	Description
Require PIN	If selected, the user will be required to enter their personal identification number in order to hear the message. The user's PIN number is specified on their profile in the user management system. If false, or if the user does not have a PIN specified, the user will only be required to press any key to continue.
Allow Acknowledgement	If true, the user will be given the opportunity to acknowledge the alarm. If false, they will only be able to hear the alarm and continue.

Creating a Voice Notification Profile

- 1. Go to the **Configure** section of the Gateway.
- 2. Select **Alarming > Notification** from the menu on the left side.



- 3. Click on the **Create new Alarm Notification Profile** link.

Alarm Notification Profiles

Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	deleteedit
Email 1		true	Email Notification	Running	deleteedit
SMS		true	SMS Notification	Running	deleteedit

- Create new Alarm Notification Profile...
- Test Pipelines and Notification Profiles...

- 4. The Add Alarm Notification Profile Step 1 window will appear. Select the **VOIP Voice Notification** radio button a click **Next**.

Add Alarm Notification Profile Step 1: Choose Type

☐ **Email Notification**
Send alarm notifications via email.

☐ **Remote Gateway Notification**
Enables alarm notification using notification profiles of a remote gateway. Will also expose the pipelines of the remote gateway directly to tags.

☐ **SMS Notification**
Send alarm notifications via SMS using your Airlink Raven XE.

☒ **VOIP Voice Notification**
Telephone notification using VOIP, compatible with most SIP based telephony systems.

Next >

5. On the **New Alarm Notification Profile** page, configure your settings. Enter the Gateway Address or domain name of your SIP Gateway. You may be required to enter a username and password. There are also some additional call settings that may need to be setup depending on how you want to manage your calls, and based on your preference. When completed, click **Create the New Alarm Notification Profile**. Your new Voice Notification Profile will be displayed with a status of **Registered with VOIP Host**.

Alarm Notification Profiles

Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	<button>delete</button> <button>edit</button>
Email 1		true	Email Notification	Running	<button>delete</button> <button>edit</button>
SMS		true	SMS Notification	Running	<button>delete</button> <button>edit</button>
Voice		true	VOIP Voice Notification	Registered with VOIP Host	<button>More ▾</button> <button>edit</button>



TIP

Once your Notification profile is created, you can use it in an [Alarm Pipeline](#).

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Remote Gateway Notification



INDUCTIVE
UNIVERSITY

Remote Alarms

[Watch the video](#)

Overview

The Remote Alarm Provider offers several ways to utilize the alarm notification capabilities of a remote Ignition Gateway. Alarms can be sent through a central Gateway, with full support for remote acknowledgement delivery. This means that you can manage your Email, SMS, and Voice equipment in a central location, and provide those services to a multitude of Gateways.

After configuring a remote provider, the Ignition Gateway will automatically expose the alarm pipelines configured on the remote system. Tags can then target those pipelines directly, with no further configuration required. Alternatively, custom pipelines can be created, and can utilize the remote notification profile through the Notification Block, thus expanding the already significant power of alarm pipelines to include the notification capabilities of other Gateways.

The Remote Alarm Provider is provided by the Alarm Notification Module, which must be present on all Gateways involved. However, notification modules, such as SMS and Voice, only need to be present on the Gateways that are ultimately executing the notification.

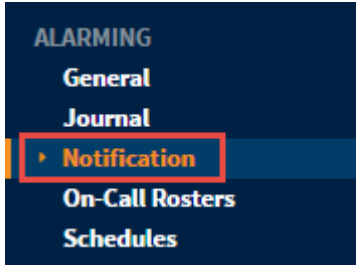
REMOTE GATEWAY MUST BE ON THE SAME GATEWAY NETWORK

In order to use the alarm notification capabilities of a remote Ignition Gateway, your Gateway and the Gateway you want to connect to must be connected to the same Gateway network.

Enabling remote alarm notifications is easy. There are just a few steps to follow to establish your connection to the designated Gateway, expose the pipelines that are part of that Gateway, and to configure your Tag(s), thus enabling the alarm notification. Let's get started.

Connecting to a Remote Gateway and Selecting a Profile

1. Go to the **Configure** section of the Gateway Webpage, and scroll down and select **Alarming > Notification**.



2. The Add Alarm Notification Profile Step 1 window will appear. Select **Remote Gateway Notification**, and click **Next**.

Add Alarm Notification Profile Step 1: Choose Type

☐ **Email Notification**
Send alarm notifications via email.

☒ **Remote Gateway Notification**
Enables alarm notification using notification profiles of a remote gateway. Will also expose the pipelines of the remote gateway directly to tags.

☐ **SMS Notification**
Send alarm notifications via SMS using your Airlink Raven XE.

☐ **VOIP Voice Notification**
Telephone notification using VOIP, compatible with most SIP based telephony systems.

Next >

3. All the known Gateways will be displayed. Choose from the list of available Gateways or you can setup a Gateway manually, then click **Next**.

Select Gateway

💡 Known gateways appear here. If you do not see a gateway that you expect, please check the Gateway Network connection settings to verify that connections are valid.

☒ **Ignition-ubuntu**

Connection: 10.20.5.232:8088, State: Connected

☐ **Specify Gateway Name Manually**

If the gateway is not currently available or displayed here, you can specify its name manually.

Next >

4. Once your Gateway is connected, Ignition exposes all the pipelines on that Gateway. Choose from the list of available Remote Notification Profiles, and click **Next**.

Remote Notification Profile

☒ **Email**

☐ **Other Profile**

Manually specify remote alarm profile.

Next >

5. The new Remote Notification Profile window will appear. It will automatically show the name of your Gateway followed by the name of the remote Profile unless you are manually specifying a remote profile. There is a naming convention that is used when identifying the 'Name' of the New Remote Notification Profile. It is the Gateway Name followed by an underscore and the name of the Profile (i.e., Ignition-ubuntu_Email). Click Create New **Alarm Notification Profile**.

New Remote Notification Profile

Main	
Name	Ignition-ubuntu_EMail
Description	
Enabled	☒ (default: true)

Remote Gateway	
Gateway	Ignition-ubuntu
Notification Profile	EMail

Create New Alarm Notification Profile

6. The Alarm Notification Profiles window will open showing all your alarm profiles, including the one you just created.

Controller

 **Ignition**
by Inductive Automation

USER MANUAL SUPPORT
admin Sign Out
Launch Designer

HOME STATUS **CONFIGURE**

Search...

SYSTEM
Overview
Backup/Restore
Licensing
Modules
Projects
Redundancy
Gateway Settings

NETWORKING
Gateway Network
Email Settings

SECURITY
Auditing
Users, Roles
Service Security
Security Zones

DATABASES
Connections

License Incomplete ? 0:00:00
Reset Trial View Modules

Alarm Notification Profiles

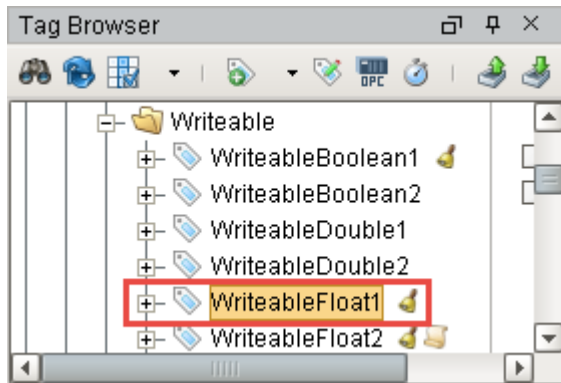
Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	delete edit
Email 1		true	Email Notification	Running	delete edit
Ignition-ubuntu_EMail		true	Remote Gateway Notification	Running	delete edit
SMS		true	SMS Notification	Running	delete edit
Voice		true	VOIP Voice Notification	Demo Expired	More edit

[→ Create new Alarm Notification Profile...](#)
[→ Test Pipelines and Notification Profiles...](#)

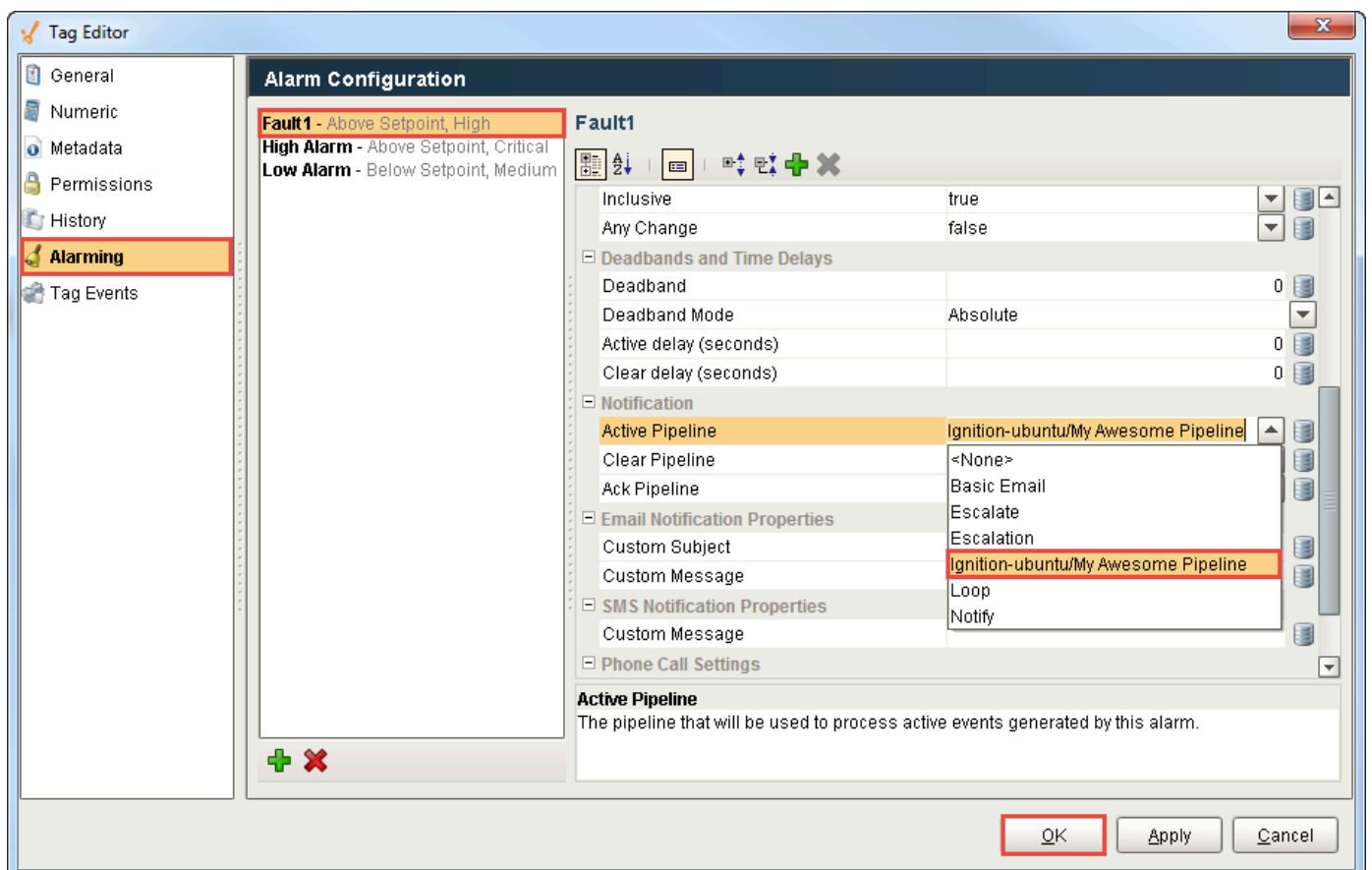
Enabling the Remote Alarm Notification

The last steps are to select your Tag, add an alarm, and configure your alarm settings. If you already have an existing alarm, then all you need to do is go to your alarm configuration, and select the remote pipeline that you want to use for your alarm.

1. In the **Designer**, under the **Tag Browser**, double click on your **Tag** (i.e., WriteableFloat1) to open the Tag Editor.



2. In the **Tag Editor**, click **Alarming**, and select your **alarm** (i.e., Fault1). (If you don't have an alarm, then you'll need to add an alarm. Refer to the section on [Configuring Alarms](#)). Scroll down to the **Notification** area and under **Active Pipeline**, select the remote pipeline (i.e., Ignition-ubuntu/MyAwesome Pipeline) that you want to use from the dropdown list. Click **OK**.



Now, you can even create new custom pipelines and use the remote notification profile through the Notification Block, while expanding the significant power of alarm pipelines to include the notification capabilities of other Gateways sharing the same Gateway network.

ⓘ NOTIFICATION MODULES NEED TO BE INSTALLED ON THE GATEWAY RUNNING THE NOTIFICATION

The appropriate Notification Modules such as SMS and/or Voice must be installed on the Gateways that are executing the alarm notification.

💡 TIP

Once your Notification profile is created, you can use it in an [Alarm Pipeline](#).

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

User, Schedule, and Roster Management



INDUCTIVE
UNIVERSITY

User, Schedule, and Roster
Management

[Watch the video](#)

Who to Notify

To make sure all the right people get notified quickly about an impending alarm, the **On-Call Roster** lets you organize individuals into groups based on their roles and schedules. When an alarm is triggered, it goes to a designated On-Call Roster where it evaluates users **schedules** so only the appropriate individuals will be notified. You can easily change schedules, assign roles, and specify who should be notified when alarms are triggered.

Another slick feature is the ability to edit the **On-Call Roster** and **Schedule** in the client in realtime, as well as in the Designer. Using roles, you can even permit individuals to edit their own contact and schedule information on-the-fly to account for their own unexpected time away from work so alarms won't be ignored, and will be sent promptly to the next available person.

Roster Management - from the Client Window

Command

Windows

Help

Ignition

by inductive automation

Overview

User Management

Schedule Management

Roster Management

Current Time:

06/30/2016 11:21 AM

12 Alarms

Logged In: admin

Lock Screen

Switch User

Logout

Roster Management

On-Call Rosters > Edit Roster

< back

Save

Add users from

default

Available Users

Maria

guest

Jane Doe

John Doe

Sal Supervisor

Sally

On-Call Roster

Jane Doe

John Doe

Sal Supervisor

Sally

Maria

Schedule Management - from the Configure > Alarming > Schedules section of the Gateway

SYSTEM

Overview

Backup/Restore

Licensing

Modules

Projects

Redundancy

Gateway Settings

NETWORKING

Gateway Network

Email Settings

SECURITY

Auditing

Users, Roles

Service Security

Security Zones

DATABASES

Connections

Drivers

Store and Forward

ALARMING

Legacy Notification

Legacy Storage

General

Journal

Notification

On-Call Rosters

Schedules

TAGS

Schedule Management

SchedulesHolidays

Edit Schedule

General

Name

Day Shift

Description

Weekday Day Shift

Observe Holidays

☒ Choose whether or not this schedule observes any configured holidays.

Schedule

All days

☐

0:00-24:00

☒ This schedule will have the same hours 7 days a week.

Weekdays

☒

8:00-12:00,13:00-18:00

☒ This schedule will have the same hours Monday-Friday, but will have no hours on weekends.

User Management - from the Designer

Users

Users

Username	Name	Roles	Contact Info	Schedule
admin		Administrator, s...		Always
george	George Johnson		email: george.jo...	Always
joe	Joe Schmo	Administrator	email: joe.schm...	Always
kate	Kate Smith		email: kate.smit...	Always
mary	Mary Swanson		email: mary.sw...	Always

Roles

Role name	# of Members
Administrator	2
Operator	0
super_admin	1

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Notification Users



INDUCTIVE
UNIVERSITY

Notification Users

[Watch the video](#)

Ignition provides the capability to send out alarm notifications to a list of people in Email, SMS, and Phone (voice dial-out). There are several steps to get alarm notifications sent out to a list people. The first thing to do is create an [Alarm Notification Profile](#) for Email, SMS or Voice, whichever method you intend to use. Then, create the users who are going to receive the alarms. Each user must have a name and contact information in order for Ignition to notify them of an impending alarm.

Let's get started!

Adding a User

1. Go to the **Configure** section of the Gateway.
2. Choose **Security > Users, Roles** from the menu on the left. The **User Sources** page is displayed.
3. Select **manage users** from the appropriate user source.

User Sources

Name	Type	Description	
default	Internal	This is the default and always present internal authentication profile.	manage users edit
opcua-module	Internal	OPC-UA clients will authenticate against this profile by default.	More ▾ edit

→ [Create new User Source...](#)

→ [Verify an Authentication Profile...](#)

4. Look for the blue arrow and select the **Add User** link to add a user. The **Manage Users and Roles for Profile** page is displayed.
5. Complete the fields that correspond to the user's information.

Manage Users and Roles for Profile 'default'

Users

Roles

Add User

User Properties

Username	admin		
Password			
Password	 Re-type password for verification.		
First Name	Harry		
Last Name	James		
Roles	☒ Administrator ☐ Administrators ☐ guests	☐ Operator ☐ Prod_Sups	☐ Test_Group ☐ Users
Schedule	Always ▾		
Language	English ▾		
Notes			

6. In the **Contact Info** section, choose the delivery method from the dropdown, and enter the appropriate Value such as an email address.

Contact Info

Type	Value	
E-Mail ▾	admin@inductiveautomation.com	Save Cancel

7. Click **Add User**. The new user will now appear under the Users tab.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

On-Call Rosters



INDUCTIVE
UNIVERSITY

On-Call Rosters

[Watch the video](#)

On-Call Roster Overview

The On-Call Roster lets you create user groups to be notified when an alarm occurs. Each group includes a list of users in a specific order. Alarm pipeline's notification blocks must choose an on-call roster which identifies the users to notify for that notification block. Depending on the alarm notification profile used, the users can be notified one at a time (sequential), or all at once (parallel). It is important to remember that when an on-call roster is used for alarm notification, only those users on the roster whose **schedules** are active will be notified.

To Create and Manage the On-Call Roster

1. Go to the **Configure** section of the Gateway.
2. Choose **Alarming > On-Call Rosters**. You can create as many rosters as you wish.
3. Click on the **Create new On-Call Roster**. You can create as many rosters as you wish.
4. On the **New On-Call Roster**, enter a **Name**, and click **Create New On-Call Roster**.
5. Now back on the On-Call Rosters page, click on the **More** button, and select **manage** from the dropdown list to the right of the roster name. The **Manage Roster** page is displayed and from here you can add one or all users from a source list, and put them in any order you wish. You can also remove users from the On-Call Roster.

Agent133

USER MANUAL SUPPORT

admin Sign Out

Launch Designer

HOME STATUS CONFIGURE

Search...

License Incomplete 0:00:00 Reset Trial View Modules

SYSTEM

- Overview
- Backup/Restore
- Licensing
- Modules
- Projects
- Redundancy
- Gateway Settings

NETWORKING

- Gateway Network
- Email Settings

SECURITY

- Auditing
- Users, Roles
- Service Security
- Security Zones

DATABASES

- Connections
- Drivers
- Store and Forward

ALARMING

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

On-Call Rosters

On-Call Rosters are ordered lists of users that should be notified when an alarm occurs. These are used when configuring a notification block inside an alarming pipeline.

Successfully updated On-Call Roster "MyRoster"

Name	Description	# Users
MyRoster		0

manage delete More edit

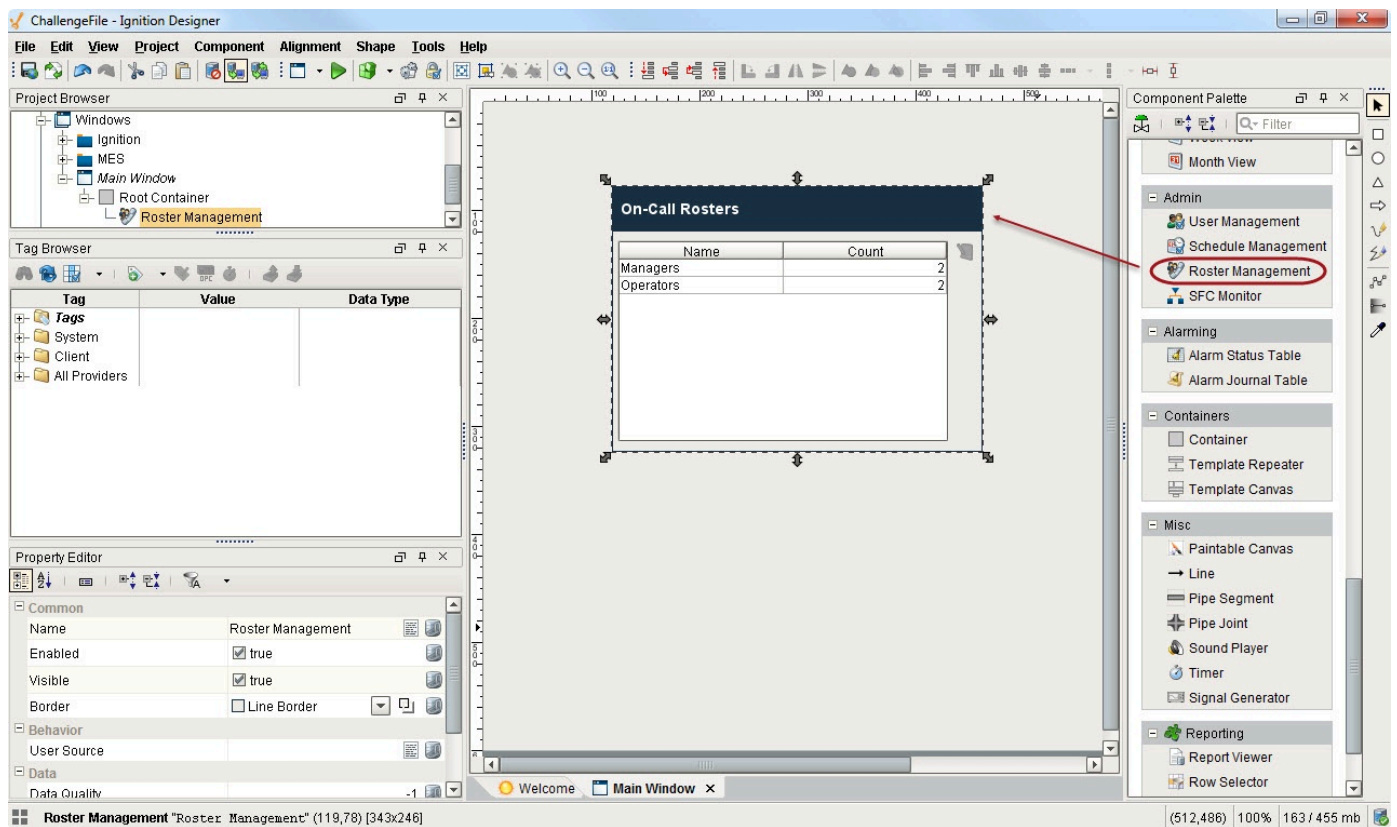
Create new On-Call Roster...

6. Click on **Save**.

To Manage the Roster from the Client

Roster management can take place from a Vision Client.

1. Open your project in the **Designer** and navigate to a window the client has access to. If this is a new window, you will have to add it to your navigation.
2. From the Component Palette, drag and drop a **Roster Management** component in your design window.



3. If you have more than one user source, go to the **Property Editor** and enter the one you want to be used by the roster in the **User Source** property.
4. **Save and Publish** your project. Client users can now manage the roster by re-arranging and adding/removing names to the roster. You can interact with the listed rosters by double-clicking them or selecting them and clicking on the **edit** icon in the upper right.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

User Schedules



INDUCTIVE
UNIVERSITY

User Schedules

[Watch the video](#)



INDUCTIVE
UNIVERSITY

**Schedule and Holiday
Scripting**

[Watch the video](#)

Schedules

Schedules define the times of users on-call availability and unavailability. For example, the **Always** schedule is a schedule that is active 24/7. You can set a schedule for each user in the alarm notification system. The notification messages are then sent only to those users with an active schedule. When a message reaches a notification block in a pipeline, that block's on-call roster is used to find the users with active schedules so they can be notified.

There are also a number of system functions that allow you to create, read, edit, and delete schedules or holidays from a user source using scripting. (i.e., [system.user.addSchedule](#), [system.user.getHoliday](#), etc.). To learn what system functions are available for user scheduling and holiday scripting, refer to the Scripting Functions in the Appendix.

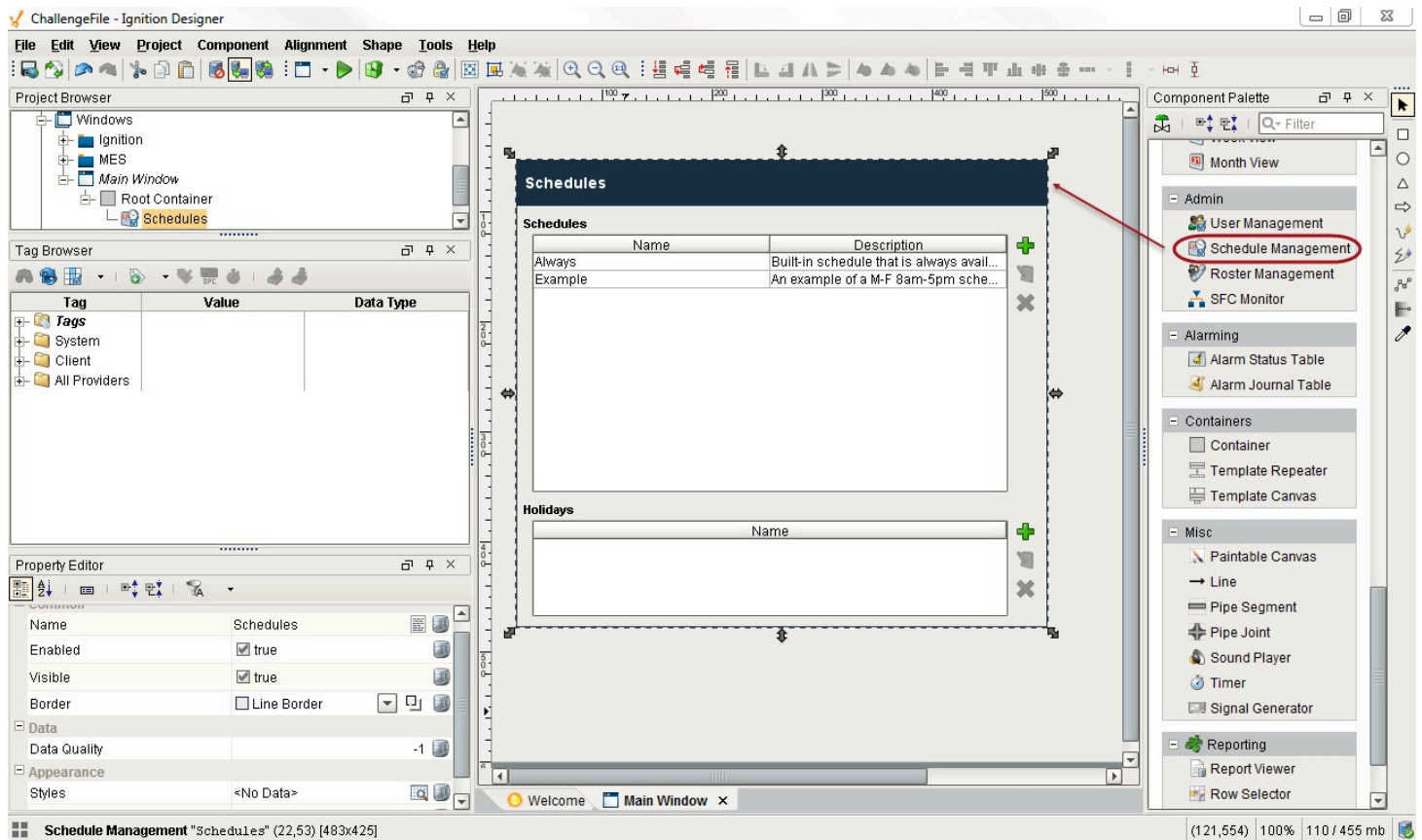
To Define a New Schedule

1. Go to the **Configure** section of the Gateway.
2. Choose **Alarming** > **Schedules** from the menu on the left.

- The Schedule Management page is displayed. Here, you can see any Always and an Example schedule.
 - The **Always** Schedule is a built-in schedule that is always available (24x7x365).
 - The **Example** Schedule is an example of a Monday-Friday 8am-5pm schedule with a lunch break. Click on the **edit** to see the detailed settings.
- 3. Click on **Create new Schedule**.
- 4. Choose a **Standard** or **Composite** schedule and click **Next**.
 - The **Standard** schedule is defined by hours of operation on each day, with repeating options.
 - The **Composite** schedule is defined by the combining of two other schedules.
- 5. On the **New Schedule** page, enter a **Name**, the **Schedule** to be notified, and any **Repeating** options, then click **Add Schedule**.
 - Each schedule is defined by which days of the week it is active for, and during what times.
 - You can define repeating patterns that repeat either weekly or daily, and even plug in holidays.
 - When defining time ranges, use a 24-hour format and commas to break up different spans. For example, if you want the schedule to be active from 8am-5pm with a 45 minute break starting at noon, you'd use: 8:00-12:00, 12:45-17:00.

Managing User Schedules from the Client

There are a few ways to manage user schedules from the Client. The first is to use the Schedule Management component on a window. This component allows you to quickly and easily manage the schedules from the Client.



For more granular control, you may instead want to use scripting to manage the schedules. This may offer a more granular control at the click of a button.

Schedule Management Code Snippet

```
# This code creates a new schedule by using an old schedule but setting observe
holidays to true.
mySchedule = system.user.getSchedule("WeeklySchedule")
if mySchedule != None and mySchedule.getType() == "basic schedule":
    mySchedule.setObserveHolidays(True)
    mySchedule.setName("NewWeeklySchedule")
    system.user.addSchedule(mySchedule)
```

Using Schedules for Alarm Notification

The alarm notification system always uses the Schedules. When an alarm notification pipeline reaches a notification block, it looks at all of the users defined in that block's configured on-call roster. Only those users whose schedules are currently active will be notified. This way, you can group people in call rosters by role, not by shift. For example, suppose you have alarms that should be sent to all supervisors. You can put all of the supervisors in one call roster, and the scheduling system will automatically only notify those supervisors who are on-shift when the alarm goes active.

Using Schedules for Restricting Login

You can use Schedules to restrict users' ability to log in. To enable this, select the **Schedule Restricted** option on the user source in question. That user source will then reject logins for users whose schedule is inactive, even if their credentials were accurate.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Notification Pipelines



INDUCTIVE
UNIVERSITY

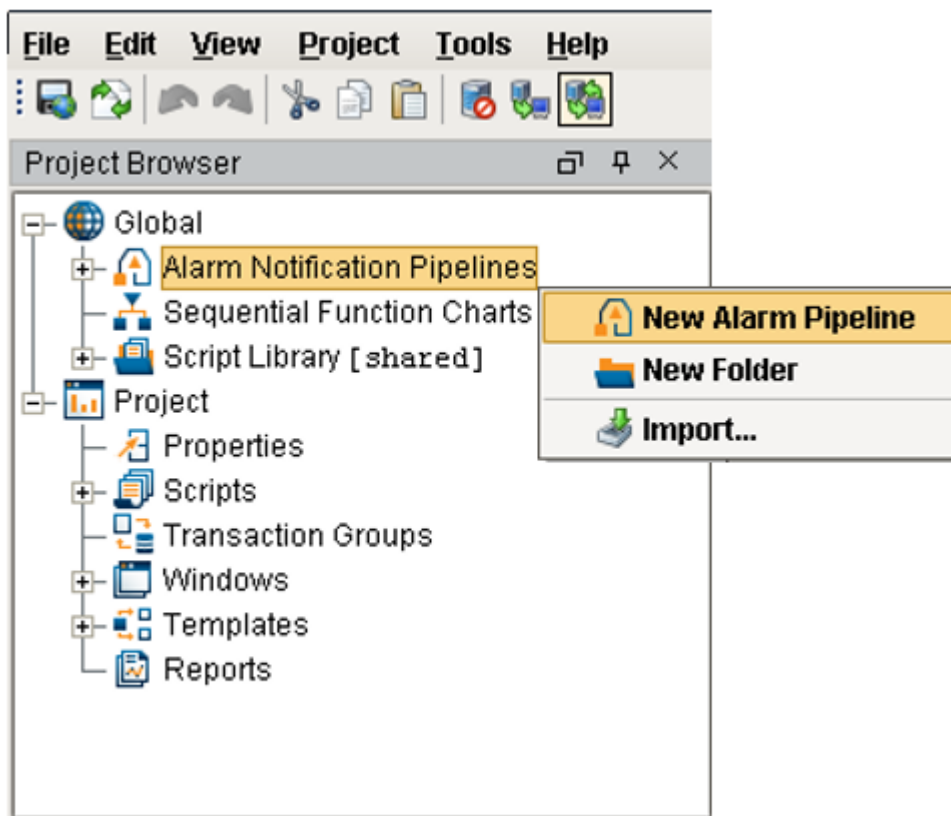
Alarm Pipeline Overview

[Watch the video](#)

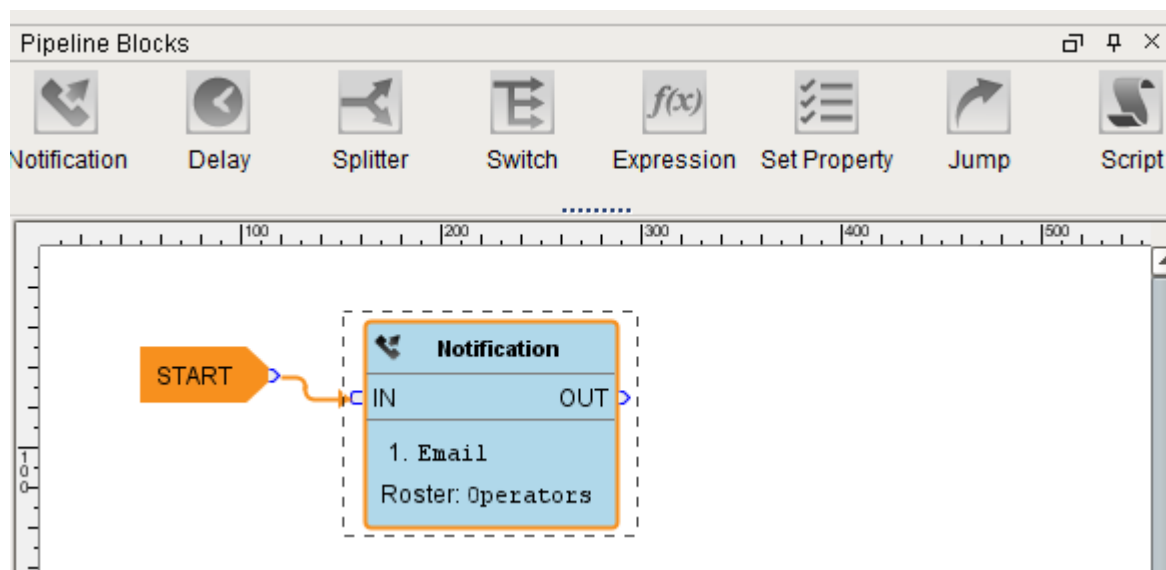
Alarm Notification Pipelines are logic diagrams that control how alarm notifications are sent out, including who they are sent to and when. They can be used to achieve many advanced alarming scenarios, such as delays, escalation, and parallel delivery. Pipelines provide an easy to use a drag and drop mechanism for creating complex notification scenarios.

Creating Pipelines

Before you create a pipeline in the Designer, you need to configure at least one **alarm notification profile** and at least one **roster** in the Gateway. Then in the Designer, you create the Pipelines under the **Global** node of the **Project Browser**. Alarm Pipelines, unlike other types of resources such as windows and Transaction Groups, are global resources, and are not part of a project. A defined pipeline runs in the Gateway, and you can see the same pipelines available to edit regardless of the project that is currently open. If you already have a Remote Gateway configured to support alarm notifications, you can expand the already significant power of alarm pipelines by also exposing alarm pipelines that are configured on the **Remote Ignition Gateway** system.



When you create a new pipeline, or select one to edit, the Designer Switches to the Pipeline Workspace. The workspace is basically a blank canvas, on which you arrange and connect together various **Pipeline Blocks**. Each pipeline block has an input, and potentially has outputs. Using the mouse, you can draw connectors between the output of one block into the input of another. The START block, which you will see when you first create the pipeline, cannot be deleted and is the only fixed part of the workspace. It represents the entry point for alarm events into the pipeline, and is treated like an output block.



Multiple outputs can be connected to a single input. Also, it is perfectly acceptable to write the output of a downstream block back to the input of an earlier block, creating a loop. Of course, some care should be taken when doing this, as it can easily result in unexpected behavior, like sending out emails indefinitely for one event!

Understanding Pipelines and Events

Understanding how events flow through pipelines is crucial to leveraging them fully. Alarms are configured on individual Tags, and each alarm can specify a target pipeline for active and clear conditions. When the condition for that alarm is met a new alarm event is generated and sent to the entry point of the associated pipeline. The alarm event then moves through the different logical elements of the pipeline until it finally reaches a pipeline endpoint, or until the alarm matches the pipeline's dropout condition, upon which it then exits the pipeline. There can be multiple alarm events in a pipeline at any given point in time and you can view the current status of the pipeline from the Alarm Pipelines status screen in the Ignition Gateway. It is possible to forward events from one pipeline to another, making it possible to create multipurpose and reusable pipelines.

Each alarm event progresses sequentially through the blocks of a pipeline. However, multiple alarm events can exist in parallel in the pipeline, and new events may enter the pipeline at any time. Some block settings may result in alarm events being held up, such as consolidated notification, but one alarm event cannot affect other events in the pipeline. In this particular case, when multiple events will be collected together for notification, the events that don't drop out of the pipeline after notification are individually forwarded to the output of the block.

At every transition, and occasionally inside of the block execution as well, the dropout conditions will be evaluated. If the event no longer meets the conditions, it will drop out of the pipeline. It is crucial to understand how this works in order to understand pipeline execution. For example, a common scenario is to place a Delay block directly in front of a Notification block. If the delay is 5 minutes, and "Clear" is one of the pipeline drop out conditions, it effectively means that only events that stay active for longer than 5 minutes will actually result in notifications. If acknowledge is one of the dropout conditions, it would mean that operators (or anyone viewing alarm status) would have 5 minutes to acknowledge them, and only then would notifications be sent.

Event Instances and Runtime Properties

As mentioned, a new "alarm event" is generated each time an alarm transitions to active. This event consists of core data, but also has arbitrary properties available on it. The alarm configuration properties are available, as well as associated data, but in addition there are "runtime properties", that only exist while the alarm event is in memory, and are only used by pipelines. These properties can be accessed by several block types, and runtime properties can be created and updated with the "Set Property" block type.

During an alarm event's lifetime, its properties may be updated at any time. When an alarm goes to clear, for example, the system sets information about that state on the alarm event. In the pipeline, if a block accessed the "state" property, it would see a different value than if it had checked just moments earlier.

It may be possible for a specific alarm event to exist multiple times in a pipeline. This is especially true when using the Splitter block, which takes events and forwards them concurrently to multiple outputs. When this occurs, the alarm event is "branched" and a new "event instance" is created. Each event instance will start with identical properties, but future modifications of properties will only affect the particular instance. All instances of an event will properly reflect properties set later by the system, such as acknowledgement or clear state.

Pipeline Lifecycle

Given the potentially long running nature of pipelines, it's important to understand how they operate when you edit them. Each pipeline normally has only one instance running at a time, which handles all of the events that go into it. When you edit a pipeline and save, however, a new instance is instantiated. The old instance is "retired", but continues to run until all of the existing events are done. The new instance only receives events generated after the time that it was created, it does not take over the previous instance's events.

Pipeline Blocks

Pipeline Blocks are the building blocks of the Alarm Pipelines. Each block, depending on the type, will perform some action for the pipeline, such as sending a notification, setting a property, or evaluating an Expression. Refer to the [Pipeline Blocks](#) section for descriptions of the different pipeline blocks that you can use to build your pipeline.

Pipeline Properties

Pipelines have only a few property settings, but they are important.

To Display and Modify the Properties

Click on the gray area of the Pipeline workspace anywhere there isn't a block.

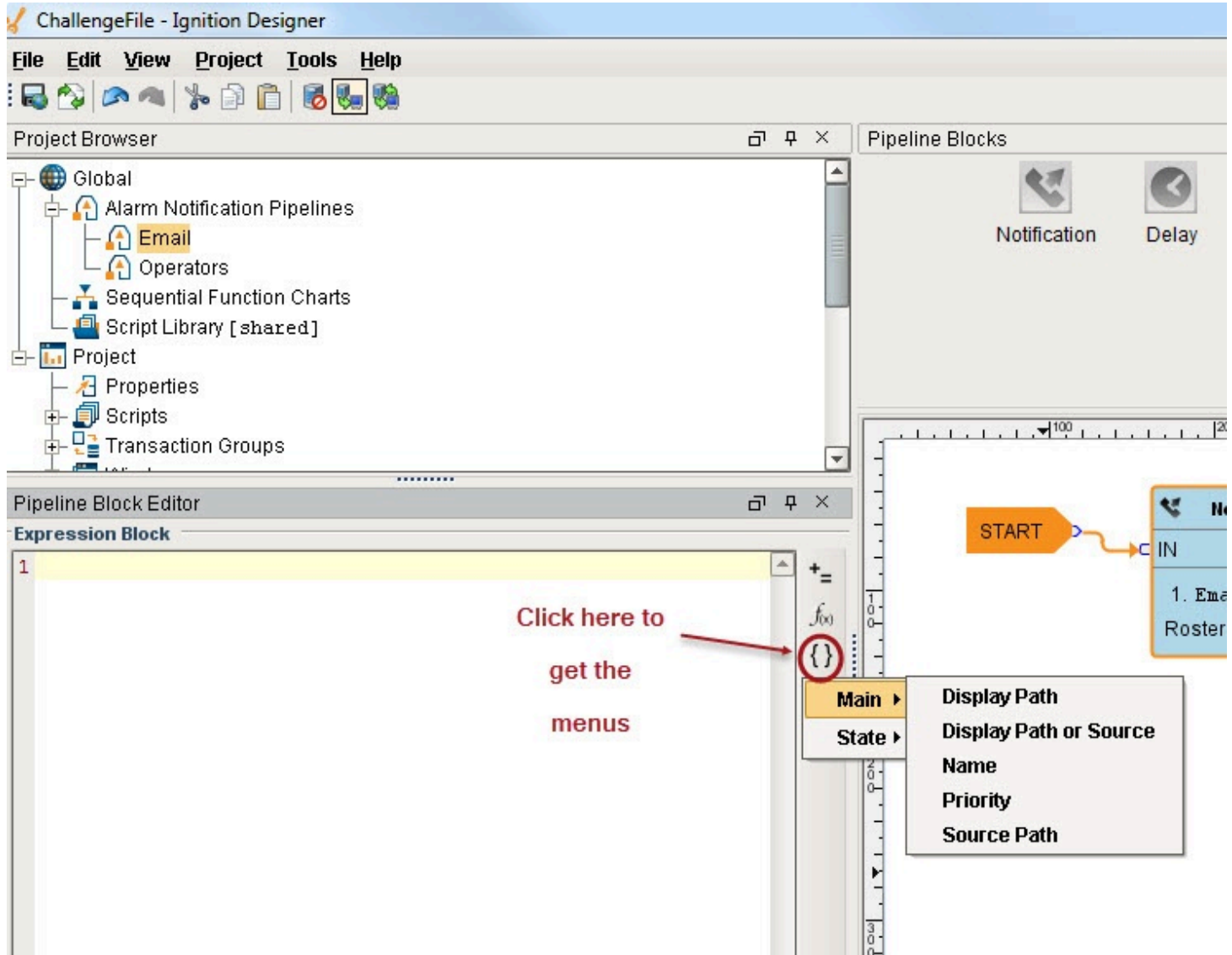
In the lower-left corner of the Designer, the Pipeline Properties are displayed. They are as follows:

- **Dropout Conditions**
 - These dictate when alarms fall out of the pipeline. There are three possible conditions: **Acknowledged**, **Cleared**, and **Shelved**. If any of the selected conditions become true for an event in a pipeline, it will "drop out" entirely, no matter where it is at. No further action will be taken for that event.
- **Pipeline Enabled**

- Indicates whether or not the pipeline is allowed to receive events.

Commonly Used Properties

The following screenshot shows how you can access the two commonly used sets of properties in the Expression Block: the **Main** and **State** properties.



The **Main** properties are information about the alarm that are independent of whether the alarm is currently active, clear, or acknowledged. They are:

- Display Path
- Display Path or Source
- Name
- Priority
- Source Path

The **State** properties are information about the alarm that change while the alarm is changing, and some of them even change each time the alarm goes from active to clear to ack (acknowledged).

- Ack Time
- Ack'ed By
- Ack'ed By (Name)
- Acked?
- Active Time
- Active?
- Clear Time
- Clear?
- Current State
- Event State
- Event Time
- Event Value
- Is Shelved?

Alarm events are made up of many different properties. Refer to the [Alarm Event Properties Reference](#) section for information on alarm properties and how they are used.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Simple Pipeline



INDUCTIVE
UNIVERSITY

Simple Pipeline

[Watch the video](#)

The most basic Alarm Notification Pipeline will simply notify users as soon as the **alarm** becomes active.

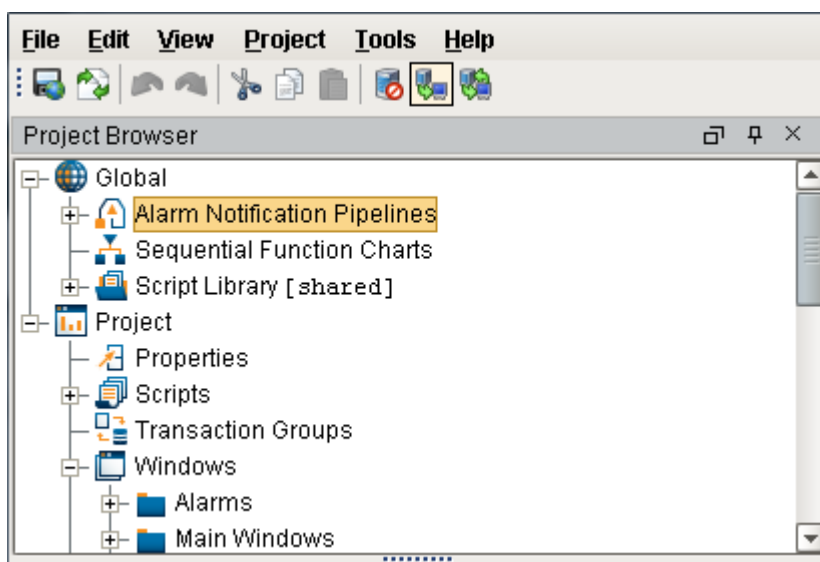
CONFIGURE ALARM NOTIFICATION PROFILE AND ON-CALL ROSTER

Before you create a pipeline in the Designer, you need to configure at least one [Alarm Notification Profile](#) and at least one [On-Call roster](#) in the Gateway.

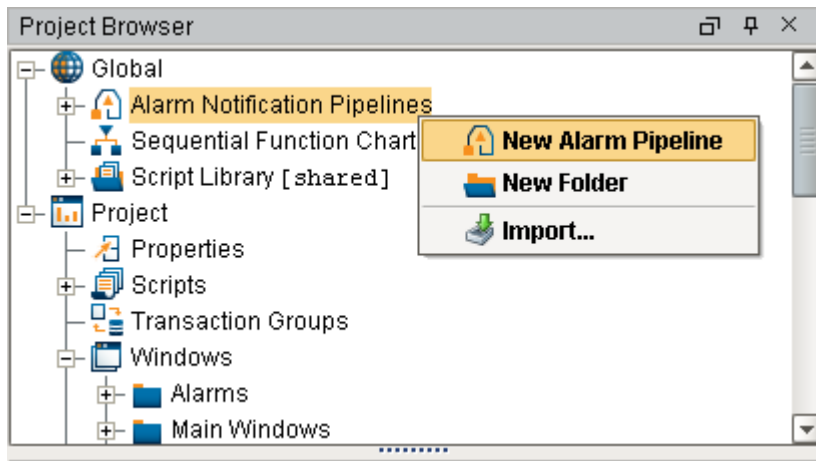
To Create an Alarm Notification Pipeline

The following example will create a simple alarm notification pipeline in Ignition.

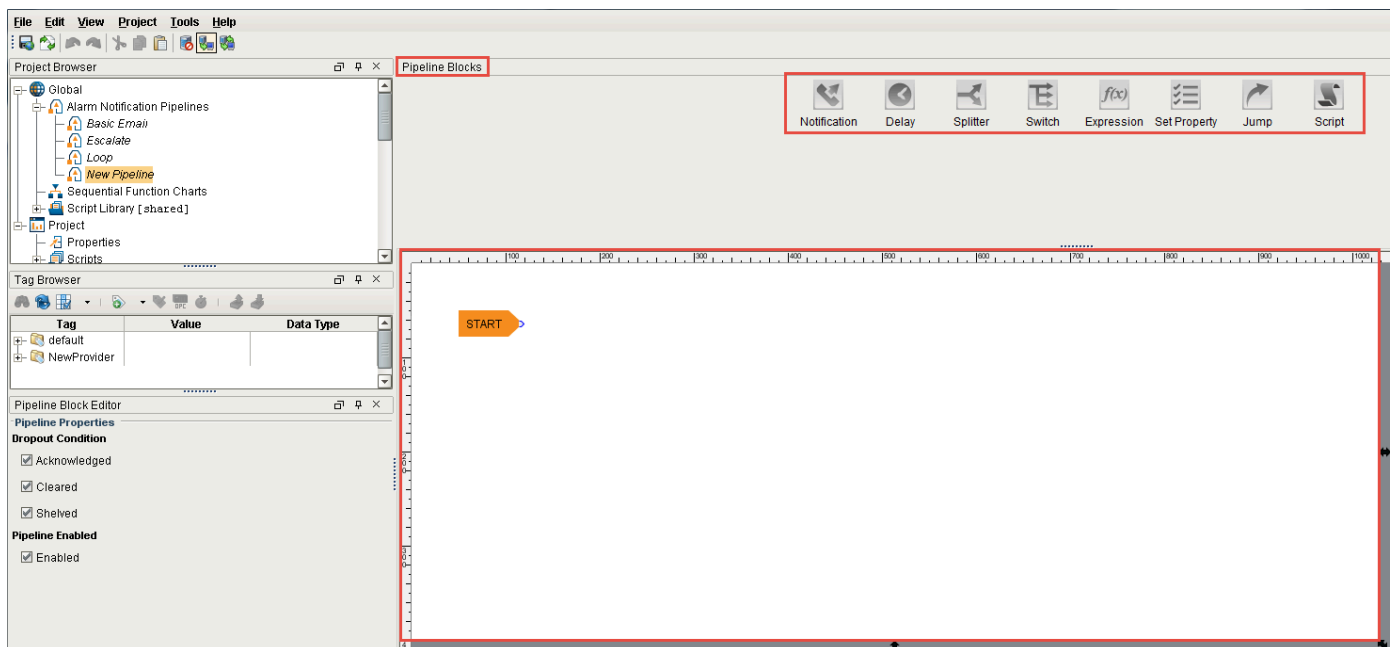
1. In the **Designer**, create a new pipeline in the **Project Browser** by going to the **Global** area.



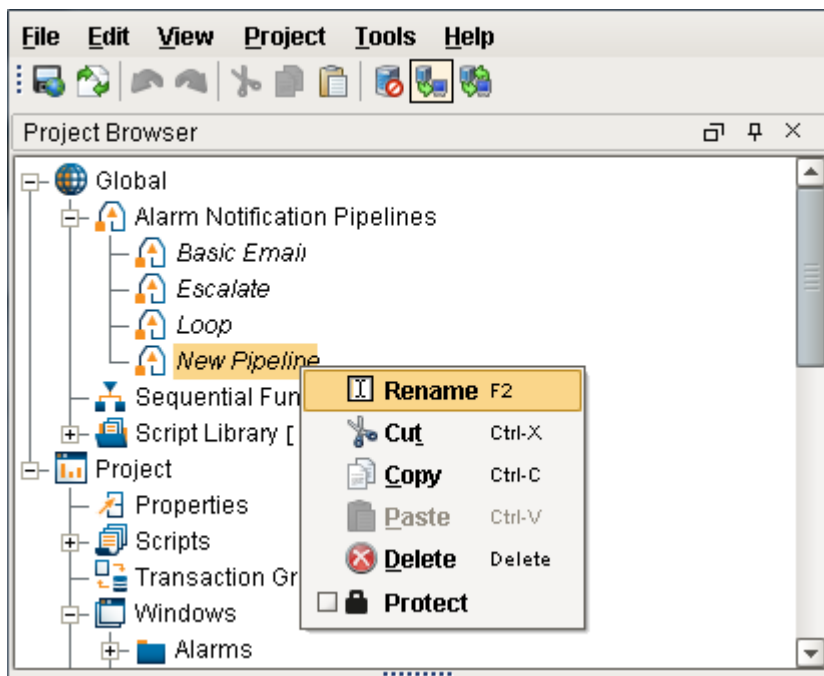
2. Select **Alarm Notification Pipelines**, and right-click to create a **New Alarm Notification**.



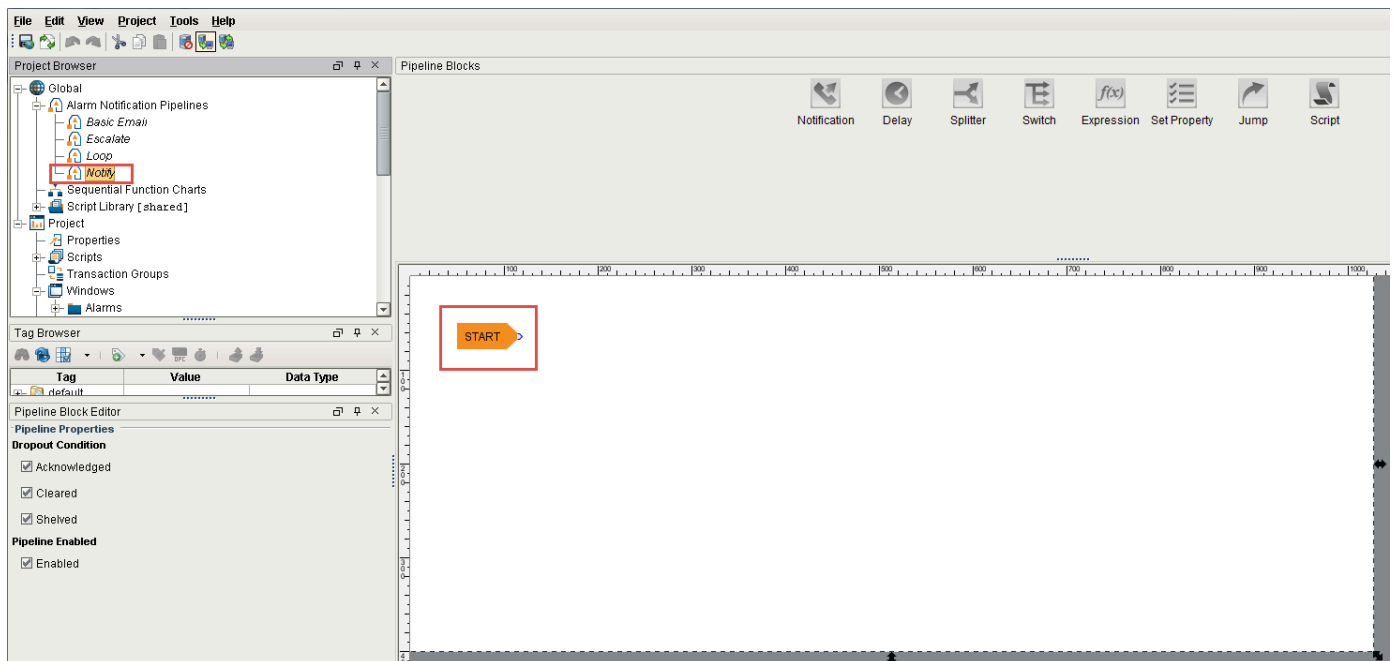
When you create a new pipeline, the Designer switches to the Pipeline Workspace. The workspace is essentially a blank canvas and can contains various **Pipeline Blocks**. By default, the name of the new pipeline is called "**New Pipeline**".



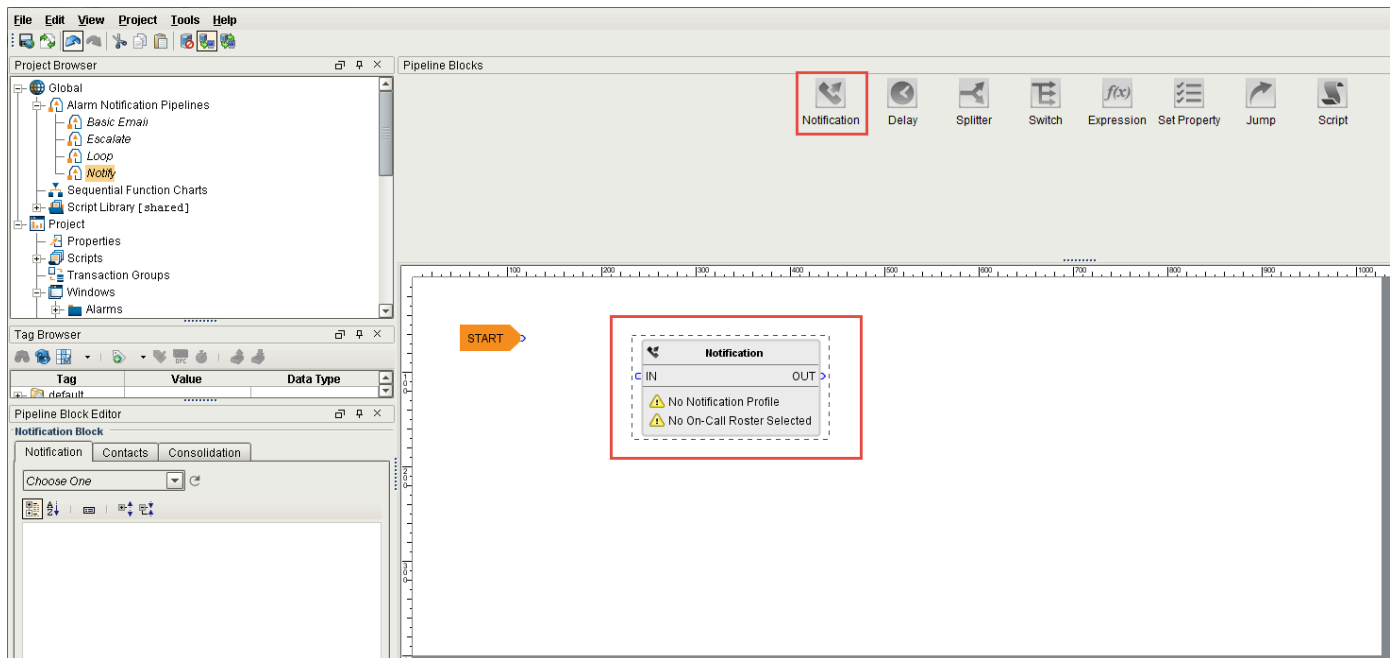
3. It is a good practice to give a meaningful name to the pipeline. Commonly, the name acts as a way to describe what the pipeline does. Select your '**New Pipeline**', right click and **Rename** it to '**Notify**'.



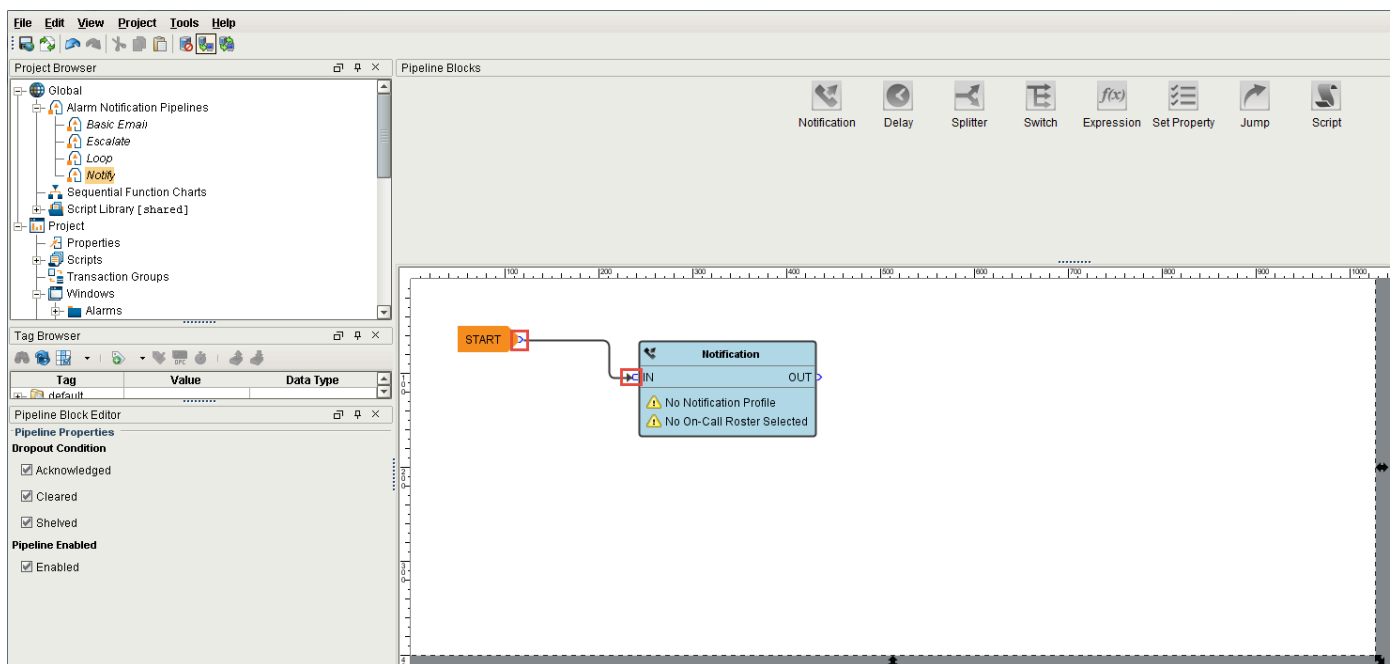
Every pipeline has a **Start Block** which you will see when you create your pipeline. When an alarm is triggered, it automatically goes to the **Start Block**, and from there you can use other pipeline blocks to determine what you want to do with that alarm. All the available pipeline blocks are shown at the top of the Pipeline Workspace.



4. The most basic pipeline you can create is to simply notify someone right away when an alarm occurs. Drag the **Notification Block** on to your canvas and place it next to the **Start Block**. Keep in mind that even though the two blocks are on your canvas, they are not yet connected.

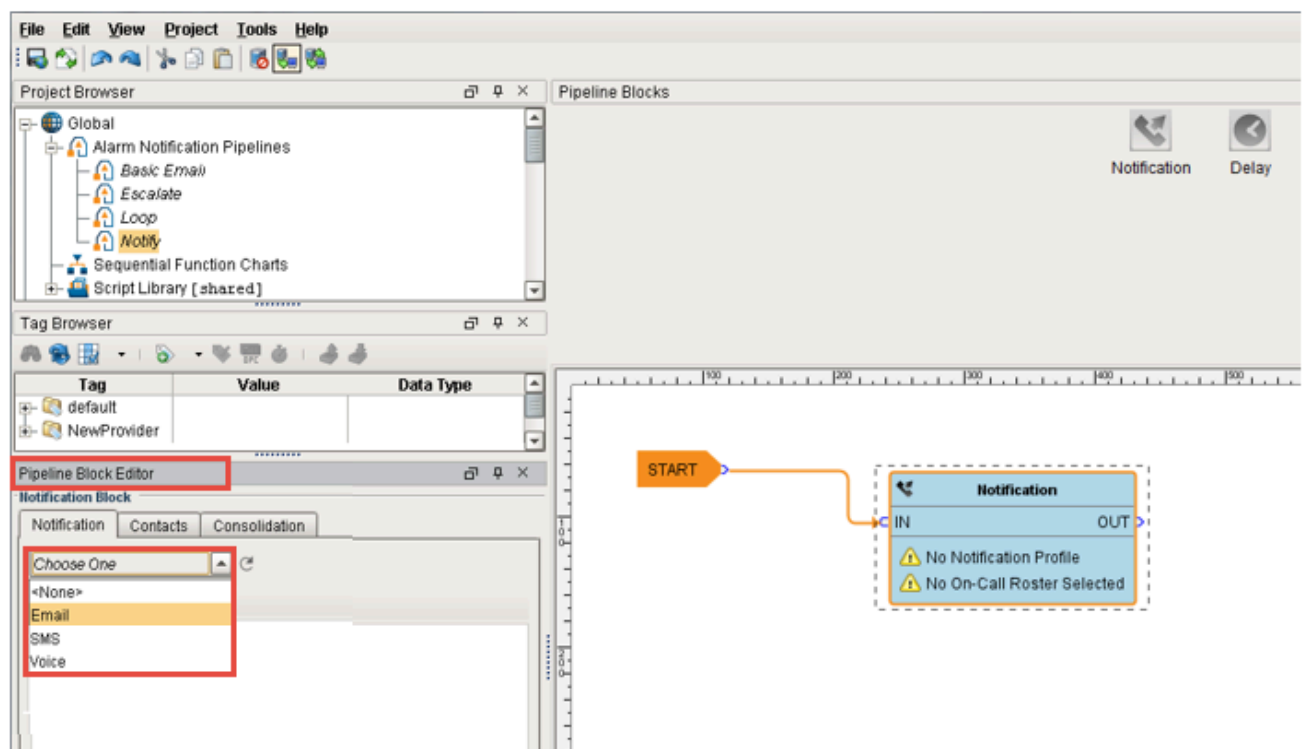


5. Drag the output pin from the **Start Block** to the input pin of the **Notification Block** connecting the two blocks. When an alarm occurs, it's going directly into the Start Block and then into the Notification Block.

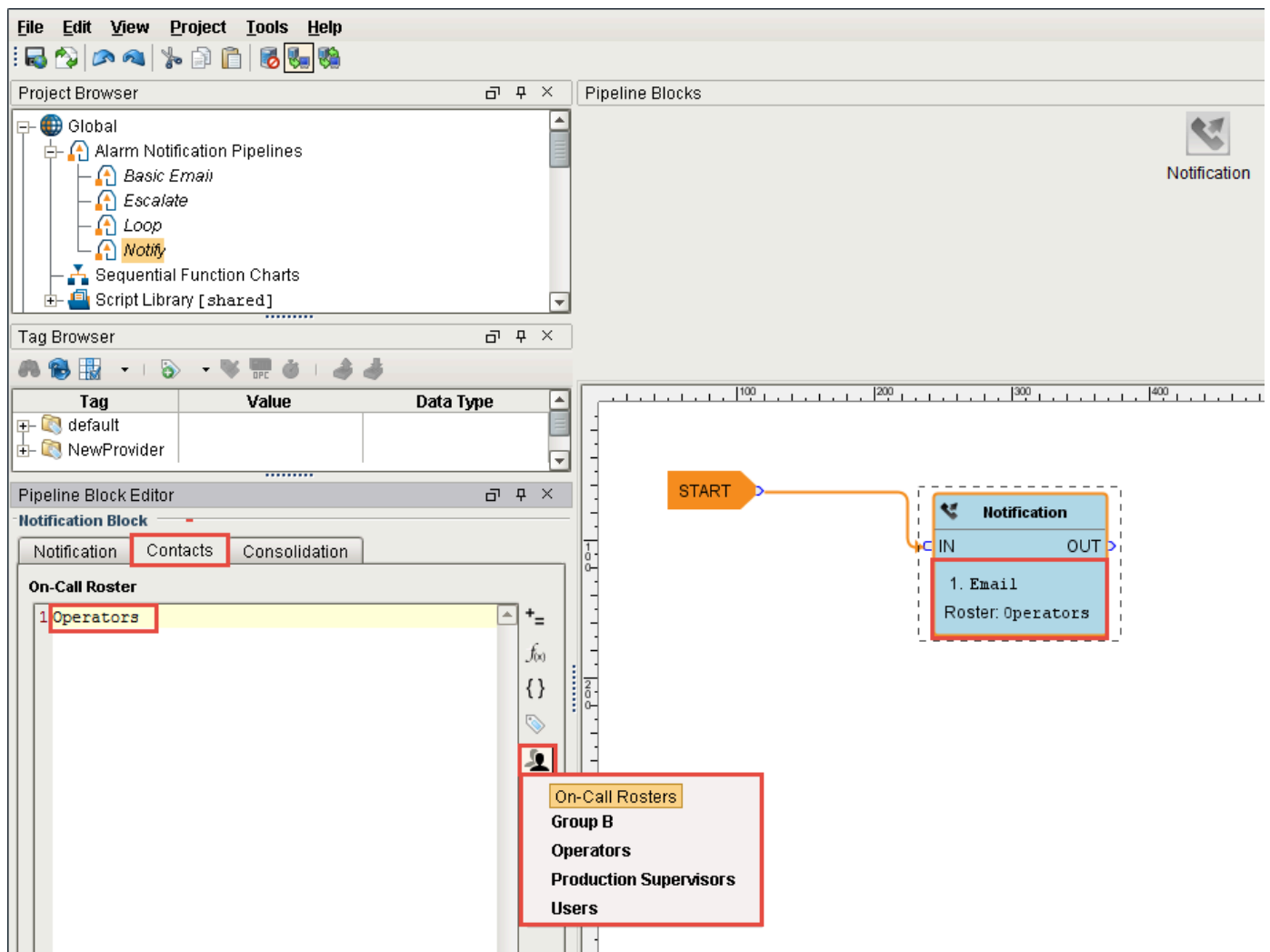


6. On the **Notification Block**, you have to configure the **Alarm Notification Profile** that you are going to use and who you are going to send it to. You should have already configured your Alarm Notification Profile before creating your new pipeline.

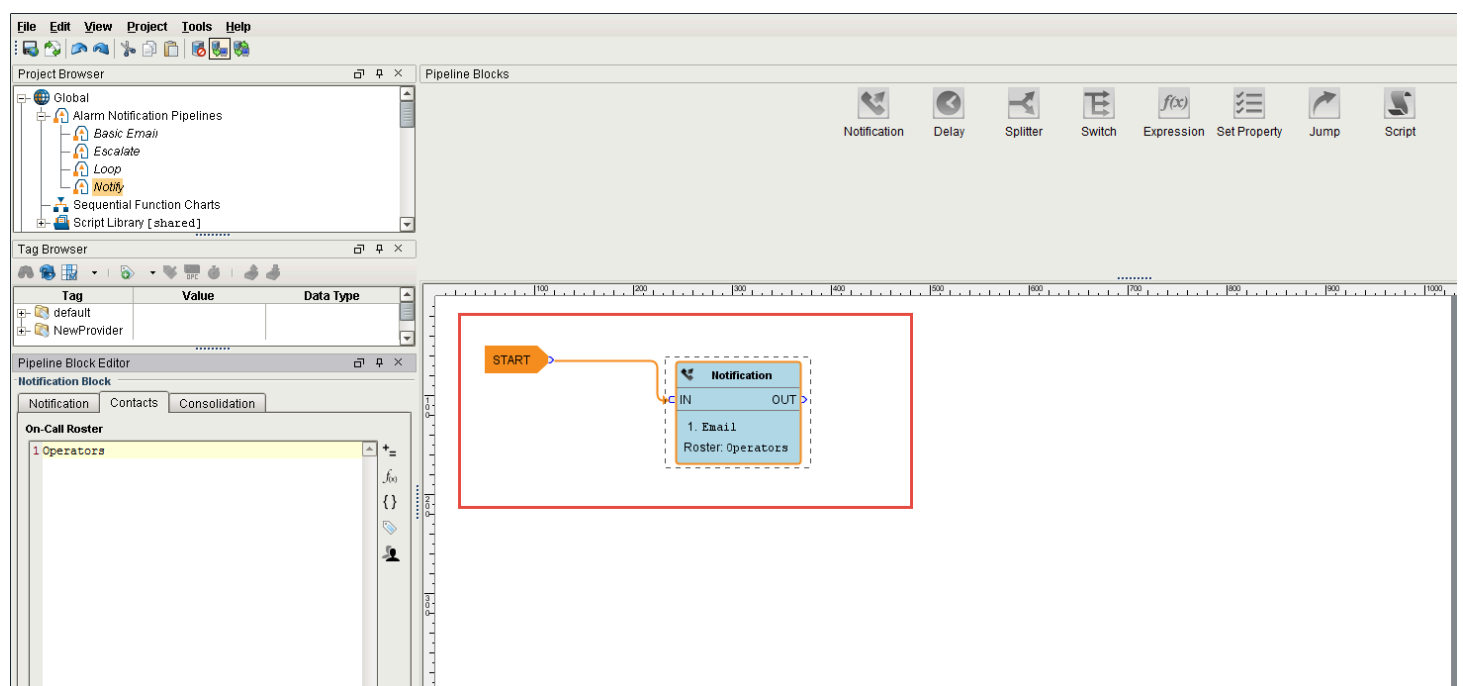
In your pipeline workspace, click on the **Notification Block**, and the Notification tab will open. Choose which **Notification Profile** from the dropdown that you want to use: **Email**, **SMS** or **Voice**. You can see that all three profiles are configured, but for this example, choose **Email**.



7. In the **Pipeline Block Editor**, under **Contacts**, click the **On-Call Roster** icon on the right side. Select from users and groups who you want to notify about this alarm. Your On-Call Roster should have already been created before creating your pipeline.



You just created a very simple Alarm Pipeline. When this alarm becomes active, it goes directly to the Start Block and notifies all the users in the Operator group via Email one time.



Pipelines don't do anything until an alarm is told to use it. Once you configure your Alarm Notification Pipeline, the next step is to tell your alarm to use your 'Notify' pipeline when the alarm becomes

Active, Cleared or Acknowledged. How to make your alarm use a pipeline is covered in the next section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Adding Pipelines to Tags



INDUCTIVE
UNIVERSITY

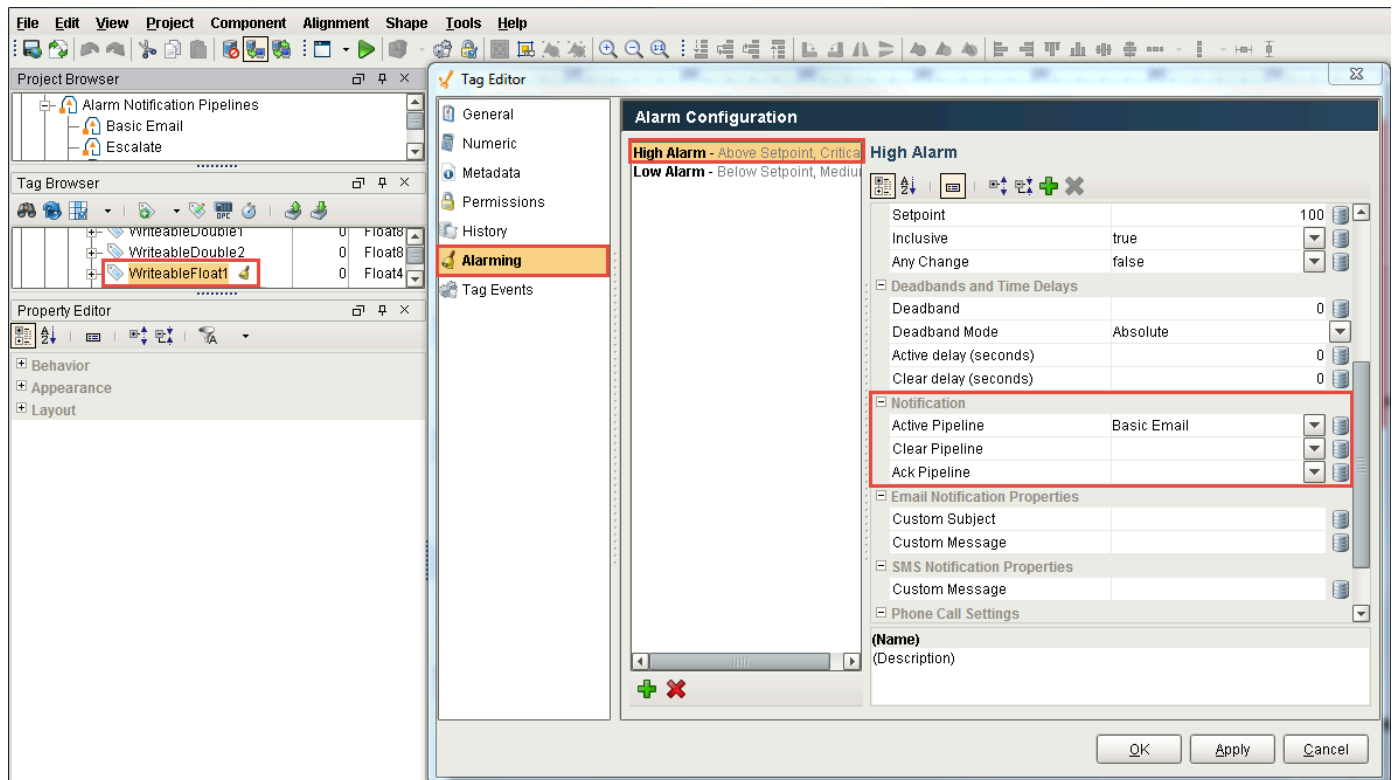
Adding Pipelines to Tags

[Watch the video](#)

Any alarm in Ignition can use an existing alarm notification pipeline on the local Gateway and/or remote Gateway. Alarms can share pipelines resulting in a single pipeline providing notification services to many alarms. When an alarm becomes active, cleared, or acknowledged it can execute a designated alarm pipeline. The information about the alarm is ported automatically into the alarm pipeline.

To Associate a Pipeline with the Alarm

1. Double-click on a Tag that already has an alarm configured (i.e., WritableFloat1). The **Tag Editor** is displayed and you can access its properties.
2. Go to the **Alarming** tab.
3. Select an Alarm. If you don't have an alarm configured, create one. To learn more about creating alarms, go to [Configuring Alarms](#).
4. Under the **Notification** section, select the desired alarm pipeline from the drop-down for any of the alarm status states. Typically, operators want to know when an alarm goes active. This example uses the Basic Email pipeline when the alarm status becomes Active.
5. Click **OK**.



Now, every time the High Alarm Tag goes Active, it will use the logic in the Basic Email pipeline to notify the operators.

Alarm Pipeline Events

Each alarm has 3 events that can push it into a pipeline: Active, Clear, and Acknowledge (Ack). While they all work the same way, it is helpful to know a few things about each event. For all pipelines, the default settings are to drop alarms from the pipeline if that alarm is Acknowledged, Cleared, or Shelved. This means each Alarm Pipeline Event requires slight changes to these properties, so it is a good idea to have special "cleared" or "shelved" versions of any pipelines that will be called from these other events.

Active Pipeline

The Active Pipeline event is the most commonly used event for sending alarm notification. This happens when the alarm goes into the active state (alarm has happened). The default settings are commonly OK because any Acknowledged, Cleared, or Shelved alarms are usually not considered active anymore.

Clear Pipeline

The Clear Pipeline event is the most commonly used event for sending notification that the issue is no longer active, but may still need attention. This happens when the alarm goes from the active state to

the cleared state. The default setting to **drop out if "Cleared"** is **NOT OK** because any alarm event that started as a "Clear" event will always drop out at the Start Block.

Ack Pipeline

The Acknowledge (Ack) Pipeline event is the most commonly used event for telling your users that someone is working on the problem. This happens when the alarm goes into the acknowledged state (usually an operator pressing the Acknowledge button). The default setting to **drop out if "Acknowledged"** is **NOT OK** because any alarm event that started as an "Acknowledged" event will always drop out at the Start Block. It is usually a good idea to also set the "Cleared" dropout condition to False since any acknowledge notification is probably wanted regardless of if the alarm is cleared.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Pipeline Blocks



INDUCTIVE
UNIVERSITY

Pipeline Blocks

[Watch the video](#)

Pipeline Blocks are the building blocks for the Alarm Pipeline Notification system. Each block has an input and zero or more outputs. Each block has its own unique functionality and performs a specific action for the pipeline, such as sending a notification, setting a property, or evaluating an expression.

Let's look at the different pipeline blocks that come with the Alarm Notification system. There are eight blocks above your Pipeline Workspace that you can choose from to build your alarm notification pipeline.

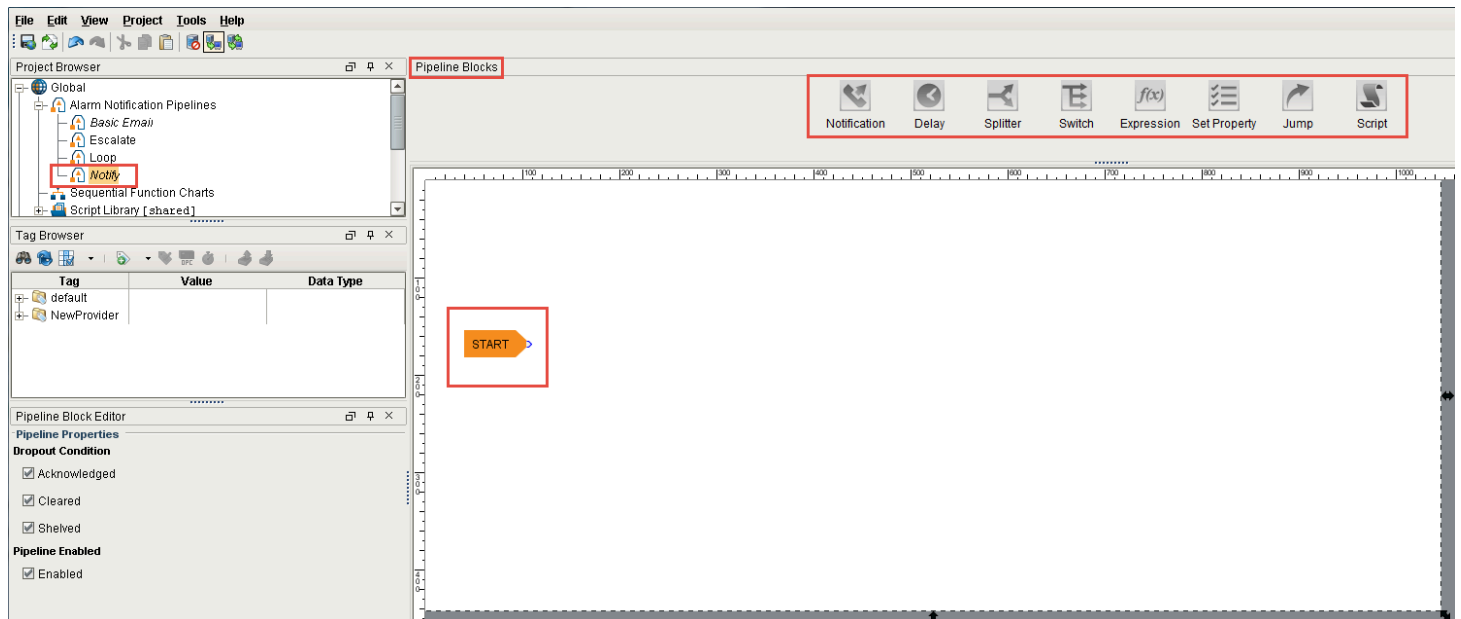
Pipeline Blocks

The **Pipeline Blocks** are described individually in this section, but here is a brief overview:

- **Notification** Delivers a notification through the selected Notification Profile (i.e., Email, SMS and Voice).
- **Delay** Blocks the alarm event for the specified amount of time.
- **Splitter** Forwards a single event for the specified amount of time.
- **Switch** Evaluates a non-boolean Expression, and forwards to an output based on the result.
- **Expression** Evaluates an Expression that results in either "True" or "False", and forwards the event to that output.
- **Set Property** Evaluates an Expression, and sets the result as a runtime pty on the alarm event.
- **Jump** Forwards the alarm event to a different pipeline.
- **Script** Executes a task outside the pipeline.

Start Block

When you create a new pipeline, you always get a **Start Block** placed in your Pipeline Workspace. You cannot delete the **Start Block** because it represents the entry point for alarm events into the pipeline.



Notification Block

The **Notification Block** is the block that controls the actual notification. When an alarm event reaches this block in the pipeline, some sort of notification will go out. In it, you can setup what type of notification it will use and who it will notify. You can learn more about the **Notification Block** by going to the [Notification Block](#) page.

Delay Block

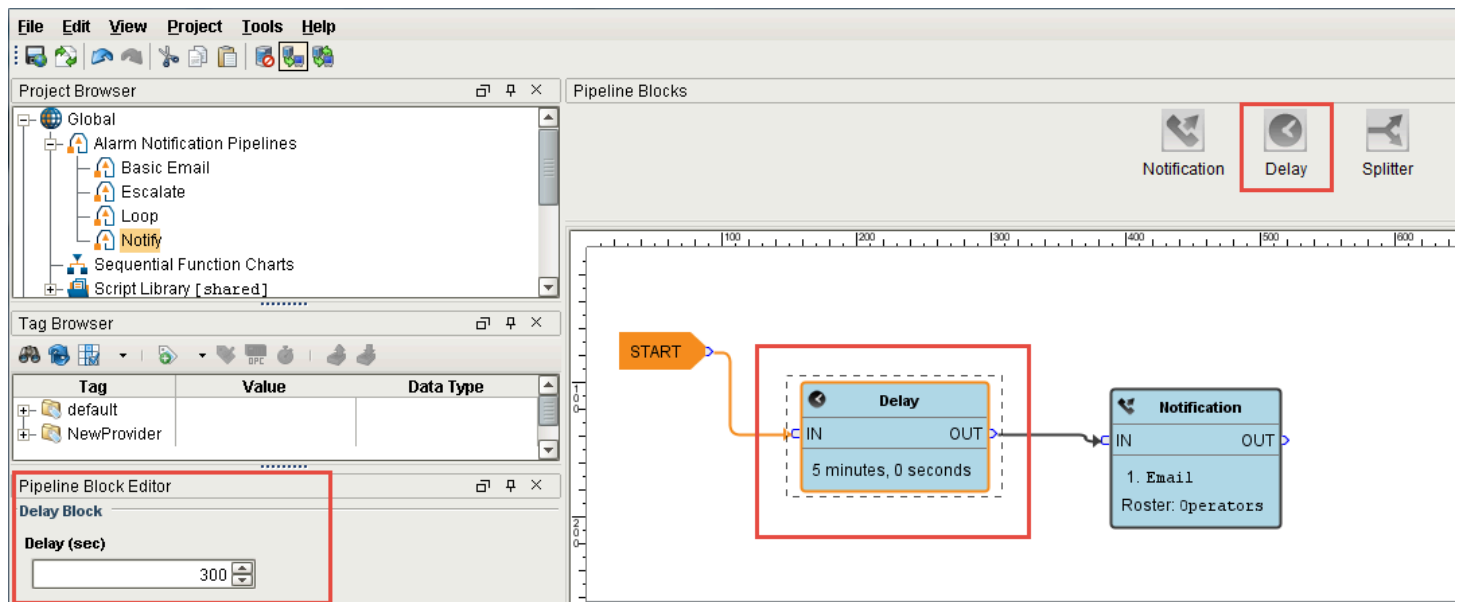
The **Delay Block** simply blocks alarm events for a specified period of time before moving them to the next block. They are generally used to wait for the dropout condition to become satisfied for an alarm.

For example, a 5 minute delay might be used to give operators viewing control screens a chance to acknowledge, and only send notifications if they haven't (the "active delay" deadband on the alarm could be used to delay the alarm as well, but in that case it wouldn't actually be active, and therefore not displayed, for the delay time).

Delays are often also used to control flow in a pipeline. For example, in an "escalation" scenario, a notification block might be used to send emails. Since emails are sent instantly, and acknowledgment occurs "asynchronously" (the user must see the email and click the link or log into the system), a

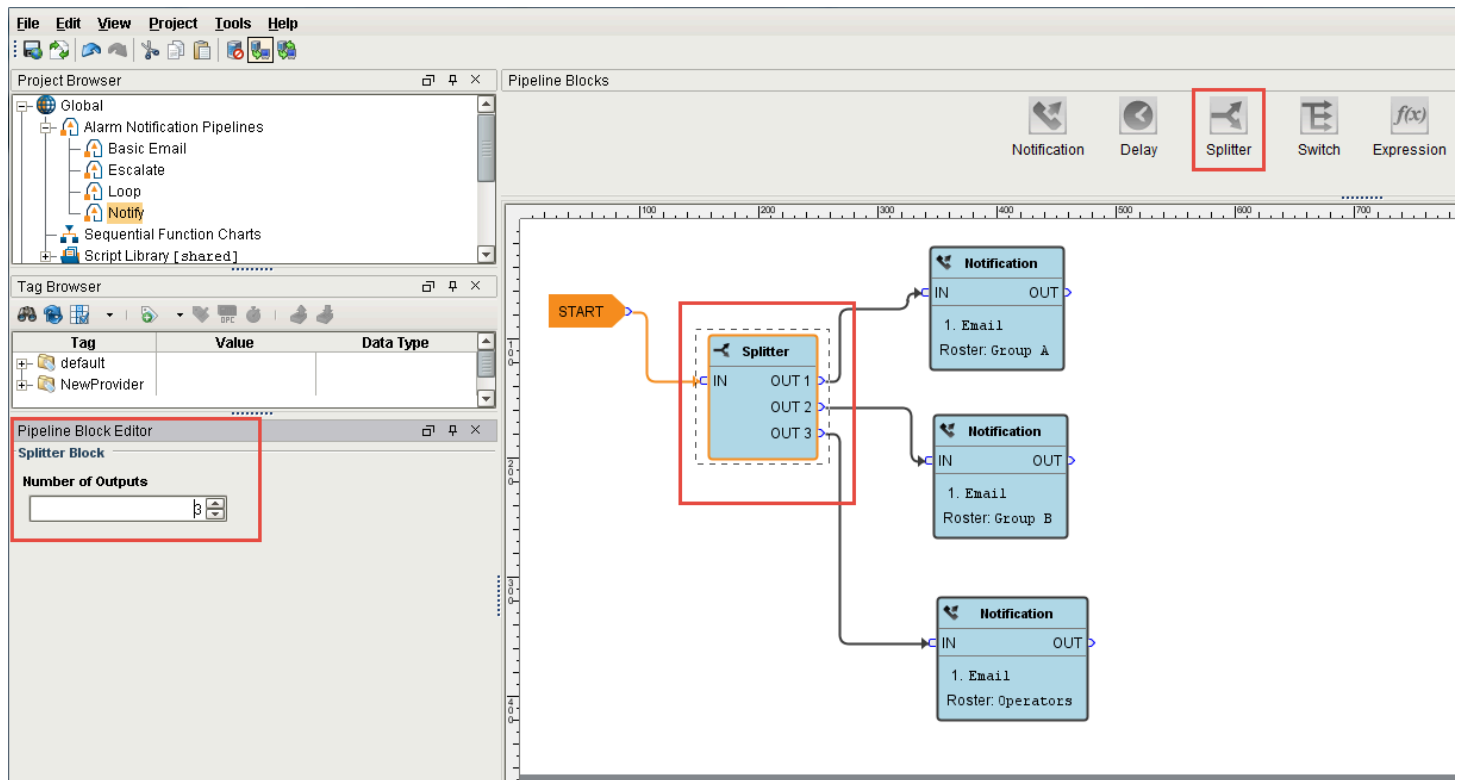
delay block could be used to provide time for the user to acknowledge, before moving on to a voice notification block. There is no practical limit to how long a delay can be. The delay is calculated independently for each event entering the block, meaning that each event will be held for the specified time.

Click the **Delay Block** and a spinner will appear in the Pipeline Block Editor. You can specify the number of seconds to delay. In this example, if you enter '300' seconds, the alarm will wait 5 minutes before sending the notification to the Operators.



Splitter Block

The **Splitter Block** simply forwards a single alarm event to multiple outputs at the same time. It creates a copy of the alarm event for each output and executes the alarm notifications along multiple paths in parallel. You can specify as many outputs as you want. If care is not taken, it is possible to end up with an exponential number of alarm event copies active in the pipelines at one time. Since each notification block operates on a specific on-call roster, Splitters are useful for delivering events to multiple notification blocks at once as shown in this example.

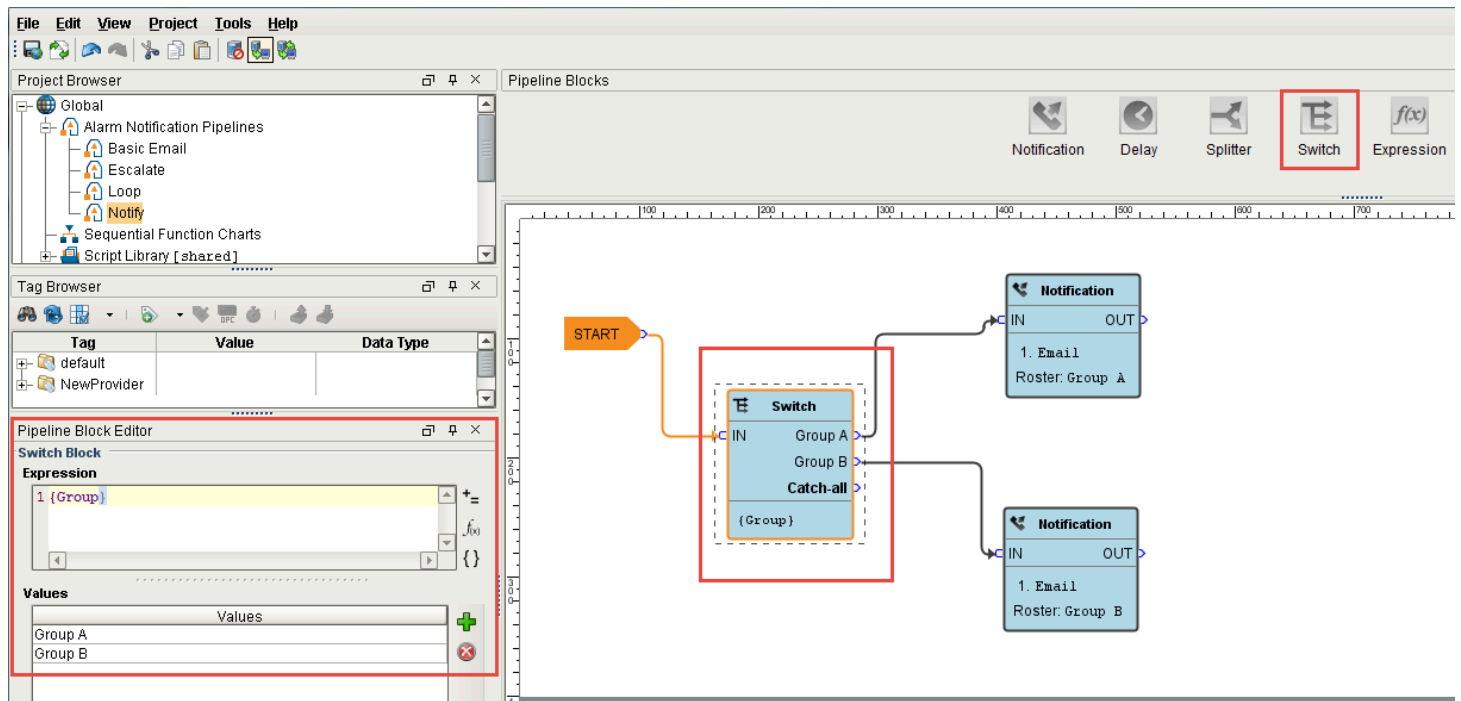


Switch Block

The Switch Block evaluates a non-boolean expression and forwards it to an output based on a given result. It can be a number or string, and it will be looked up against a defined list of expected results. If a match is found, the alarm will follow that path, otherwise the "Catch-all" output is used.

You can create an expression and even use the alarm event's priority, associated data, value and more. The following example has associated data for the 'Group' that they belong to with 'Group A' and 'Group B' values. The Switch Block will evaluate the expression results against these defined values, and if a match is found, the alarm will follow that path. If neither of the values are met, the Catch-all output will be used.

This example demonstrates that a pipeline can send out an alarm to 2 different lists of people based on the '**Group**'.



Expression Block

The **Expression Block** contains an Expression which is evaluated against each alarm that enters it. The result is expected to be a boolean value, either 'True' or 'False' and forwards the alarm to the specific output. The Expression executed by the block is written in the same syntax as other Expressions in Ignition. However, in this context, it is possible to refer directly to properties of the alarm event that is being evaluated by using the standard `{path}}` reference syntax. In addition, the following functions are available to quickly determine the state of the alarm.

Examples

isActive()

This single function returns whether the current event is active or not. An Expression block like this could be used to then dispatch to an "active" pipeline, and a "clear" pipeline (both the active and clear pipeline settings on the alarm would be set to this dispatch pipeline). This kind of setup allows you to later modify how actives are handled vs. clears in one place, without having to modify many alarms.

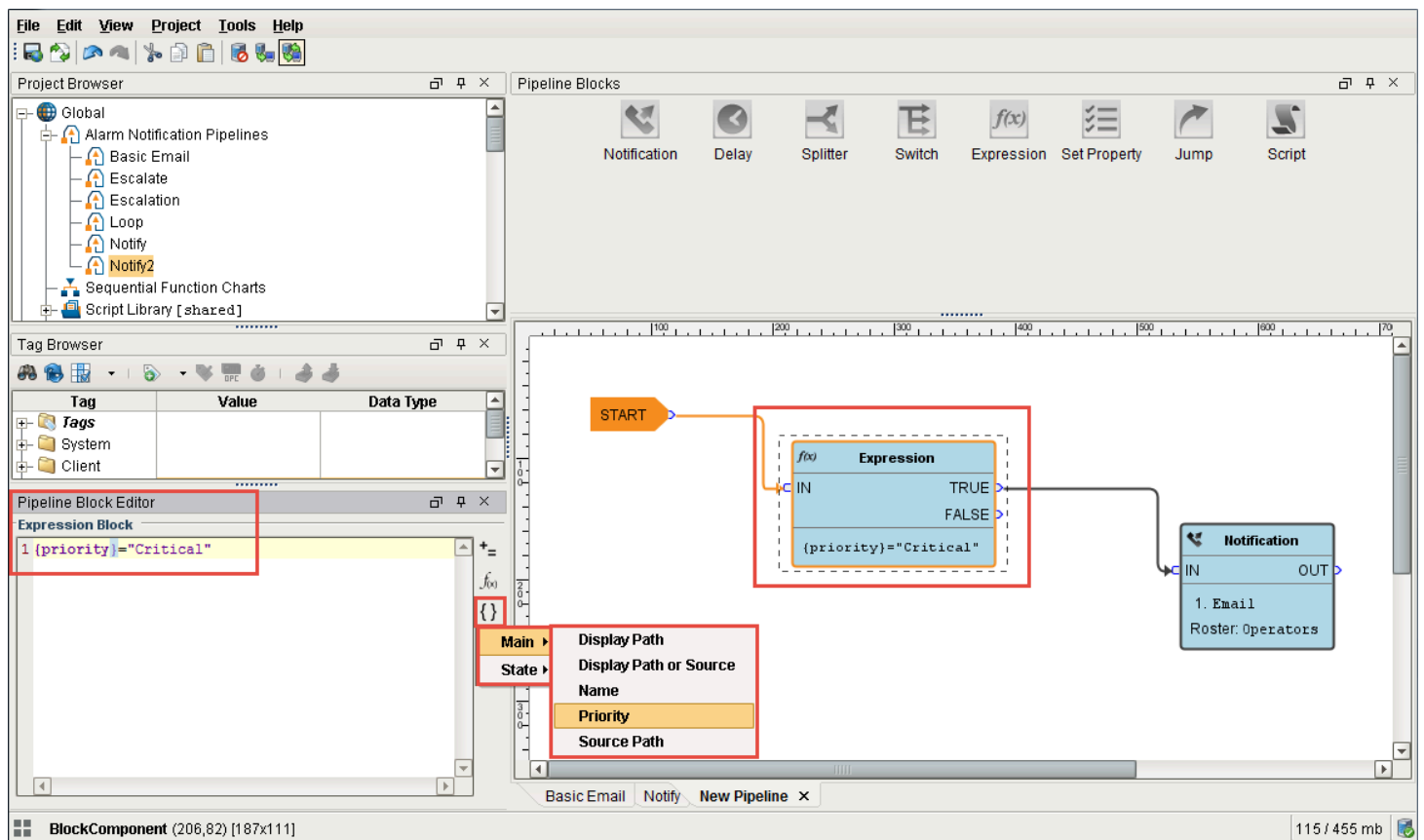
toInt({priority})>2 && {displayPath} like "East Area"

This block would forward all High and Critical alarms from the "east area" to the true path. The others would go to false, which may not actually be bound to another block, making this Expression block act like a filter.

isPropertyDefined("isEscalated")

This Expression checks if a property exists on the event. The "isEscalated" property is not a system defined property. Instead, in this example, it might be set using a Set Property block before forwarding back to the start of the pipeline. The first time through, this Expression would fail, but the next time, it would pass, and the "escalated" behavior could be executed.

In the **Pipeline Block Editor**, you can select the Alarm Properties from any of the icons in the Expression Block. This example uses an expression (`{priority}="Critical"`) to check to see if the priority is 'Critical,' and if so, the alarm notification will follow the 'True' path. If it is not 'Critical', it will follow the 'False' path. As you can see, the 'False' output is not configured. You don't have to configure every output if you don't want to.



Set Property Block

The **Set Property Block** allows you to define or redefine a property on the alarm event. The value is created from the Expression Block, which can refer to other properties or the same properties. Typically, the Set Property Block is used as a counter for the number of times the alarm has been looping or notifying people in your on-call roster. Settings modified in this way will only exist while the alarm is in the pipeline, they will not be stored to the journal, or show up in the status table.

Examples

If you want to attempt notification up to three times before stopping, you could create a pipeline that looked like [Notification Block] > [Set Property] > [Expression], with the Expression looping back to the notification block (perhaps with a delay in between).

The Set Property block would look like:

Property Name: counter

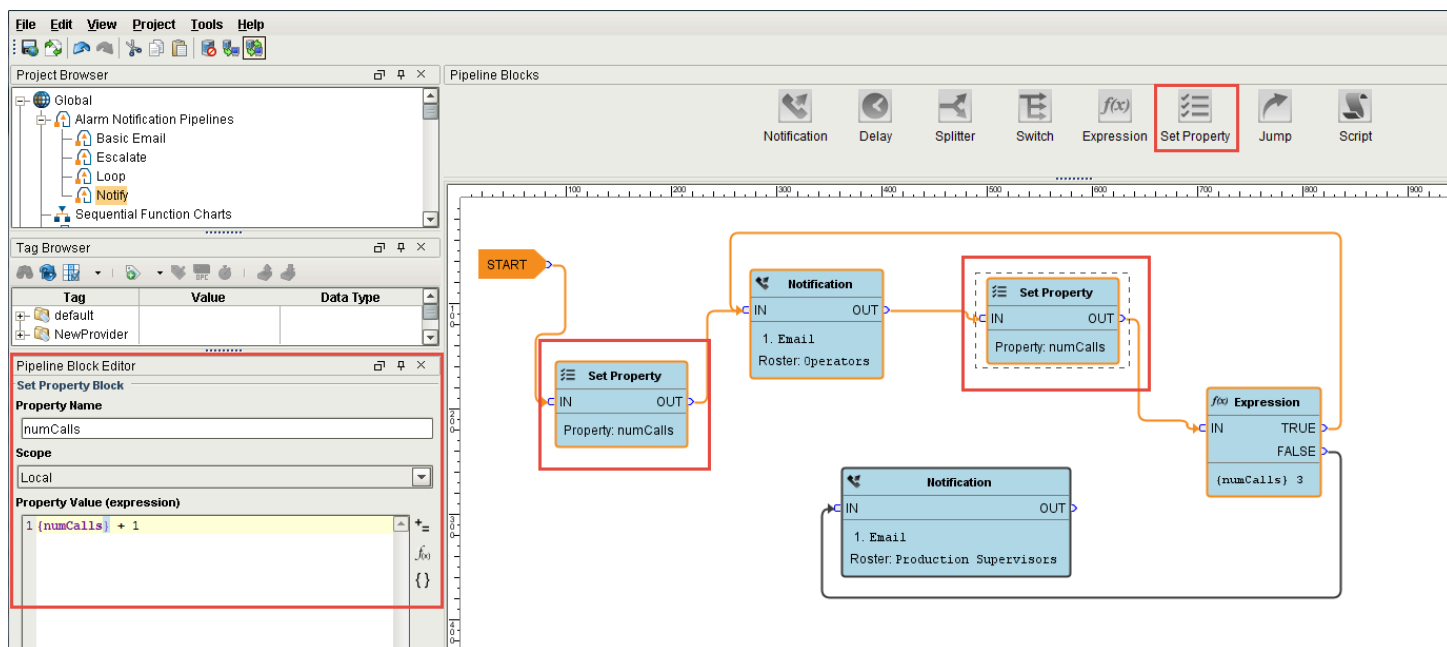
Expression: `coalesce({counter}, 0) + 1`

Note that the first time the block is hit, the counter property will not exist, and therefore the value will be null. The coalesce function returns the first non-null value, allowing us to start with 0. Another way to write this might be:

```
if(isPropertyDefined("counter"), getProperty("counter"), 0)+1
```

The `getProperty()` function is functionally equivalent to simply referencing the property in brackets (that is, `{counter}`). The Expression block in this example would simply be: `{counter}<3`.

Here is another example where the operators are notified of an alarm event 3 times before being escalated to the Production Supervisors. Drag 2 **Set Property Blocks** to your pipeline workspace. Assign the first Set Property block to a Property Name 'numCalls' and a Property Value of '0'. The second Property Block you need to increment the counter. Enter `{numCalls} + 1` as shown in the Pipeline Property Editor. Each time the alarm goes through this block, the 'numCalls' variable increments by '1.' If no one acknowledges the alarm, it is sent to the Production Supervisor's on-call roster.

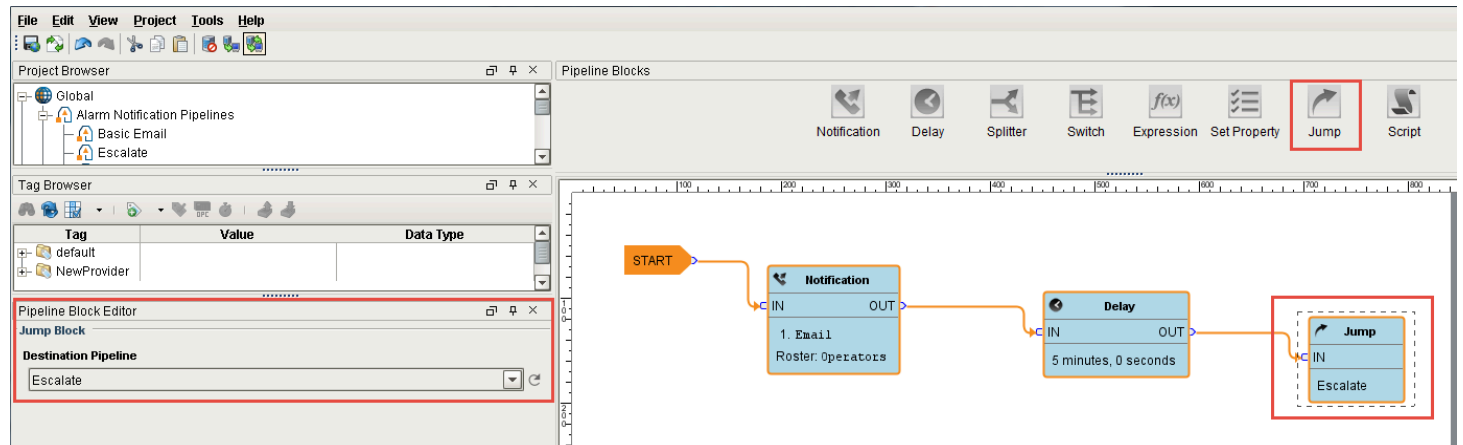


Jump Block

The **Jump Block** forwards an alarm event to a different pipeline. You can breakup a pipeline into several different pipelines and use the Jump Block to go back and forth between pipelines. This is perfect for an escalation pipeline because often times they can become one massive pipeline, and the Jump Block can make the escalation pipeline less complicated and more compact.

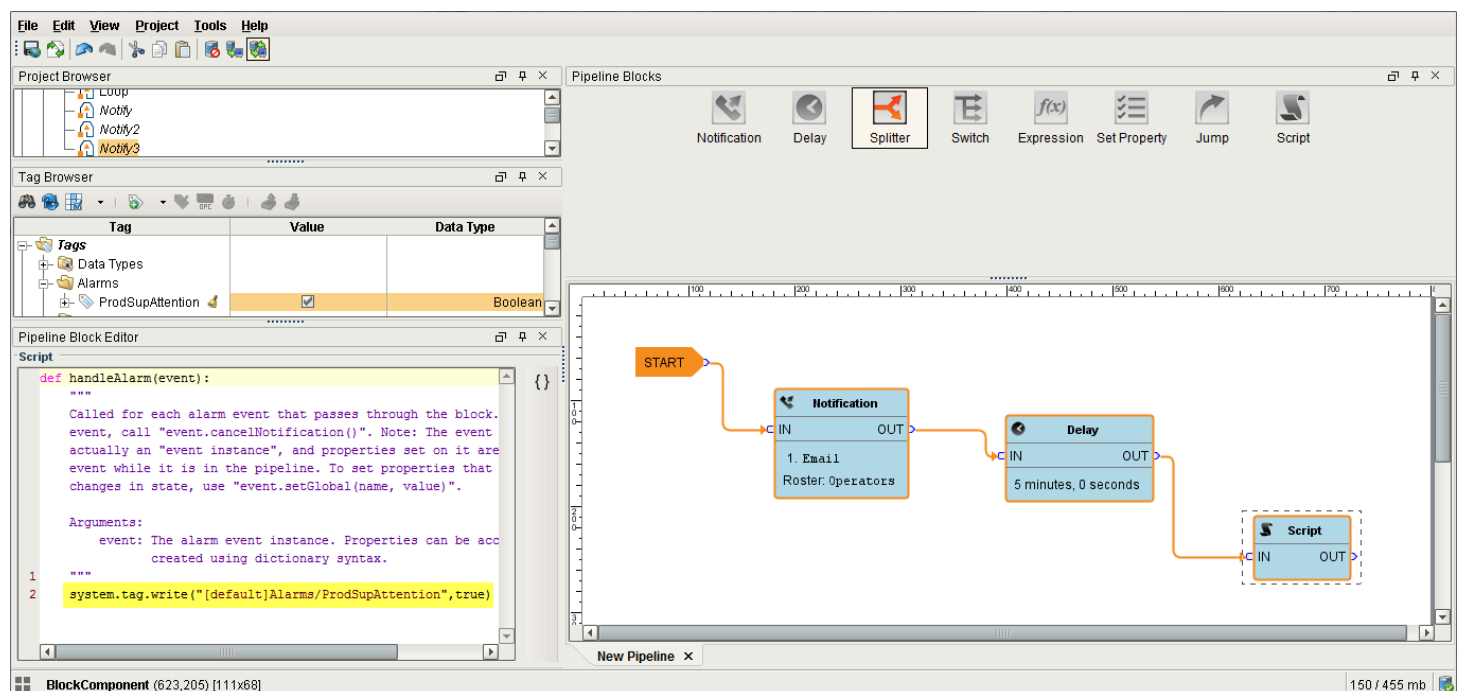
In the Pipeline Block Editor, enter the Destination pipeline called '**Escalate**'.

In this simple example, the **Jump Block** forwards the alarm to the Escalate pipeline and ends there.



Script Block

The **Script Block** allows you to execute a task outside the pipeline like writing to another alarm tag or database. In this example, the Script Block was used to notify the production supervisors of an alarm using another alarm tag as shown in the Pipeline Block Editor.



In the next sections, you will see how to use a combination of these different pipeline blocks to create your own unique pipelines.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Notification Block

As the name suggests, the **Notification Block** is responsible for sending notifications.

Each notification block can target multiple notification profiles, which represents different methods of notification, but it will only notify the user on the first profile that the user has contact information for. So having a notification block with both email and sms listed will first attempt to contact a person with the profile type listed first. If that fails because the user does not have contact information for that profile type, it will try the second profile. Each type of notification profile will have its own relevant properties for configuration, although some features are common to all profile types. The various profiles will also behave differently, according to their capabilities, in regards to how notifications are ordered. For example, the Email profile is capable of sending a message to many recipients at once, while the SMS and Voice notification profiles must step through each contact sequentially.

There are two required settings for notification blocks: the Notification Profile and the On-Call Roster. The profile will dictate the method through which notifications occur, and the on-call roster will dictate who receives the notifications. The on-call roster provides an ordered list of contacts, which will first be pared down according to each user's schedule. The resulting contacts will be notified in order, based on how the profile operates. The settings displayed will depend on the type of profile selected.

Notification

The notification tab allows you to select which **notification profile** you would like to use of the ones that you have currently configured in your gateway. The settings that can be configured differ, depending on the type of notification profile selected. However, all of the notification types share a setting, outside of the property area called **Delay Between Messages**. This property allows you to set a delay that will happen in between each notification. So if your delay is set to 30 seconds on an email notification, it will email the first person in the roster and then wait for 30 seconds before emailing the next person. It is set to zero seconds by default.

Voice Settings

- **Require PIN:** If true, the user will need to enter their PIN in order to hear the alarm notification messages. The user's PIN is defined in the user management section of the Gateway. If false, then anyone will be allowed to hear the messages after answering the phone.
- **Allow Acknowledgement** - If false, users will only be allowed to listen to alarms, not to acknowledge them.
- **Retries Per Contact** - The number of retry attempts per contact per alarm. The default is set to 1.
- **Delay Between Retries** - Introduces a delay between calling each contact, for each alarm.
 - The pipeline dropout conditions are checked regularly between calls and while waiting, so this would provide time for the alarm to drop out before calling the next person.
 - The delay is only enforced after following a "successful" call (a call that was answered). Unanswered or errored calls will move on to the next contact right away. Note that long delays can block other alarms in the call queue.
 - The delay is applied to all contacts for a particular alarm. **For most phone systems, a delay of at least 1 second is advised.**
 - The system must have time between each call to hang up properly before dialing again.
- **Test Mode** - If enabled, messages will be logged to the console indicating when and to whom calls would have been made, without actually making a call.

The screenshot shows the 'Pipeline Block Editor' window. The 'Notification Block' is selected, and the 'Voice' block is highlighted. The configuration table for the 'Voice' block is as follows:

Voice	
Require PIN	false
Allow Acknowledgement	true
Retries Per Contact	0
Delay Between Retries (Seco...	0
Test Mode	false

At the bottom, there is a 'Delay Between Messages (seconds):' field with a value of 0 and a spinner control.

SMS Settings

- **Message** - The message to use for single events.
- **Consolidated Message** - Like email, the message to use when multiple events are sent together, due to the block's consolidation settings.

- **Delay Between Notification** - As with voice, a delay between each message sent, to give the recipient time to acknowledge and prevent further notification.
- **Test Mode** - If enabled, logs messages that would have been sent to the Gateway console, and does not actually send them.
- **New in 7.9.1** **Acknowledge Message** - When two-way SMS notification is enabled, this property allows for the configuration of the acknowledgment instructions the users receive in the notification. The '%s' will be replaced by the user's acknowledgement code.

The screenshot shows the 'Pipeline Block Editor' window with the 'Notification Block' selected. The 'Notification' tab is active, showing a list of notification types with '1. SMS' selected. Below the list, there are icons for adding, removing, and reordering blocks. The 'SMS' configuration table is visible, showing fields for Message, Consolidated Message, Acknowledge Message, and Test Mode. At the bottom, there is a 'Delay Between Messages (seconds)' field set to 0.

SMS	
Message	At {eventTime hh:mm:ss}, alarm
Consolidated Message	{alarmEvents.Count} alarm ev
Acknowledge Message	To acknowledge, reply '%s'
Test Mode	No

Delay Between Messages (seconds):

Message Properties

Both the email and SMS notification types have a few different message settings. There are properties that contain alarm values that can be used in these message settings. To use the property, place "{}" around it. The table below contains all of the properties available to them.

Property Name	Property Display Name	Description
Count	<code>{AlarmEvents.Count}</code>	The number of alarm events that are in the pipeline. Typically used with alarm consolidation.
Maximum Priority	<code>{AlarmEvents.MaxPriority}</code>	The maximum priority of the alarms in the pipeline. Typically used with alarm consolidation.
Minimum Priority	<code>{AlarmEvents.MinPriority}</code>	The minimum priority of the alarms in the pipeline. Typically used with alarm consolidation.
Display Path	<code>{displayPath}</code>	The display path of the alarm.
Display Path or Source	<code>{displayPathOrSource}</code>	The display path of the alarm, or the sourcepath if no display path was defined.
Name	<code>{name}</code>	The name of the alarm.
Priority	<code>{priority}</code>	The priority of the alarm.
Source Path	<code>{source}</code>	The source path of the alarm.
Ack Time	<code>{ackTime}</code>	The time when the alarm was acknowledged.
Ack'ed By	<code>{ackUser}</code>	The username of the person that acknowledged the alarm.
Ack'ed By (Name)	<code>{ackUserName}</code>	The name of the person that acknowledged the alarm.
Acked?	<code>{isAcked}</code>	A True or False on if the alarm is acknowledged.
Active Time	<code>{activeTime}</code>	The time the alarm went active.

Property Name	Property Display Name	Description
Active?	<code>{isActive}</code>	A True or False on if the alarm is active.
Clear Time	<code>{clearTime}</code>	The time the alarm went to a cleared state.
Clear?	<code>{isClear}</code>	A True or False on if the alarm is cleared.
Current State	<code>{state}</code>	The current state of the alarm. (Active, Unacknowledged/Active, Acknowledged/Cleared, Unacknowledged/Cleared, Acknowledged)
Event State	<code>{eventState}</code>	The state of the alarm event. (Active, Acknowledged, Cleared)
Event Time	<code>{eventTime}</code>	The time that the alarm event occurred. Can format the time to a specific format like: <code>{eventTime hh:mm:ss}</code>
Event Value	<code>{eventValue}</code>	The value of the tag for the alarm event.
Is Shelved?	<code>{isShelved}</code>	Is the alarm shelved.

Contacts

New in 7.9.4



INDUCTIVE
UNIVERSITY

Dynamic Rosters

[Watch the video](#)

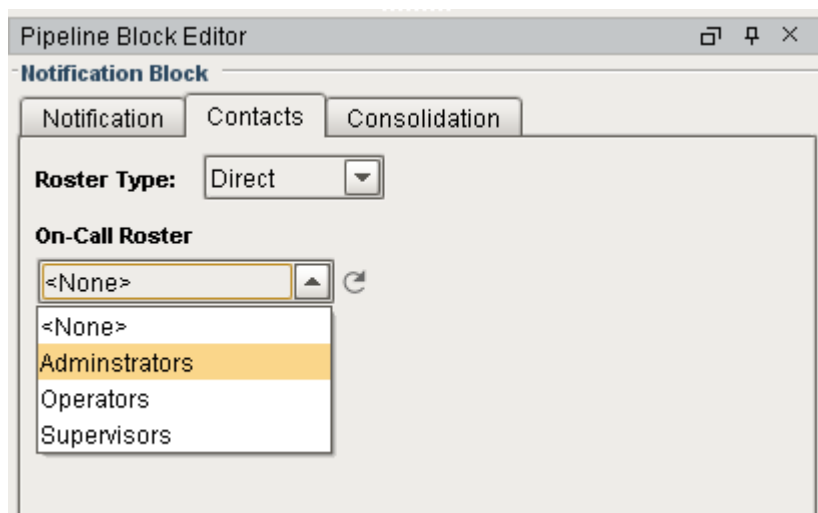
From the Contacts tab, you can specify who you want to send the alarm notification to by specifying what roster type you want to use. There are three different ways to setup a roster:

- **Direct** - Set a specific roster to use.
- **Expression** - Set up an expression that will determine which roster to use.
- **Calculated** - Use scripting to build a roster within the pipeline.

Each has different advantages, so it's important to understand the difference so you can select the method that best fits your needs.

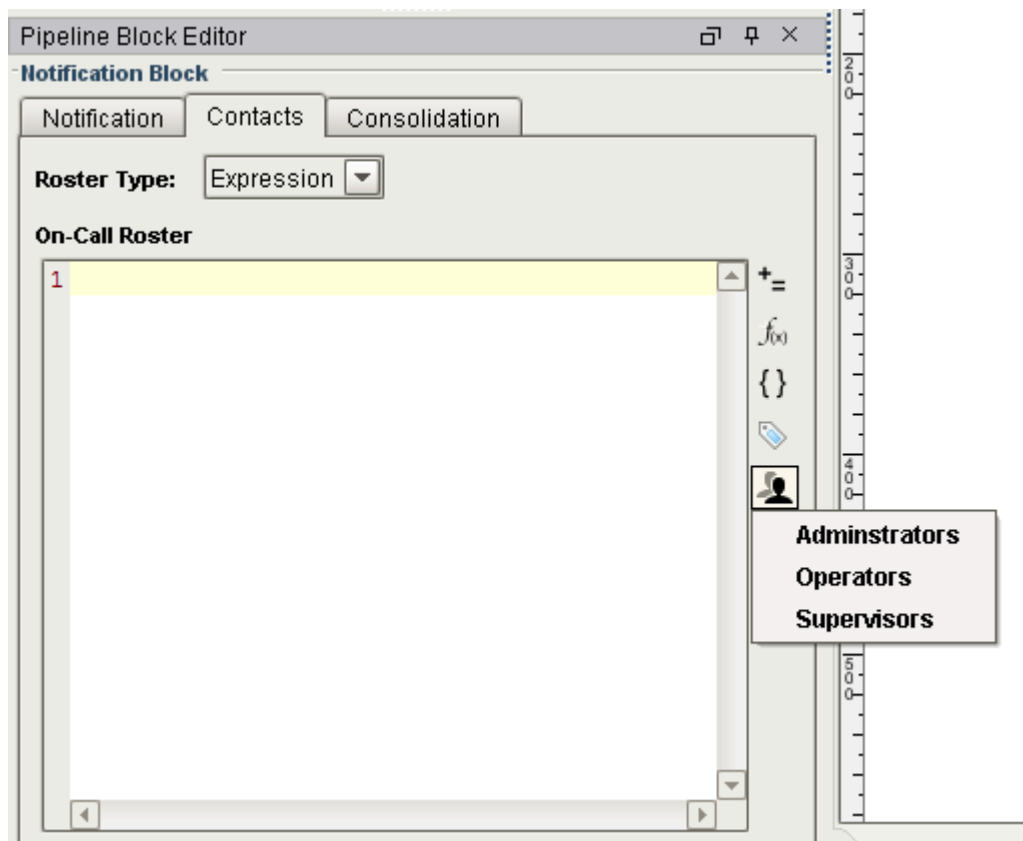
Direct

The Direct roster type is the simplest to set up and use, but also the least flexible. Any On Call Roster defined on the Gateway can be used directly.



Expression

The Expression roster type makes it possible to dynamically choose a single, predefined roster from within the pipeline. The expression can make use of Tag values, alarm properties, and associated data. The alarm properties are accessible from the Alarm Properties (`{}`) button located on the right side of the panel. Associated data can be manually entered using the `{` and `}` characters. A list of rosters is available from the On-Call Rosters () button.



The expression below checks for associated data named "Group" on the alarm. If the value is equal to "A", the Group A roster is used, otherwise the Group B roster will be notified.

```
if({Group}= "A","Group A","Group B")
```

When Should I Use Quotation Marks on the Roster Name?

It is recommended to **always place quotation marks around roster names**. However, it is not always required for notifications to be sent out successfully.

If the expression consists of only the name of the roster, then quotes can be omitted. For example, specifying that a Notification block should always use the roster named **"Operators"** could look like the following with quotes:

```
// This expression will evaluate successfully.  
"Operators"
```

Which would return the string **Operators**, a valid Roster.

In this scenario, quotation marks could be removed. This normally causes a syntax error, because strings in expressions must be encased in quotation marks. However, in this case, the Notification block will assume you simply typed out the name of the roster, so the following example will also successfully use the **Operators** roster:

```
// This will fail to evaluate as an expression, so the block will instead look for  
a roster named "Operators",  
// and the users will still be notified.  
Operators
```

If the expression is any more complicated than the above example, such as using an expression function, then quotation marks must be used. The following example would check the priority of the alarm. If the alarm's property was "critical", then the **High Priority Roster** would be notified. For all other properties, the **Low Priority Roster** would be used.

```
// This expression will evaluate successfully.  
if({priority} > 3, "High Priority Roster", "Low Priority Roster")
```

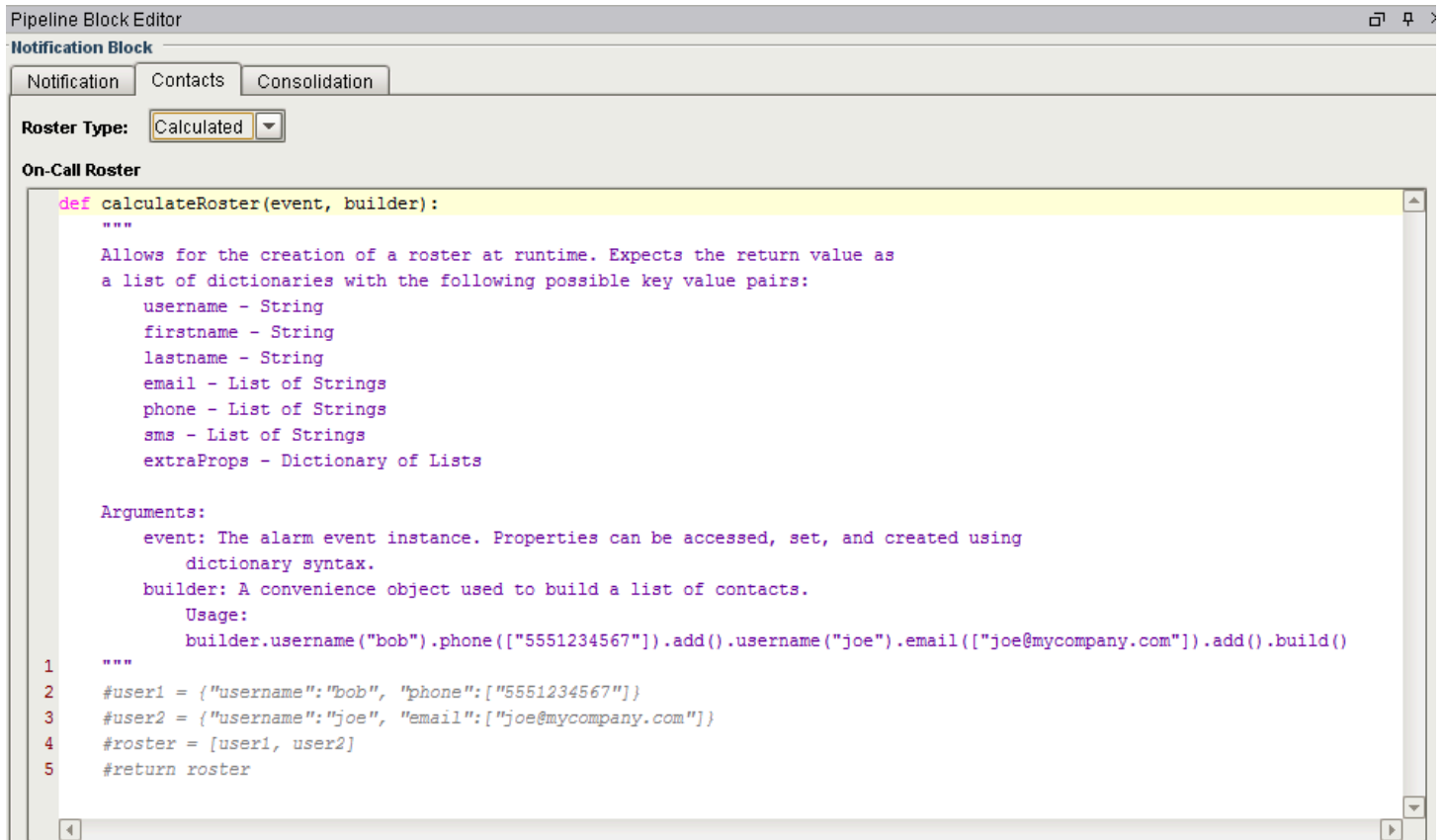
In this scenario, if the quotes were removed from the roster names, the expression would fail with a syntax error:

```
//This expression will fail, so all the text is assumed to be the name of a Roster  
if({priority} > 3, High Priority Roster, Low Priority Roster)
```

When this expression fails, the Notification block will assume that the text is actually the name of a roster, and will attempt to use a roster named `**if({priority} > 3, High Priority Roster, Low Priority Roster)**`. For this reason, it is a good idea to always place quotes around Roster names, as well as check your expressions for syntax errors.

Calculated

The Calculated roster type allows you to use scripting to create a roster at that time using scripting. This becomes extremely useful for situations when you are not sure who should be contacted at the time. Instead of creating an on-call roster for each possible combination of users, you can determine who needs to be notified in the script based on the circumstances of the alarm, and then build the roster to fit that scenario. While this is extremely powerful and offers the most flexibility, it is the most difficult to configure, so you should only use it when absolutely necessary.



The Calculated roster type requires a return value which is a list of dictionaries, where each dictionary is a set of contact information. The contacts that you use are specific to that instance, meaning neither the user nor contact information have to exist on the Gateway. This can either be done manually, by putting together dictionaries with the appropriate contact info and then placing them into a list, or you can use the builder to put together the list for you. Both end up with the same results, so we advise that you use the one that you find easier. The builder has a set of methods which align with the key:value pairs the dictionary is expecting.

Name	Description	Datatype
username	A username for this particular user. Note: this can be anything that you want, the user does not have to exist in the Gateway. This will be used for acknowledgement and auditing purposes, so it is important you create meaning usernames.	String
firstname	A first name for this particular user.	String
lastname	A last name for this particular user.	String
email	A list of possible email addresses for the user.	List of Strings
phone	A list of possible phone numbers for the user for voice notification purposes.	List of Strings
sms	A list of possible SMS numbers for the user for SMS notification purposes.	List of Strings
extraProps	Extra properties that can be added to a user. This is useful when dealing with any third party functionality.	Dictionary of Lists

User Data Created Manually

When putting together a **dictionary** manually, uou can use some or all of the properties listed in the table above. You can create as many dictionaries of users as you want, adding in as much or as little contact info. Once you have your dictionaries created, you then need to place them in a list. You then need to use that list as the return value, which the pipeline will use determine who to contact.

Manual User List

```
# create your user dictionaries
user1 = {"username": "mary", "email": ["mary@company.com"]}
user2 = {"username": "john", "firstname": "John", "lastname": "Smith", "email":
["john@company.com"], "phone": ["1234567890"], "sms": ["8905671234"]}

# create a list of the user data
userList = [user1, user2]

# return the user data
return userList
```

You can use this in conjunction with a database query to dynamically create a list of user information. See below for an example using the Builder.

User Data Created with the Builder

If you instead opt to use the builder, the builder object has a few extra methods that it can use besides what were listed above.

- `add()` - will add all previous info as a single dictionary.
- `build()` - will return the completed list of dictionaries.

The builder object is used by calling consecutive methods on it. Using the same info as the example above, you can add all of the contact info to the builder on a single line.

One Big Builder

```
# Here, we are putting together contact information in our builder using one big string
# Notice that even though it is one long string of code, we are adding both users.
# The first add() at the end of Mary's information acts as a divider between the details
# The second add() at the end of the code will then add all of John's information.
builder.username("mary").email(['mary@company.com']).add().username('john').firstname
```

While the above will work just fine, it may be a bit confusing to put together. However, you don't have to add all of the users at once like that. You can instead break it up, making sure to call `add` at the end of each user's information.

Breaking up the Builder

```
# Here, I still add all of the same contact information to my builder, but this time,  
# The difference here is that I have to call builder again in the second line when adding  
builder.username("mary").email(['mary@company.com']).add()  
builder.username('john').firstname('John').lastname('Smith').email(['john@comapny.com']).add()
```

This can be really useful as it can allow you to add users one at a time, making use of flow control methods like a for loop. You can then use external information, such as database records or Ignition's built in user and security functions to determine who to add.

Looping through Users

```
# This would grab a table from the database that contains a username, email address,  
number.  
# We must specify the database connection name because this is in the gateway scope.  
data = system.db.runQuery("SELECT username, emailaddress, smsnumber FROM user_info_table  
"DB")  
  
# Loop through the returned query data  
for row in data:  
    # add users to the builder row by row  
    # note the extra [] in the email and sms functions  
  
builder.username(row["username"]).email([row["emailaddress"]]).sms([row["smsnumber"]])
```

Once you have all of the appropriate contact information added to the builder, you then call build() on the builder object. This will give us a new built object. We then return our userList as before. Using our example from above:

Complete Builder Example

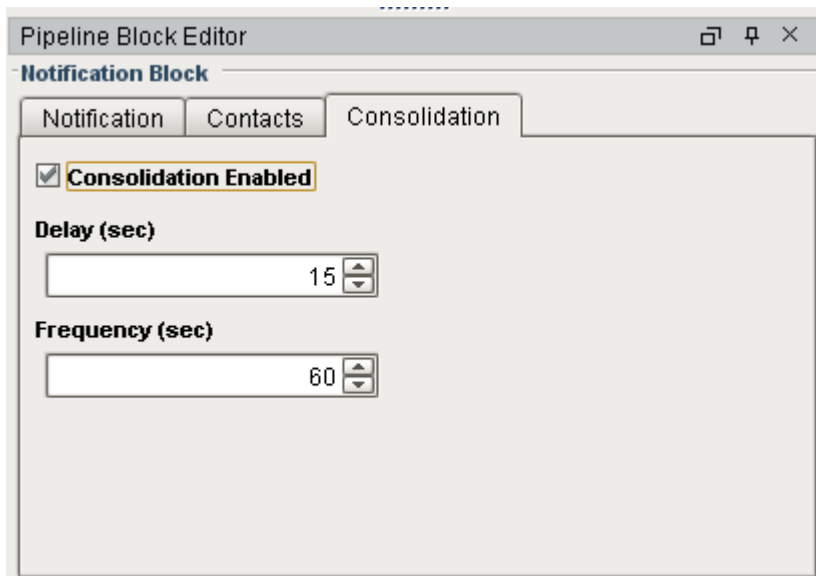
```
# get user information from the database  
data = system.db.runQuery("SELECT username, emailaddress, smsnumber FROM user_info_table  
"DB")  
  
# Loop through the returned query data  
for row in data:  
    # add each user to the builder  
  
builder.username(row["username"]).email([row["emailaddress"]]).sms([row["smsnumber"]])  
  
# use the builder object's build() function to create the finished list of data  
userList = builder.build()  
return userList
```

Consolidation

Notification consolidation allows you to limit the number of notifications sent by specifying a delay during which incoming events will be collected. When the delay expires, all events that have arrived during the period of time will be sent together to the notification profile. The manner in which the profile implements consolidation will vary, but in general the result will be a shorter message, or fewer messages, than would have occurred otherwise.

Consolidation is defined with two parameters:

- **Delay** - How long to wait after the first eligible alarm arrives. This setting forces a pause **before** sending a notification. The delay is used to protect against situations where an event might generate many alarms together. In other words, if an alarm comes into the Notification Block, it will be delayed for this amount of time in case there are other alarms arriving soon.
- **Frequency** - The max frequency with which the profile can send alarms. This setting forces a pause **after** sending a notification. The frequency is used to ensure that contacts aren't notified too often; for example, if an alarm is rapidly going in and out of the active state.



The screenshot shows a window titled "Pipeline Block Editor" with a tabbed interface. The "Notification Block" tab is selected, and within it, the "Consolidation" sub-tab is active. A checkbox labeled "Consolidation Enabled" is checked. Below this, there are two numeric input fields: "Delay (sec)" with a value of 15, and "Frequency (sec)" with a value of 60. Both fields have up and down arrow buttons for adjustment.

Notification Block	Contacts	Consolidation
☒ Consolidation Enabled		
Delay (sec)		
15		
Frequency (sec)		
60		

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Pipeline - Filter on Alarm Priority



INDUCTIVE
UNIVERSITY

**Pipeline - Filter on Alarm
Priority**

[Watch the video](#)

NOTE

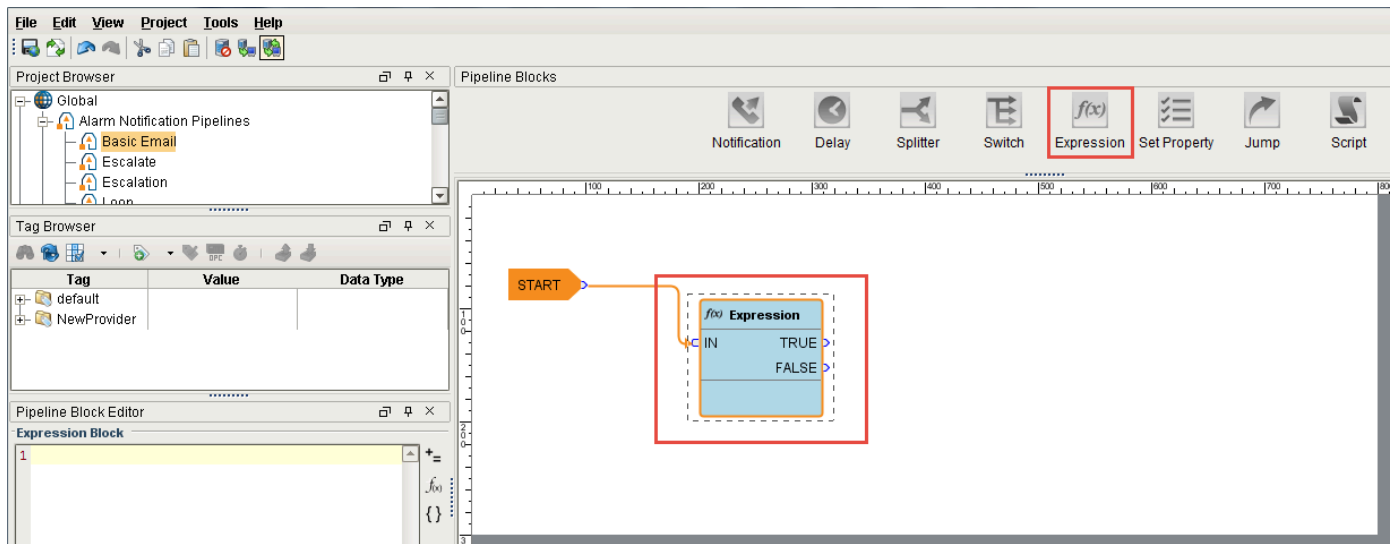
This section assumes that alarms are configured. To learn more, go to [Configuring Alarms](#), [Alarm Notification](#), and [On-call Rosters](#).

How to Filter on Alarm Priority

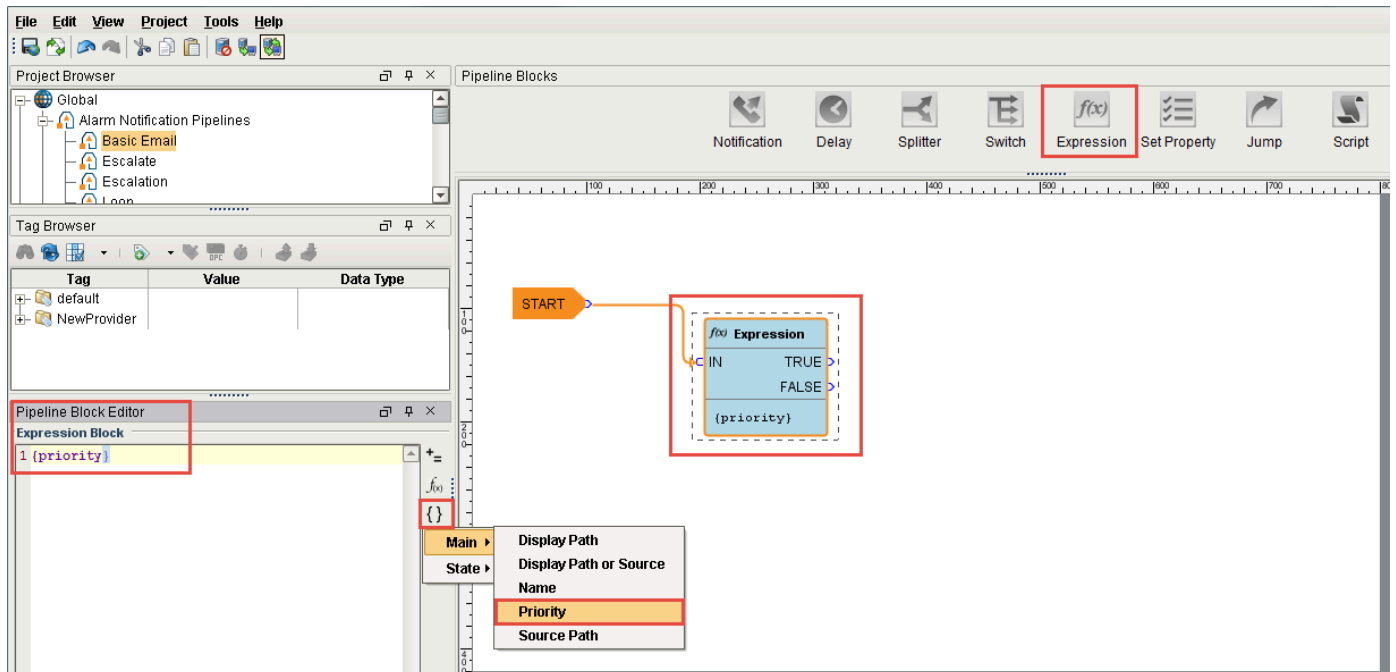
You can create a filter for an alarm priority in an alarm notification pipeline that notifies a different list of users based on the alarm's priority.

1. There are two **pipeline blocks** that can be used for filtering, either the **Switch Block** or the **Expression Block**. The **Switch Block** allows you to switch on the priority, doing something different for every single priority that exists. The **Expression Block** allows you to check for a specific priority, and it returns either a 'True' or 'False' value. You can do something if it is 'True' and something else if it is 'False'. This example uses the Expression Block.

Drag in the **Expression Block** in your pipeline, and connect the **Start** to the **IN** of the Expression Block. You will notice that the Expression Block is going to return a '**True**' or '**False**' value.



2. In this expression, you want to check for the alarm's priority. Go to the Pipeline Property Editor, click the curly brace icon > Main and select the 'Priority' expression. Priority is a placeholder until you enter the priority value you want to filter on. When the alarm is triggered, it will be replaced with the actual priority of that alarm.



3. Alarm priorities have predefined integer values for filtering and sorting. Because this value is an int, you can test against it using `=`, `<`, `>`, `>=`, or `<=`. In this example, we are going to filter for all alarms with a priority of **Critical**. Critical has a value of 4.

i ALARM PRIORITIES

0 = Diagnostic

1 = Low

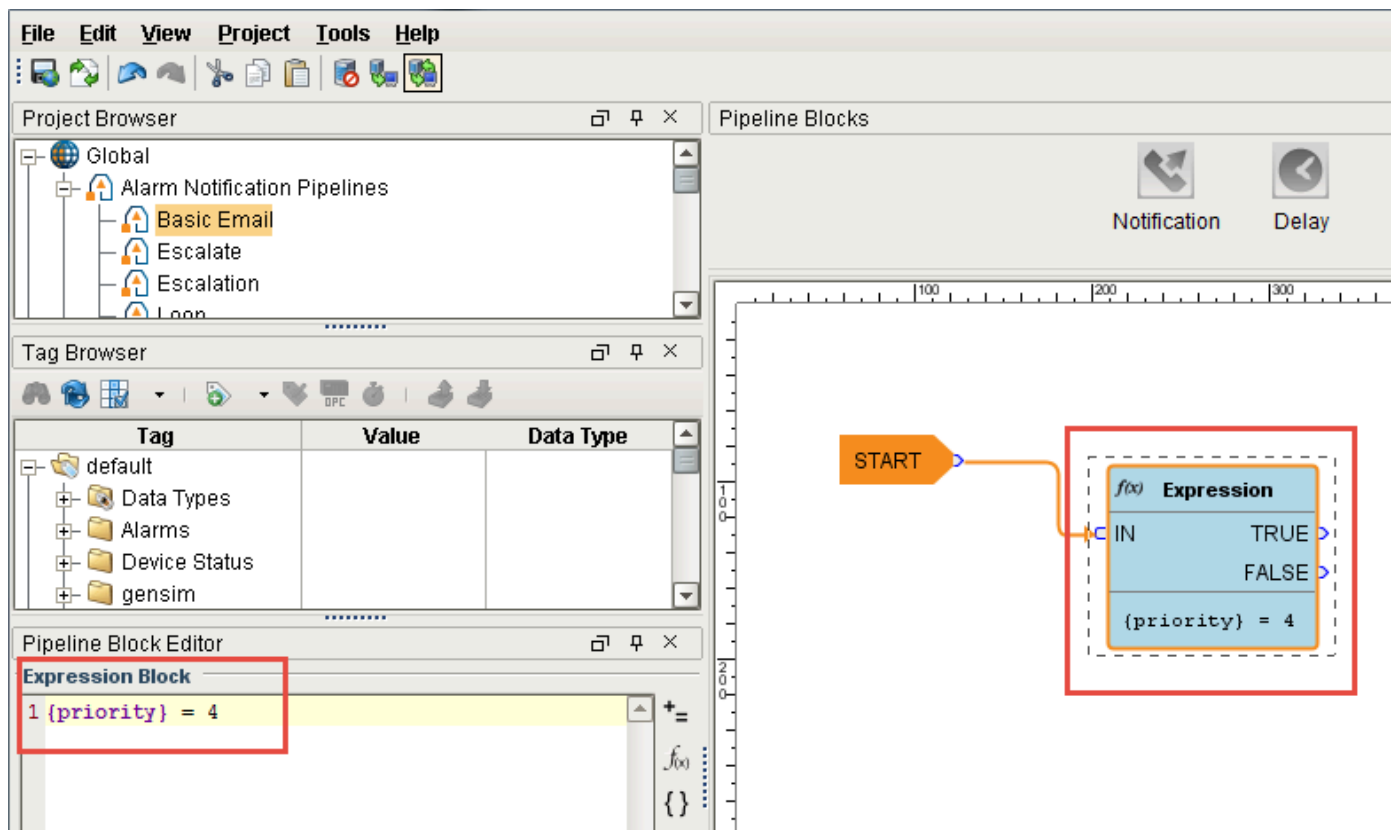
2 = Medium

3 = High

4 = Critical

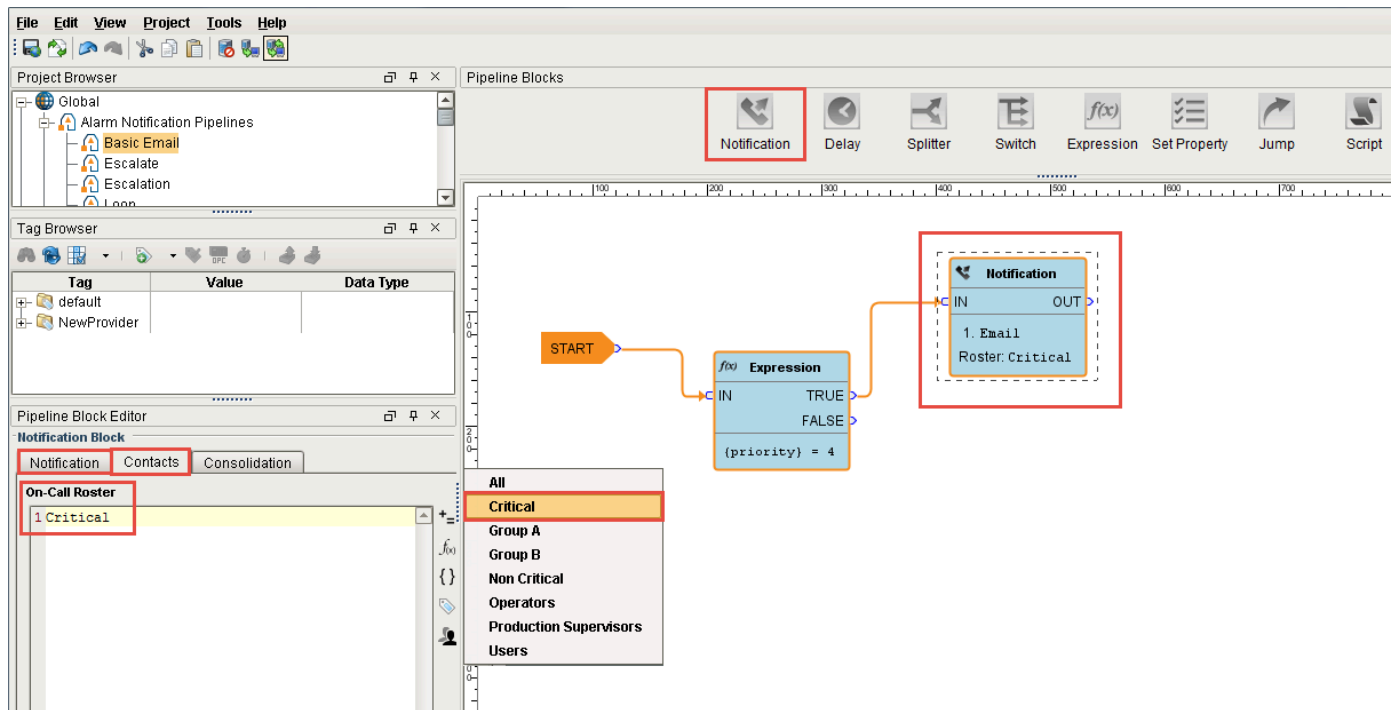
Refer to [Alarm Event Properties Reference](#) for more information.

If the alarm priority is Critical, the alarm will go to the output pin of the 'True' value in the Expression block. If the priority is not Critical, the alarm will go to the output pin of the 'False' value. Notice, we are doing something different for each of these conditions.



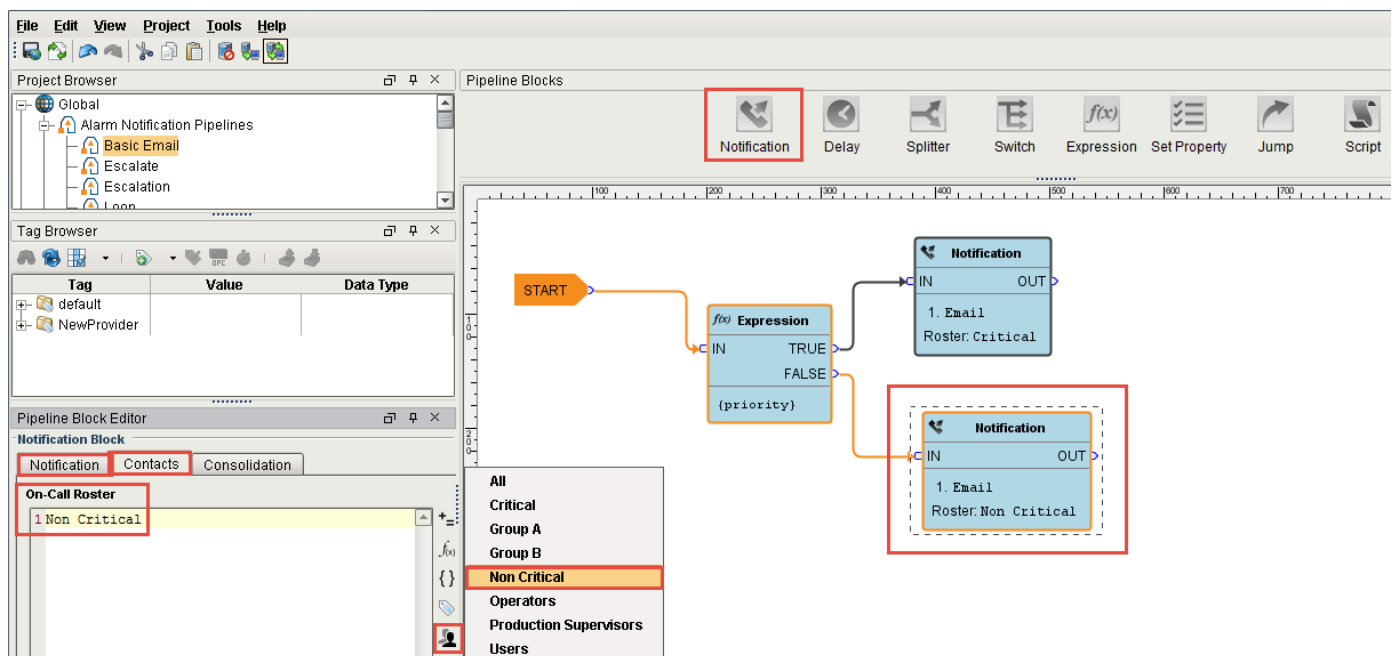
4. Drag in a **Notification Block**, and connect the 'True' output pin into the **IN** of the Notification Block.

In the **Pipeline Block Editor** under the **Notification tab**, choose the notification type. On the **Contacts tab**, under the **On-call Roster icon**, select the '**Critical**' list of people you want to email.



5. Drag in a second **Notification Block**, and connect the 'False' output into the **IN** of the Notification Block..

In the **Pipeline Block Editor**, under the **Notifications** tab, choose the notification type. In the **Contacts** tab, under the **On-call Roster** icon, select the 'Non Critical' list. The Critical list is a different list of people from the Non Critical list.



As you see in this example, the alarm notification pipeline notified different lists of people based on the alarm's priority.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Pipeline - Filter on Alarm Associated Data



INDUCTIVE
UNIVERSITY

**Pipeline - Filter on Alarm
Associated Data**

[Watch the video](#)

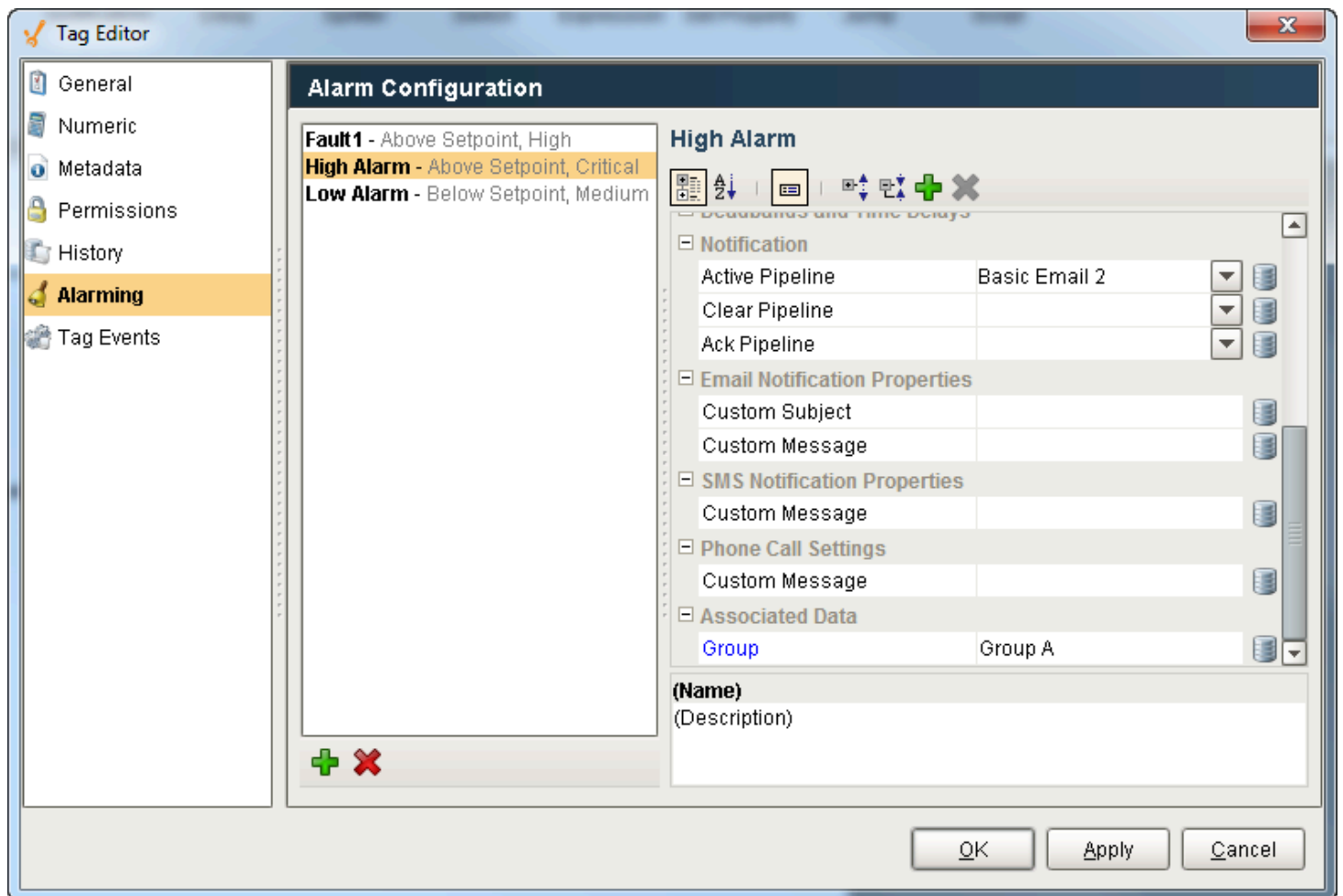
NOTE

This section assumes that alarms are configured. To learn more, go to [Configuring Alarms](#), [Alarm Notification](#), and [On-call Rosters](#).

How to Filter on Alarm Associated Data

Once associated data is added to an alarm, you can easily filter for associated data in an alarm pipeline. It's possible to add associated data that can be either static or dynamic. Below is an example of how to filter a notification pipeline based on associated data added to an alarm.

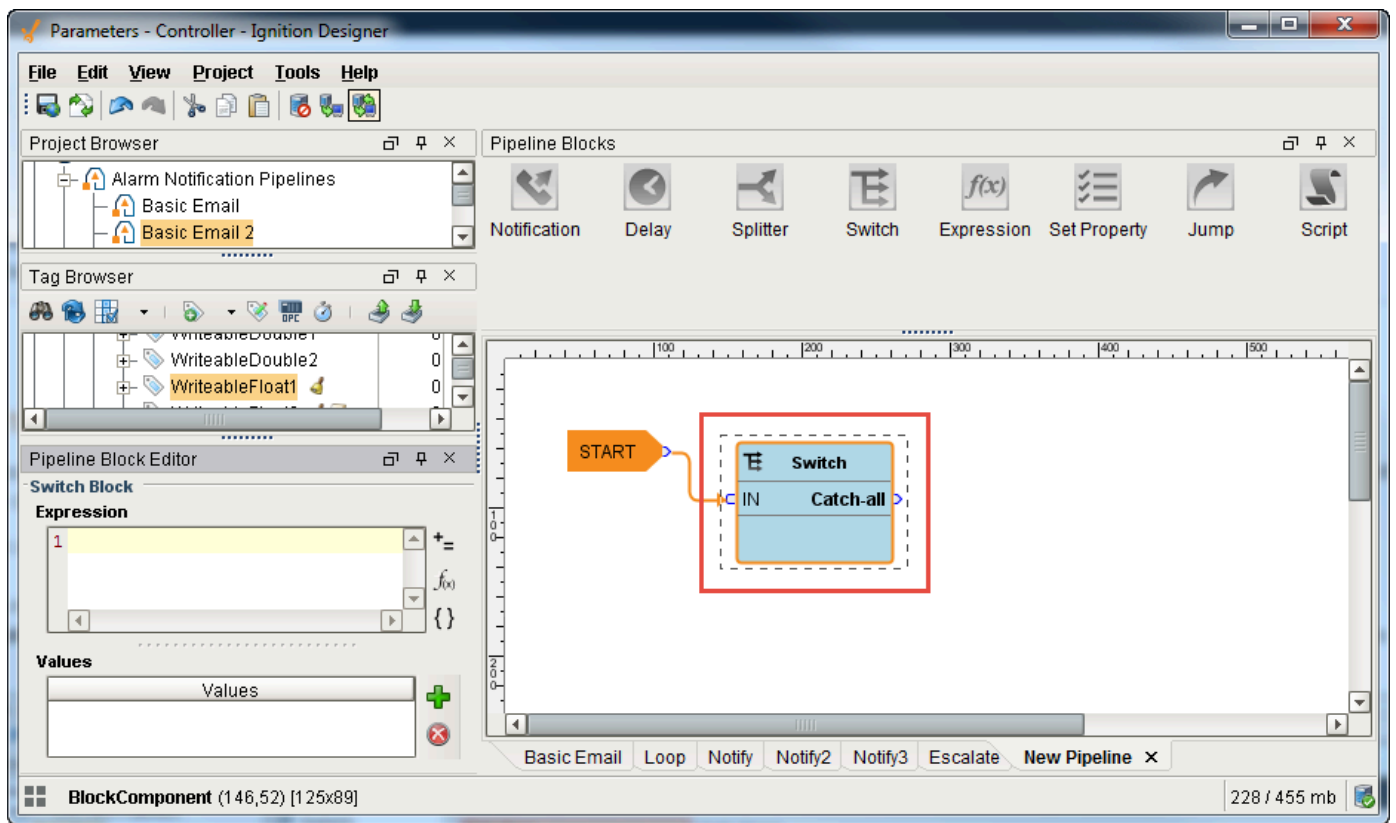
A WriteableFloat1 Tag is used for this example. Go to the **Tag Browser**, double click on **WriteableFloat1**, the Tag Editor will open, and click **Alarming**. You'll notice this example has three alarms setup. You can see the **High Alarm** configuration has **Associated Data** called '**Group**', and a value of either Group A or Group B. We are going to use this associated data to filter the alarm notification pipeline.



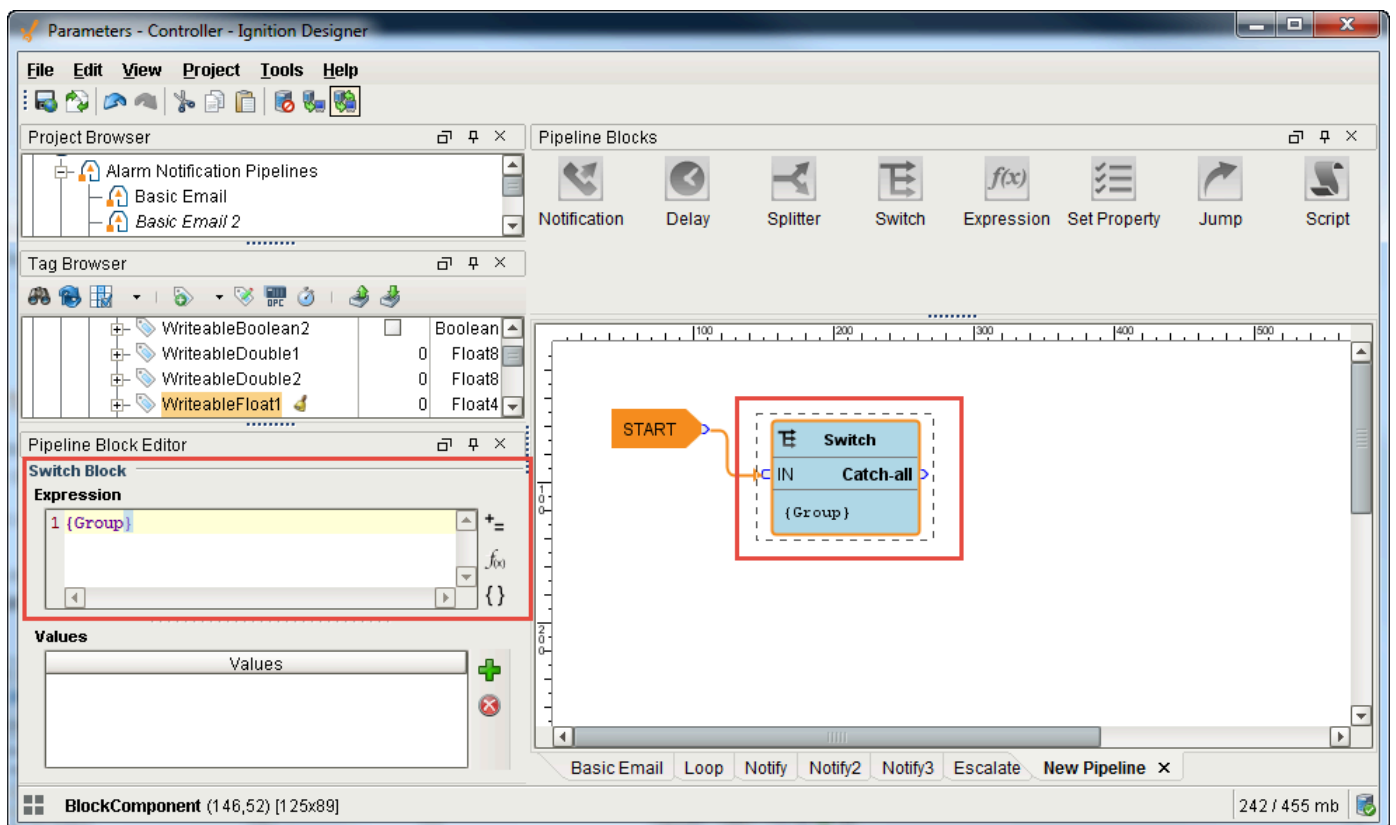
Let's create a pipeline that filters on associated data.

1. There are two blocks that can be used for filtering, either the **Switch Block** or the **Expression Block**. The **Switch Block** allows you to switch on a value and do something different for Group A and Group B. The **Expression Block** does the same thing, but only returns values of 'True' or 'False.'

For this example, drag in the **Switch Block** to your pipeline, and connect the **Start Block** to the **IN** of the Switch block.

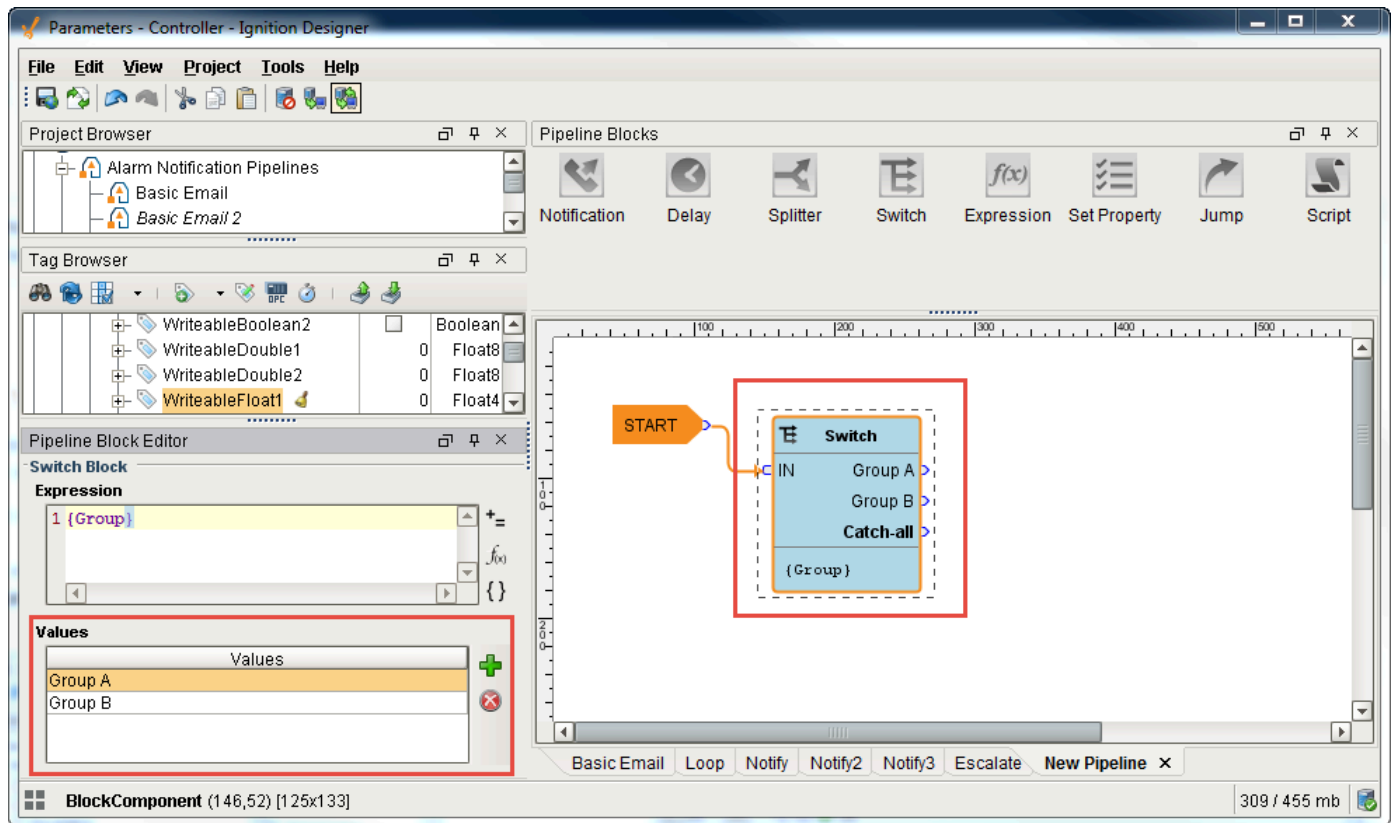


2. In the **Pipeline Property Editor**, you have to manually enter the expression because you will not find any of the associated data properties in the dropdown list. Enter the name of the associated data property in curly braces as follows, `{Group}`. The '**Group**' associated data property acts as a placeholder for the real values.



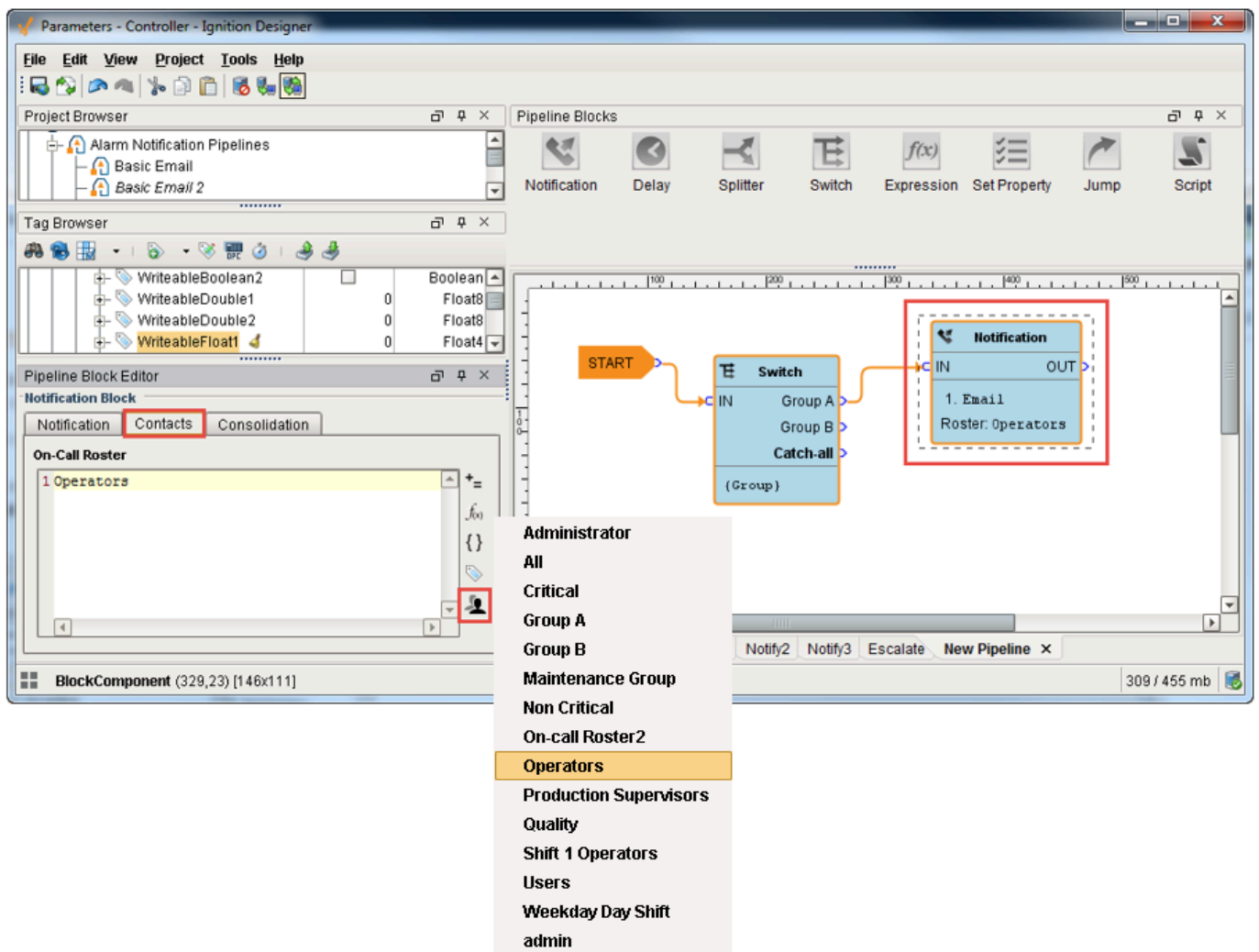
3. Next, you need to add some values to the expression. You can add as many values that you have values for. Click on the **green plus icon** and add two values: '**Group A**' and '**Group B**'.

Once you enter both values, you will have outputs in the Switch block for **Group A**, **Group B**, and by default, **Catch-all**. Now you can notify a different list of people based on each Group.

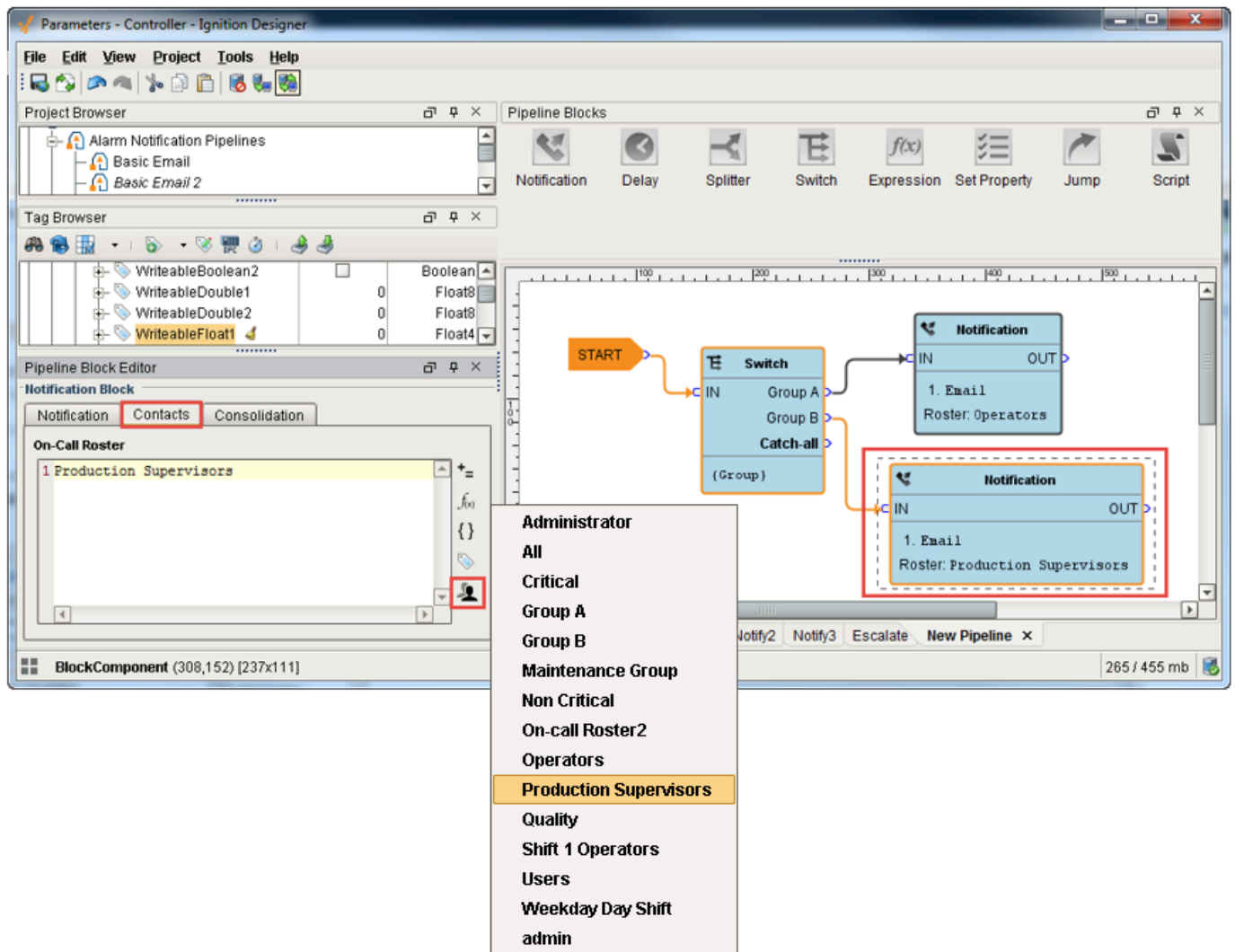


4. Drag a **Notification Block** into your pipeline, and connect the output of **Group A** in the Switch block to the **IN** of the Notification block.

In the **Pipeline Block Editor**, under the **Notification** tab, choose **Email** as your notification type. In the **Contacts** tab, under the **On-call Roster icon**, select the roster you want to notify (i.e. Operators).



5. Drag a second **Notification Block** to your pipeline, and connect **Group B** from the Switch block to the **IN** on the second Notification block.
Choose **Email** as your notification type, and specify a different **On-call Roster** to notify a different group of people (i.e. Production Supervisors).



Now, you are notifying people based on Group A (Operators) and Group B (Production Supervisors) by filtering on alarm associated data.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Pipeline - Simple Loop

NOTE

This section assumes that alarms are configured. To learn more, go to [Configuring Alarms](#), [Alarm Notification](#), and [On-call Rosters](#).

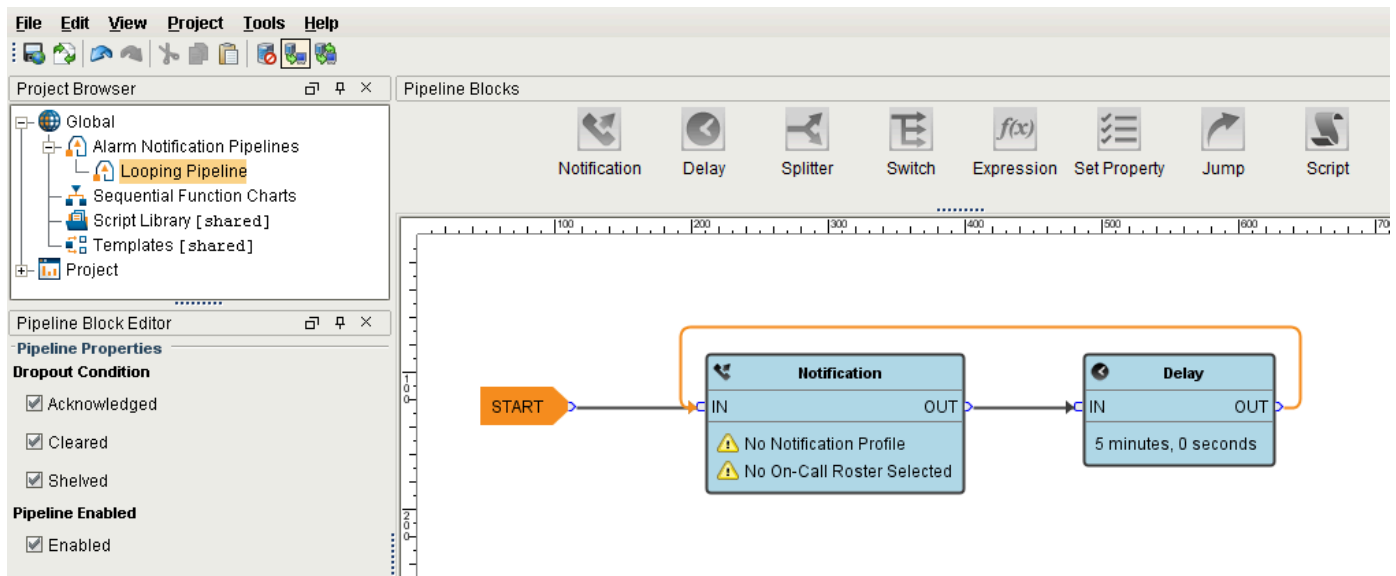
How to Create a Looping Alarm Notification

You can create a loop in an alarm pipeline to notify users continuously until the alarm is dealt with. It is important to pay close attention to your dropout conditions when you are creating a loop. If you remove all dropout conditions the loop will continue until the alarm pipeline is manually stopped from the Gateway Webpage.

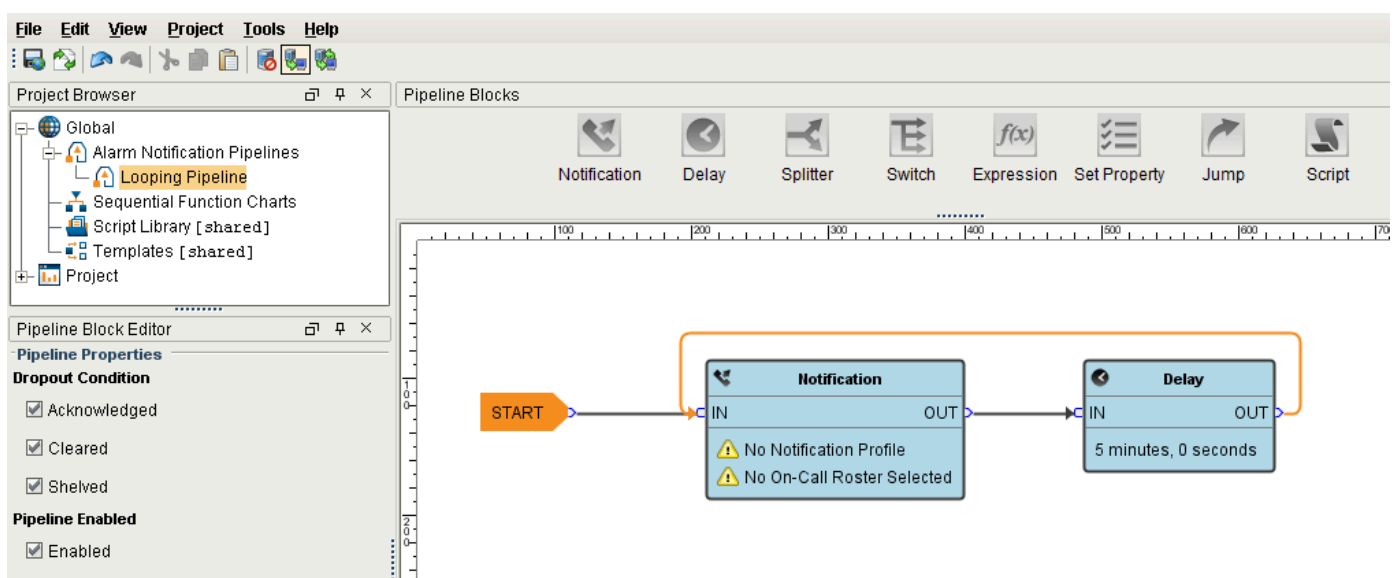
There are two **pipeline blocks** that we need to create a loop. The **Notification Block** to send a notification, and the **Delay Block** to create a pause between each notification.

In this example, we will use the default settings for the Dropout Conditions. All three options (Acknowledged, Cleared, and Shelved) will remain checked.

1. Drag in the **Notification Block** in to your pipeline, and connect the **Start** to the **IN** handle of the Notification block. You will need to set up your Notification block, but you can see more about that on the Notification Block page. This will cause the pipeline to notify immediately then move on.

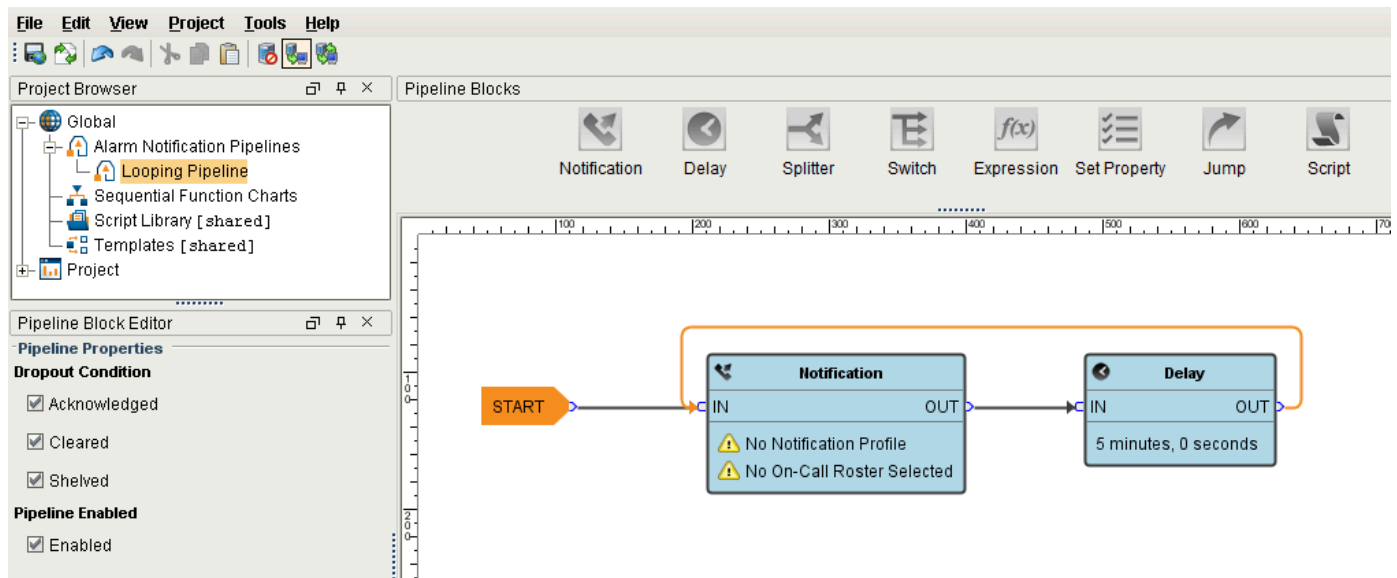


2. Drag a Delay block onto the pipeline and set the **Delay (sec)** property. We will chose 300 seconds (5 minutes).



3. Connect the two blocks by dragging from the **OUT** handle of the Notification block to the Delay block.

Now create a second connection by dragging from the **OUT** handle of the Delay block to the Notification block.



That's it! Now you have a pipeline that will notify immediately, wait 5 minutes, then notify again, and repeat until one of the dropout conditions is met.

You can use this technique with escalation to notify five times then change to another pipeline.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Pipeline - Escalation



INDUCTIVE
UNIVERSITY

Pipeline - Escalation

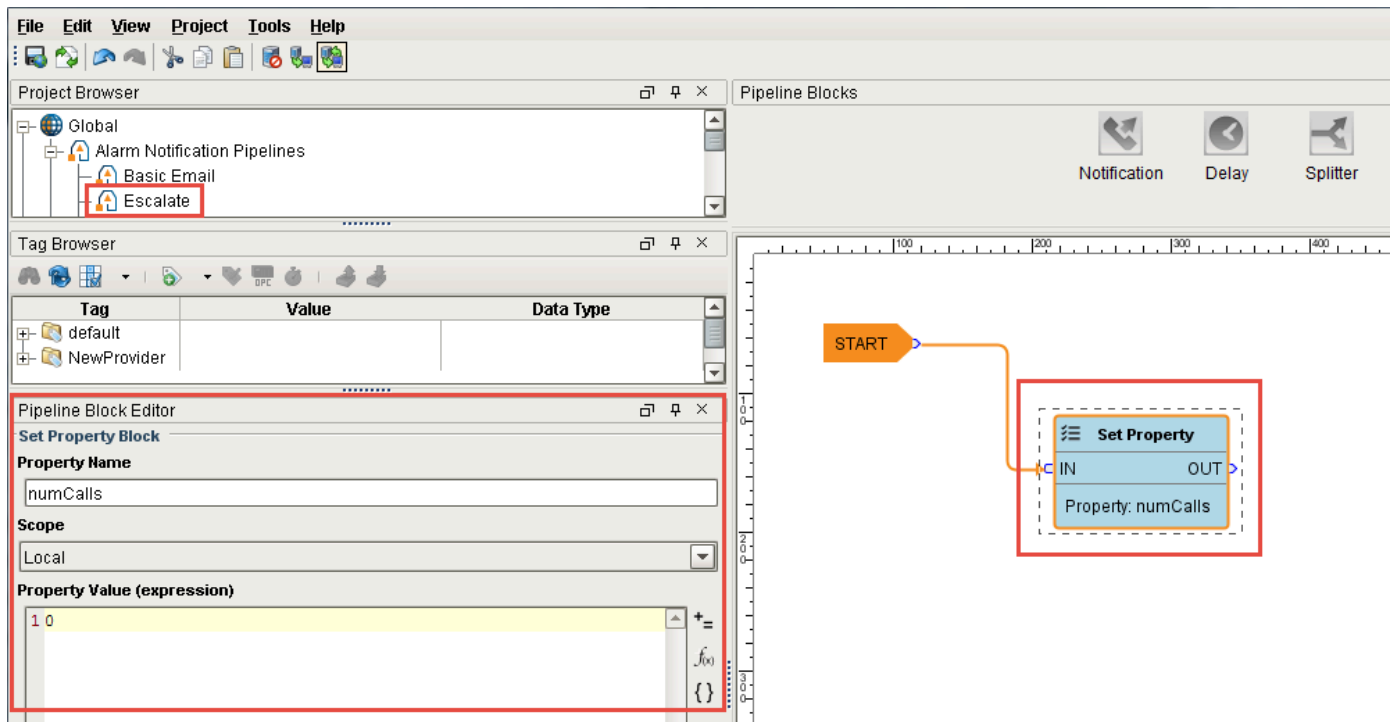
[Watch the video](#)

How to Setup Escalation Procedures

It is possible to setup escalation procedures in [alarm notification pipelines](#) in Ignition. The following example will create a pipeline that notifies operators three times. If a user does not acknowledge the alarm after the third time, the alarm escalates to the Production Supervisor's on-call roster.

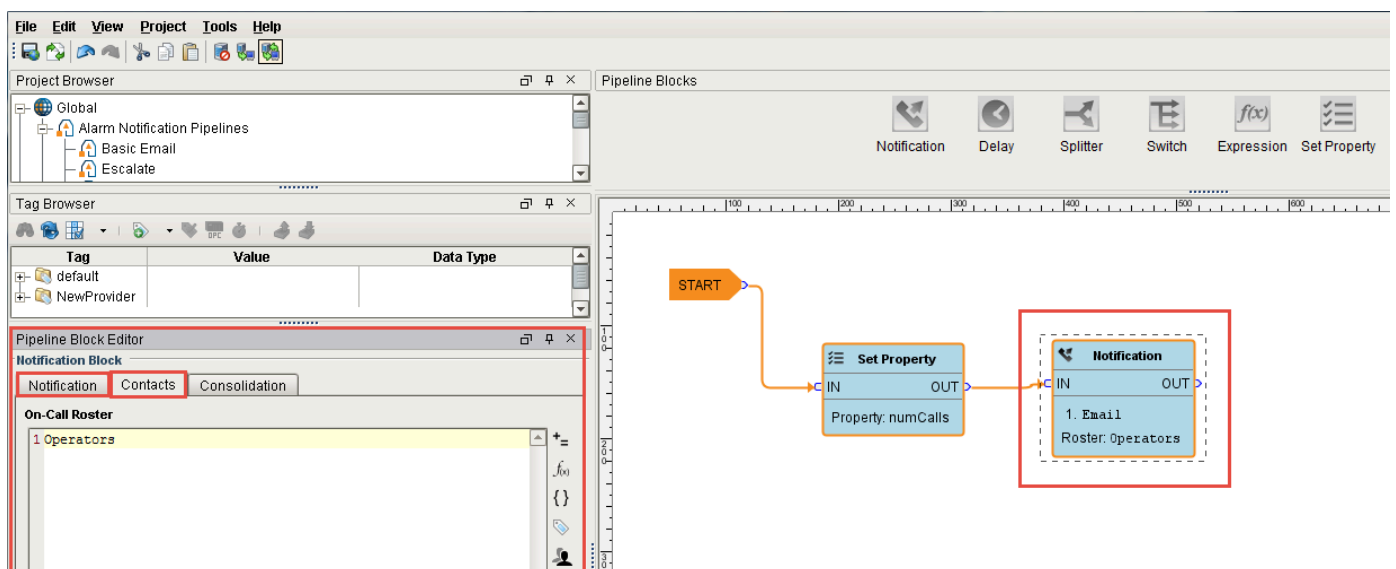
The following example requires that two separate [on-call rosters](#) have been configured ("Production Supervisors" and "Operators"), as well as an [email notification profile](#).

1. In the **Project Browser** under the **Alarm Notification Pipelines**, create a New Alarm Pipeline called '**Escalate**'.
2. In order to keep track of how many times the operators get notified, create a counter using the [Set Property block](#) inside your pipeline. Drag the Set Property Block to the Pipeline Workspace and connect the **START** block to the IN of the Set Property block. In the Pipeline Block Editor, set the **Property Name** to '**numCalls**' and initialize the **Property Value** to '**0**'.



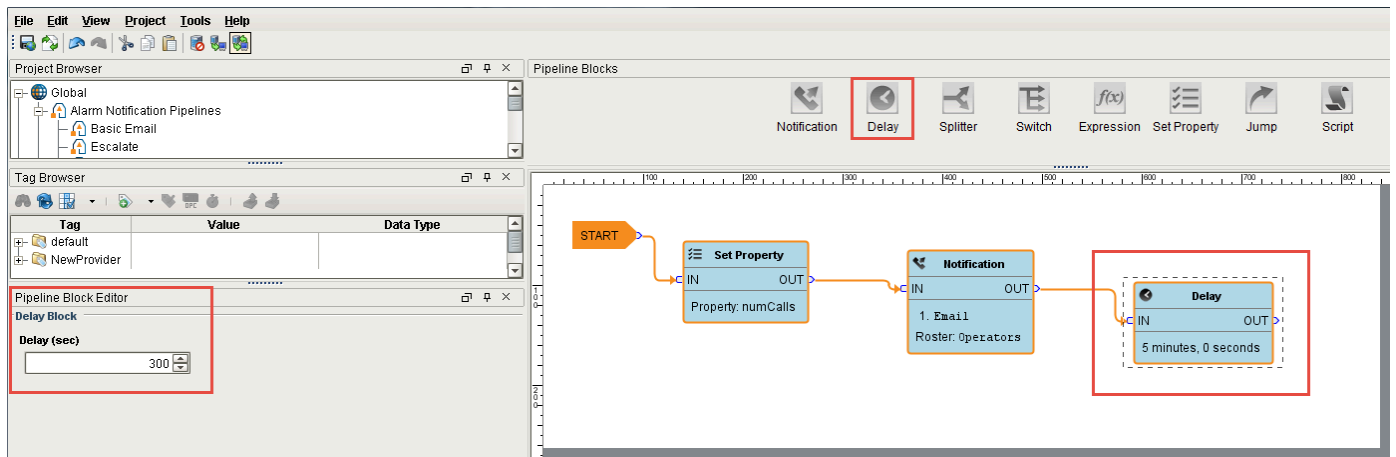
3. Drag the **Notification Block** to the Pipeline Workspace and connect the OUT from the Set Property block to the IN of the Notification block.

In the **Pipeline Block Editor**, set the Notification to use 'Email' and set the Contacts to use the 'Operators' on-call roster.



4. A Delay Block will be added so the operators have some time to respond to the alarm before another notification is immediately sent out.

Drag in the **Delay Block** and connect the **OUT** from the Notification block to the **IN** of the Delay block. In the Pipeline Block Editor, set the Delay property for '**300**' seconds. It will give the operators 5 minutes to respond to the alarm before going to the next block.



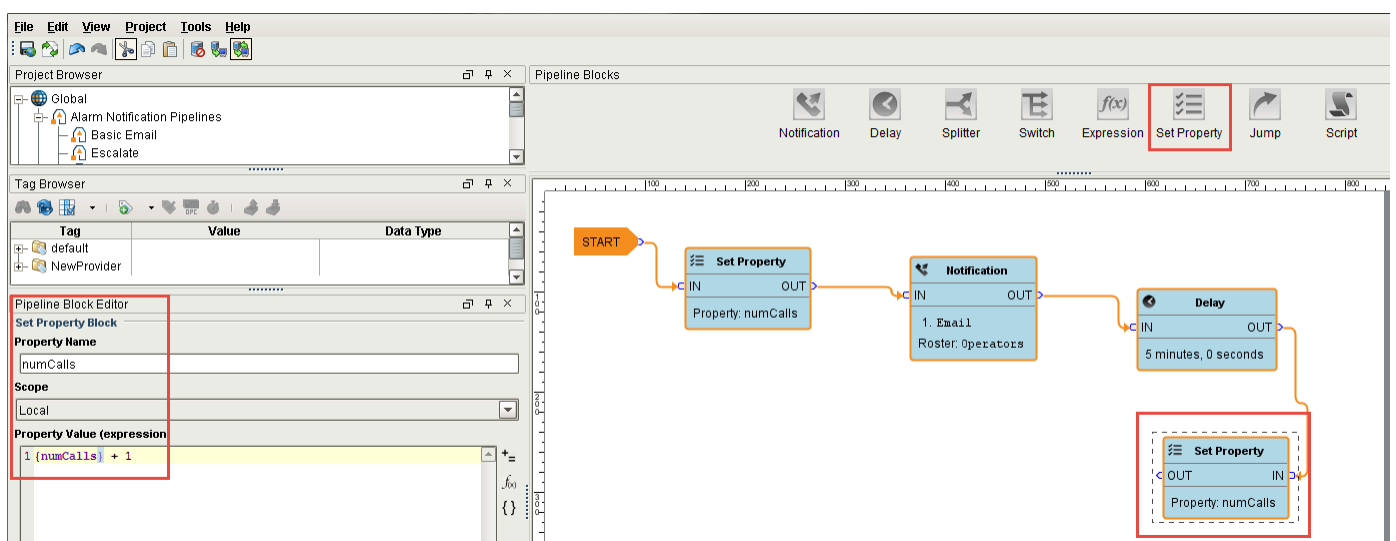
5. If the alarm is still active and not acknowledged, the counter will increment by 1. In order to increment the counter, drag in another **Set Property Block** and connect the **OUT** of the Notification block to the **IN** of the Set Property block. You can right click on the Set Property block to **reverse the orientation**.

NOTE

A Note on Syntax

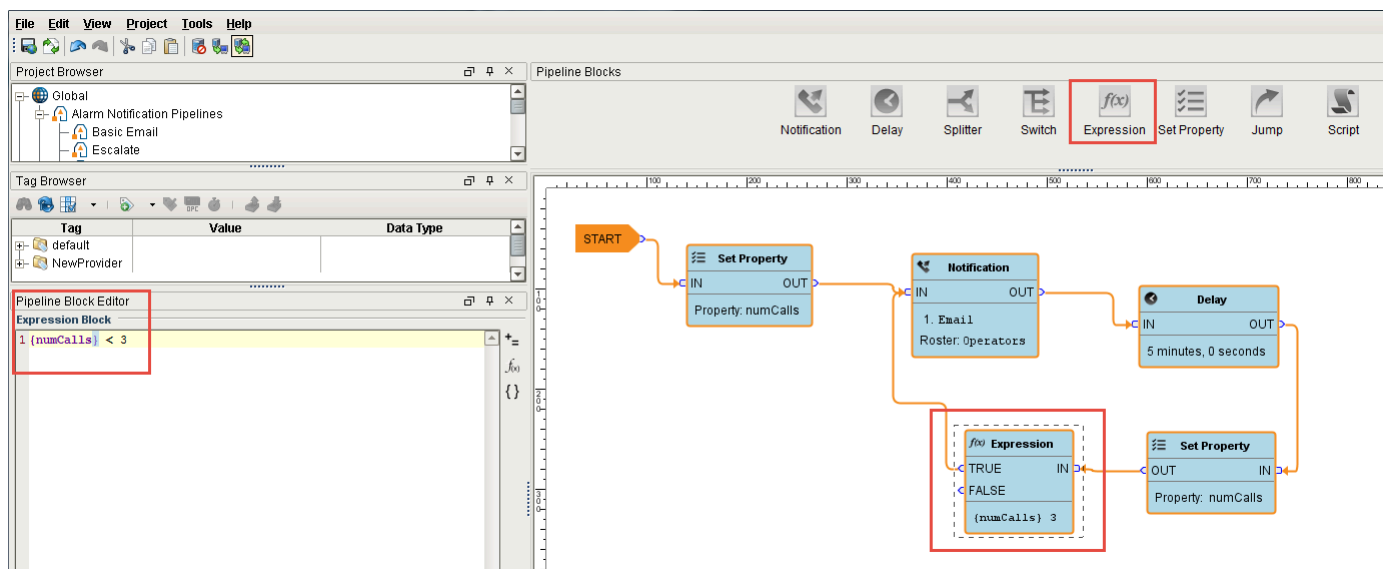
The Set Property Block's **Property Value** uses Ignition's Expression language. More information on the Expression language can be found on the [Expression Overview and Syntax](#) page.

In the Pipeline Block Editor, set the **Property Name** to 'numCalls'. In the **Property Value Expression** type in, `{numCalls} + 1`. In this expression, use the curly braces to refer to that variable so the counter gets incremented by 1.



6. The **Expression Block** will check to see if "numCalls" is < 3'. Drag in an Expression Block to your pipeline workspace and right click on the Expression block to reverse the orientation. Connect the **OUT** of the Set Property block to the **IN** of the Expression block.

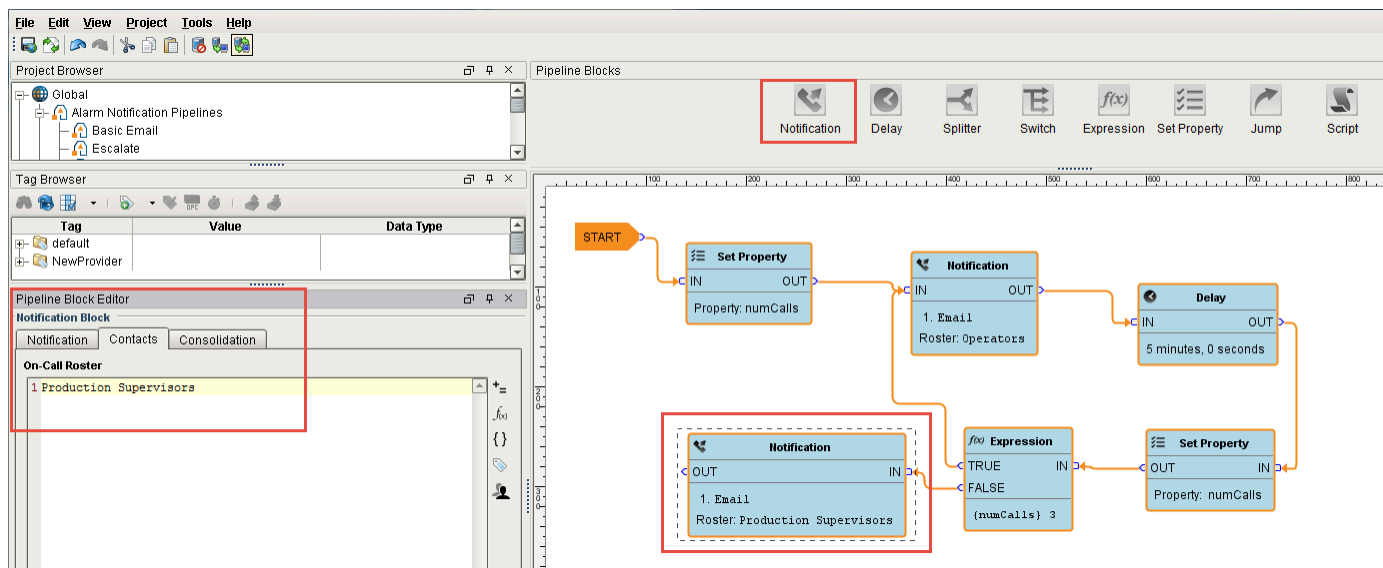
Since the value of 'numCalls' is less than 3, the alarm event will follow the 'True' path. Connect the 'True' output to the **Operator Notification block**.



7. If the '**numCalls is > 3**', the alarm event will follow the 'False' path and it will escalate to the Production Supervisors.

Drag in another **Notification Block** and reverse the block orientation. In the Pipeline Block Editor, set the Notification to use '**Email**' and set the Contacts to use the '**Production Supervisors**' on-call roster.

Connect the '**False**' pin to the **IN** of the **Production Supervisors Notification block**.



As you can see, when the Escalate Pipeline sets the 'numCalls' variable to a value of 0, it's going to notify the Operators on-call roster through email that the alarm is active. It will wait 5 minutes, and if the alarm is still active, it will increment the counter by 1. This will repeat until the 'numCalls' is > 3. After the 3rd time, if the alarm is still active and in the pipeline, then the Production Supervisors will be notified. Keep in mind, that if the alarm at any point meets the dropout conditions, it will not go to the next block of the pipeline.

**TIP**

You can also use the [Jump Block](#) as your escalation instead of a second Notification Block. This will allow you to jump into another complete pipeline when nobody answers.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Pipeline - Notification Block Consolidation



INDUCTIVE
UNIVERSITY

Pipeline - Notification Block Consolidation

[Watch the video](#)

How Consolidation Works

Notification consolidation allows you to limit the number of notifications sent by specifying a delay during which incoming events will be collected. When the delay expires, all events that have arrived during the period of time will be sent together to the notification profile.

A screenshot of the 'Pipeline Block Editor' window. The 'Notification Block' is selected, and the 'Consolidation' tab is active. The 'Consolidation Enabled' checkbox is checked. The 'Delay (sec)' is set to 15, and the 'Frequency (sec)' is set to 60. The 'Notification' and 'Contacts' tabs are also visible.

Pipeline Block Editor

Notification Block

Notification Contacts Consolidation

☒ Consolidation Enabled

Delay (sec)

15

Frequency (sec)

60

Example

Once an alarm enters the Notification block in this pipeline, it will wait for **Delay** number of seconds (in this case 15 seconds) before sending the alarm. If additional alarms enter this notification block, it will

send using the **Consolidated** Message/Subject from the **Notification** tab depending on your notification type.

Once an alarm has been sent, whether it was a consolidated notification or not, this Notification Block will wait for **Frequency** number of seconds (in this case 60 seconds) before it can send another email.

Note, that Consolidation only effects this pipeline. Any notification blocks in other pipelines will not be effected.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Pipeline Status



INDUCTIVE
UNIVERSITY

Pipeline Status

[Watch the video](#)

Monitoring the Pipeline Status

It's possible to monitor the status of your alarm notification pipelines to see exactly where your alarms are in the notification process in Ignition. You can do this from the **Ignition Gateway Webpage** in your browser.

1. Go to the **Status** tab of your Gateway webpage and on the left side of the navigation bar, click on **Systems > Alarm Pipelines**. The Alarm Pipeline page will open and display the status of all your Alarm Pipelines. This example shows five pipelines with one active pipeline for Basic Email and four inactive pipelines.

Controller

USER MANUAL
SUPPORT

admin
Sign Out

Launch Designer

HOME
STATUS
CONFIGURE

Search...

SYSTEMS

Overview
Performance
Alarm Pipelines
Gateway Scripts
Modules
Redundancy
Reports
SFCs
Tags
EAM Tasks
Transaction Groups

CONNECTIONS

EAM Agents
Databases
Designers
Devices
Gateway Network
Store & Forward
OPC Connections
OPC-UA Server
Vision Clients

DIAGNOSTICS

License Incomplete ?
0:00:00
Reset Trial
View Modules

Connections
Alarm Pipelines

Active
1/5

Name
Active?
Items

Basic Email
true
2
Details

Escalate
false
0
Details

Escalation
false
0
Details

Loop
false
0
Details

Notify
false
0
Details

- To see a detailed status of any of your alarm pipelines, click the **Details** tab for any of your alarms. This example shows that the Basic Email pipeline has two active alarms. The Details shows you the affected Source Path, Display Path, Pipeline Block, Status and the length of Time stuck in a pipeline block. You can **Cancel** the alarm to delete it from the pipeline, and view the logs by clicking on the **Pipeline Logs** tab.

Connections

Configuration

Alarm Pipelines / Basic Email

Pipeline Status
Pipeline Logs

Pipeline Status

Source ▲
Display Path
Block
Status
Time In Block

prov:default/tag:gensim/Writeable/WriteableFloat1/alm:Fault1
Machine A
NotificationBlock
Notifying user 'admin' (1 of 1).
25 minutes
Cancel

prov:default/tag:gensim/Writeable/WriteableFloat1/alm:High Alarm
Machine A
NotificationBlock
Notifying user 'admin' (1 of 1).
25 minutes
Cancel

How long an alarm stays in the in the Alarm Pipeline Status depends on how the pipeline is configured, the type of notification (i.e., Email, SMS or Voice), and what the pipeline is doing (i.e., notifying users, looping, caught in a delay, executing a script, etc.,).

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Customizing Alarm Notification

Personalizing Notifications

There are several modifications that can be made to notifications to fit specific needs or interests.

Notification Messages

When notifications are delivered, default messages are used. These messages state the time of the event, the name of the alarm, the Display Path, and the state of the alarm. However, it is possible to customize these messages. Each message can return the value for any configured alarm property, in any order desired.

Information on customizing notification messages can be found on the [Custom Email, SMS, and Voice Scripts](#) page.

Using a PIN

In the case of SMS and Voice, additional security may be desired as phones can easily be misplaced. It is possible the phone may fall into the wrong hands. To address this, SMS and Voice notifications may make use of a PIN. When enabled, the PIN must be entered before users will be able to acknowledge alarms.

Examples and more details can be found on the [User PIN for Voice](#) page.

Multi-Language Support

Voice notifications may be delivered in several different languages, and multiple languages may be supported on the same Gateway. Preferred languages are assigned to each user. When a notification block is invoked, the roster information contains the preferred language of each user, and the correct voice language is used.

More information can be found on the [Voice Languages](#) page.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Custom Email, SMS, and Voice Scripts



INDUCTIVE
UNIVERSITY

**Custom Email, SMS, and Voice
Scripts**

[Watch the video](#)

Introduction

Ignition has a built-in script that it uses when it sends out an alarm notification via Email, SMS, and Voice. There is a different script for each notification type. Each notification type has its own relevant properties for configuration, although, some features are common to all profile types that you will soon discover.

Alarm Notification Setup Requirements

Email, SMS and Voice scripts in Ignition are extremely easy to use and setup. Before you begin using alarm notifications and customizing message scripts, you need to have Email, SMS and Voice modules installed, alarm notification profiles created, and settings configured. It's best to run through these alarm notification requirements before setting up your custom scripts so you have a smooth implementation of alarm notifications. You only need to install the modules and the profiles that you plan to use.

Alarm Notification Modules

The following modules can be found on the [Ignition Downloads](#) page.

- Email Notification Module
- SMS Notification Module

- Voice Notification Module
- Text-to-Speech Voice (language)

Alarm Notification Profiles

Refer to each alarm notification profile below for how to create and modify a profile.

- **Email Notification Profile - Two-Way Acknowledgement** is enabled here if you want users who receive the alarm to acknowledge the alarm.
- **SMS Notification Profile - Two-Way Acknowledgement** is enabled here.
- **Voice Notification Profile** - Acknowledgement of the alarm is part of the call script, and provided as in instruction as part of the call.
- **User Profile** - When setting up a Voice alarm notification, each user's profile requires a contact telephone number and a Security PIN.

Alarm Notification Pipelines

- **Alarm Notification Pipelines** - Controls how alarm notifications are sent out.

Notification Roster

- **On-Call Rosters** - The group of users that will receive the alarm notifications. On-Call Rosters are available in the Contacts tab of the Notification Block.

Methods of Customizing Message Scripts and Sending Notifications

You can configure and customize scripts for Email, SMS, and Voice in two locations:

- The Alarm Configuration section of the Tag Editor
- The **Notification Block** of the alarm pipeline.

Things to Know About the Pipeline Notification Block

Every Notification Block in a pipeline has to have a message to send out. Here, you have all the alarm properties to select from to build and test your script. You can specify your Contacts, and enable Consolidation all from the same screen.

Things to Know About Using the alarm Configuration in the Tag Editor

The Alarm Configuration has fields to override the message sent from a Pipeline's Notification Block. This is particularly useful if your alarm has a dynamic Pipeline or is being sent to multiple Pipelines. If

you're creating your alarm notification for the first time, make sure to specify a Pipeline for the alarm to use.



TIP

The alarm's Custom Message property takes precedence over any Notification Block in a pipeline.

Methods of Sending Notifications

There are two methods of sending out alarm notifications:

- One notification per alarm
- Notification consolidation (i.e. multiple alarms in a single Email or SMS)

By default, alarm notifications are sent out one message for each alarm.

If you want to use **Consolidation**, you can use the default Consolidated Message script or create a custom script using the alarm properties. Consolidation is enabled in the Notification Block. When enabled, multiple alarm events are sent together in one message.

Default Message Scripts

Each alarm notification type has a default message script. The Email and SMS default message script is found in the Notification Block shown below. For Voice, the default call script is located in the **Voice Notification Profile**.

Default Message Script

```
At {eventTime|hh:mm:ss}, alarm "{name}" at  
"{displayPath}" transitioned to {eventState}
```

You can override the default message script on a per alarm basis by simply creating a Custom Message script.

CUSTOM MESSAGE ALWAYS TAKES PRECEDENCE OVER THE DEFAULT MESSAGE

All alarm notification types allow you to define a custom message. If a custom message is created, it will be used instead of the default message in the script.

Custom Message Scripts

The following sections describe how to use the default message scripts, build your own custom message scripts, and send out alarm notifications for Email, SMS, and Voice.

Email

Email notifications are the most common method of sending alarm notifications.

Customizing Email Message Scripts using Alarm Configuration in the Tag Editor

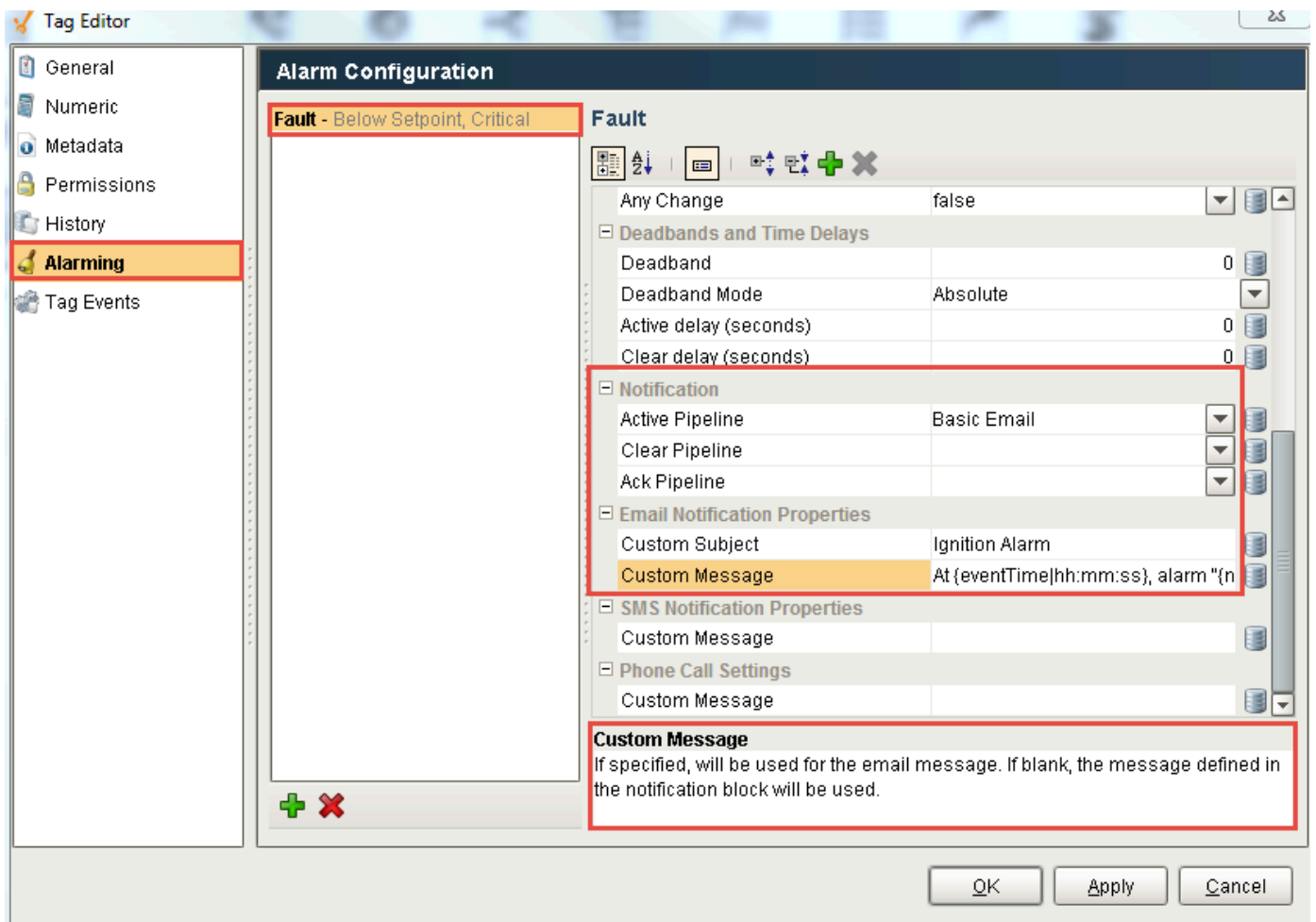
1. To customize an Email alarm notification, select your Tag, and go to the **Alarm Configuration** screen. Scroll down and you will see **Notifications**, **Email Notification Properties**, **SMS Notification Properties**, and **Phone Call Settings**.
2. In the **Notification** section, you can specify the name of the alarm pipeline that you want to trigger when an alarm becomes Active, Cleared and Acknowledged. Simply enter the name of the alarm pipeline next to the relevant state you want the email sent out from. This example uses the '**Basic Email**' pipeline, and the alarm is triggered on the '**Active**' state.
3. Enter a **Custom Subject** called '**Ignition Alarm**'.
4. Create your **Custom Message**. You can use static text, bring in Tag values and special alarm properties. You can even bind a particular property to an expression to concatenate a string that is using other values that are in the system. This example displays the alarm name, state, and time in hours, minutes, and seconds.

Custom Message Script

```
The "{name}" alarm became {eventState} at {eventTime|hh:mm:ss} with a value of "{Value}"
```

Notice that when you select any fields in the Alarm Configuration, the Description Area at the bottom of the window displays details about the property's behavior.

5. When complete, press **OK**.

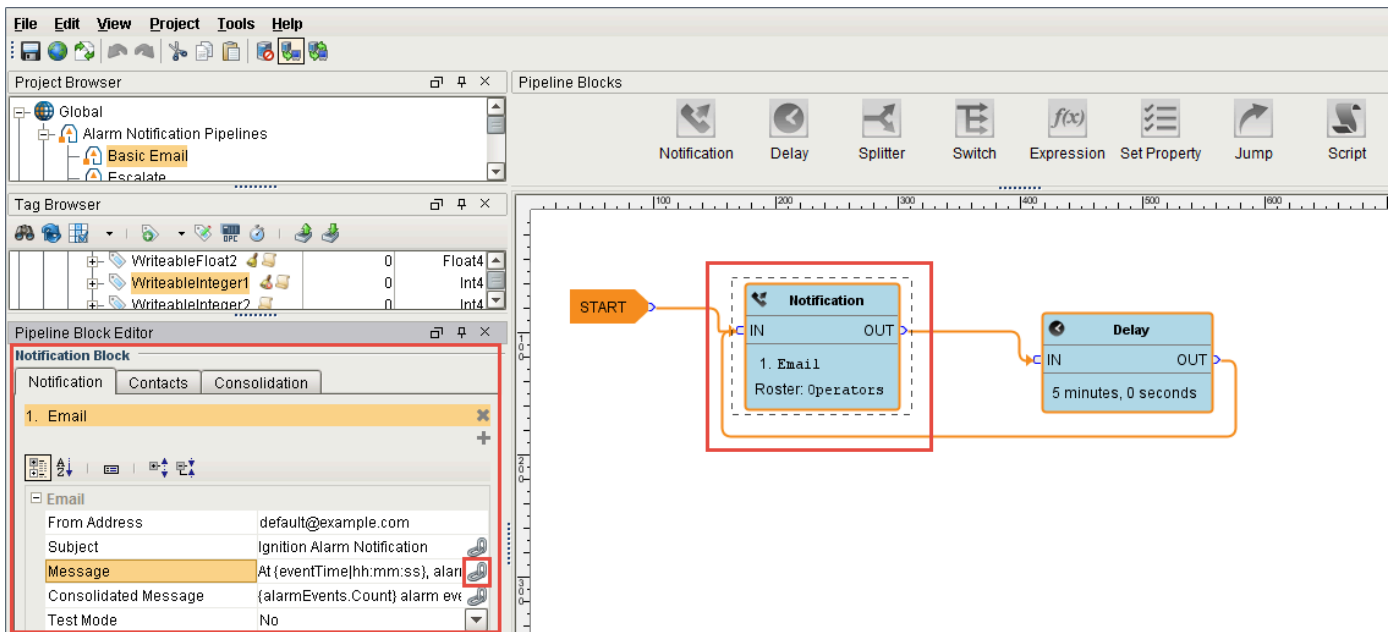


6. In the Pipeline **Notification Block**, go to the **Contacts** tab, and select the **On-Call Roster** that you want to send alarm notifications to.
7. If **Consolidation** is required, enable **Consolidation** and edit the **Delay** and **Frequency** settings.
 - **Delay** - How long to wait after the first eligible alarm arrives.
 - **Frequency** - Maximum frequency with which the profile can send alarms.

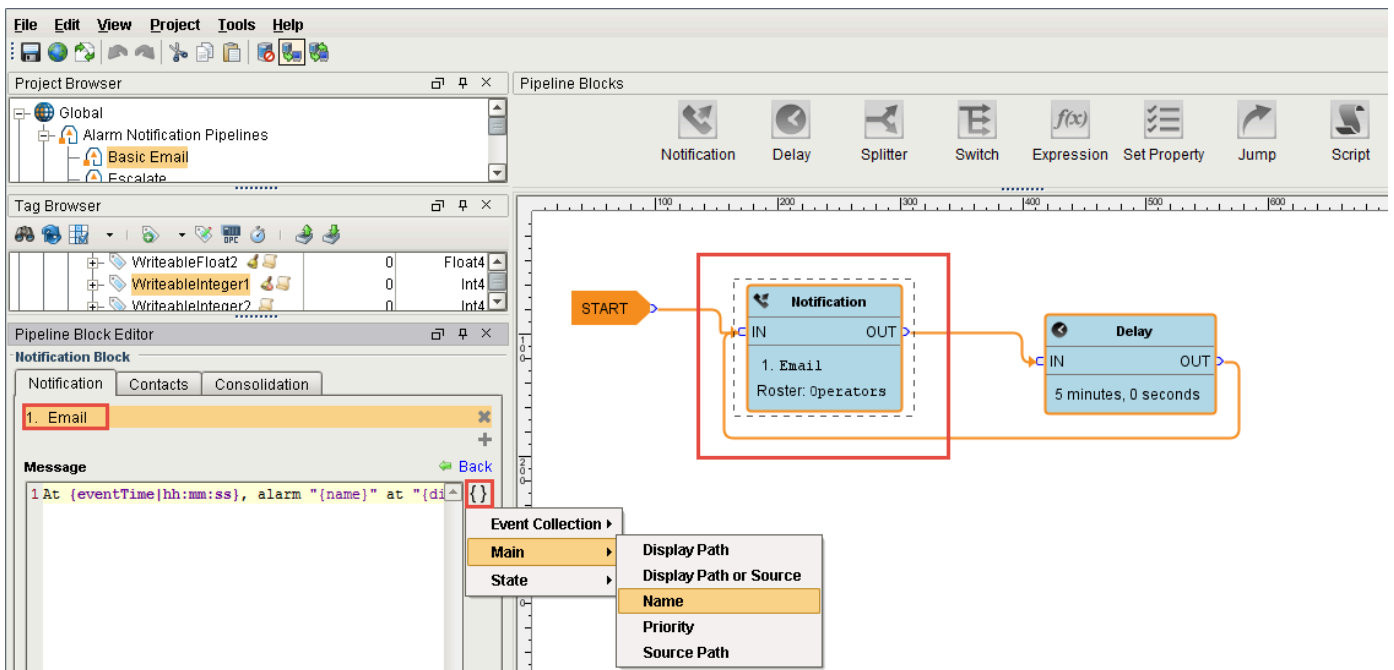
For more information on the Delay and Frequency settings, refer to [Consolidation](#).

Customizing Email Message Scripts using the Pipeline Notification Block

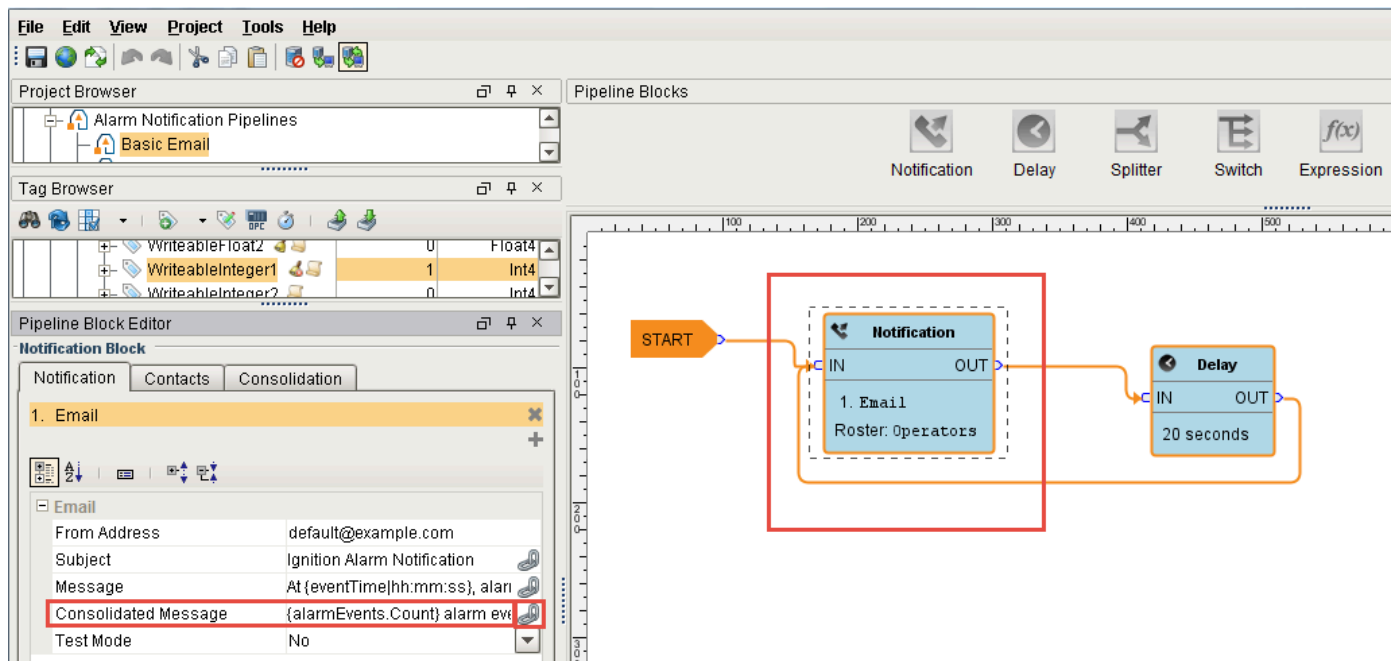
1. Click on the **Notification Block** of your pipeline, and choose **Email**. You can see all the default **Email settings** for each field in the Notification Block. Click on the **Link** icon next to the **Message** field to expand the script.



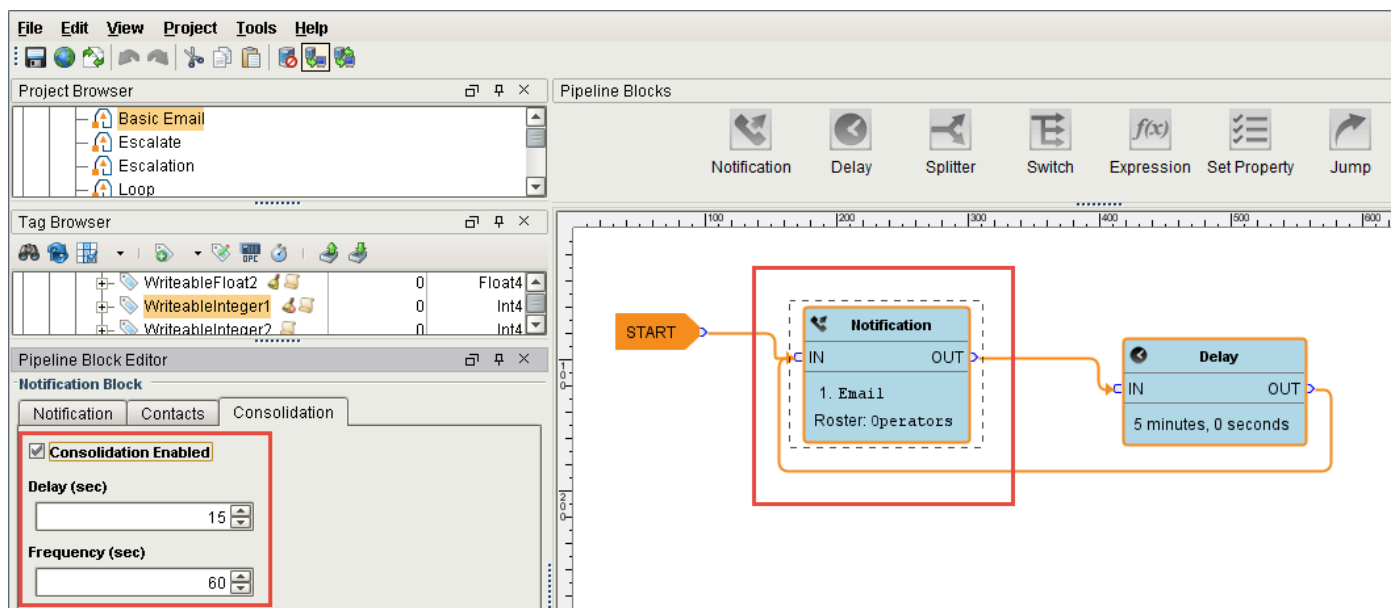
- Here, you can customize the default message script using the built-in alarm properties. Click on the **Curly Braces** icon to see a list of properties to help you build your custom message script.



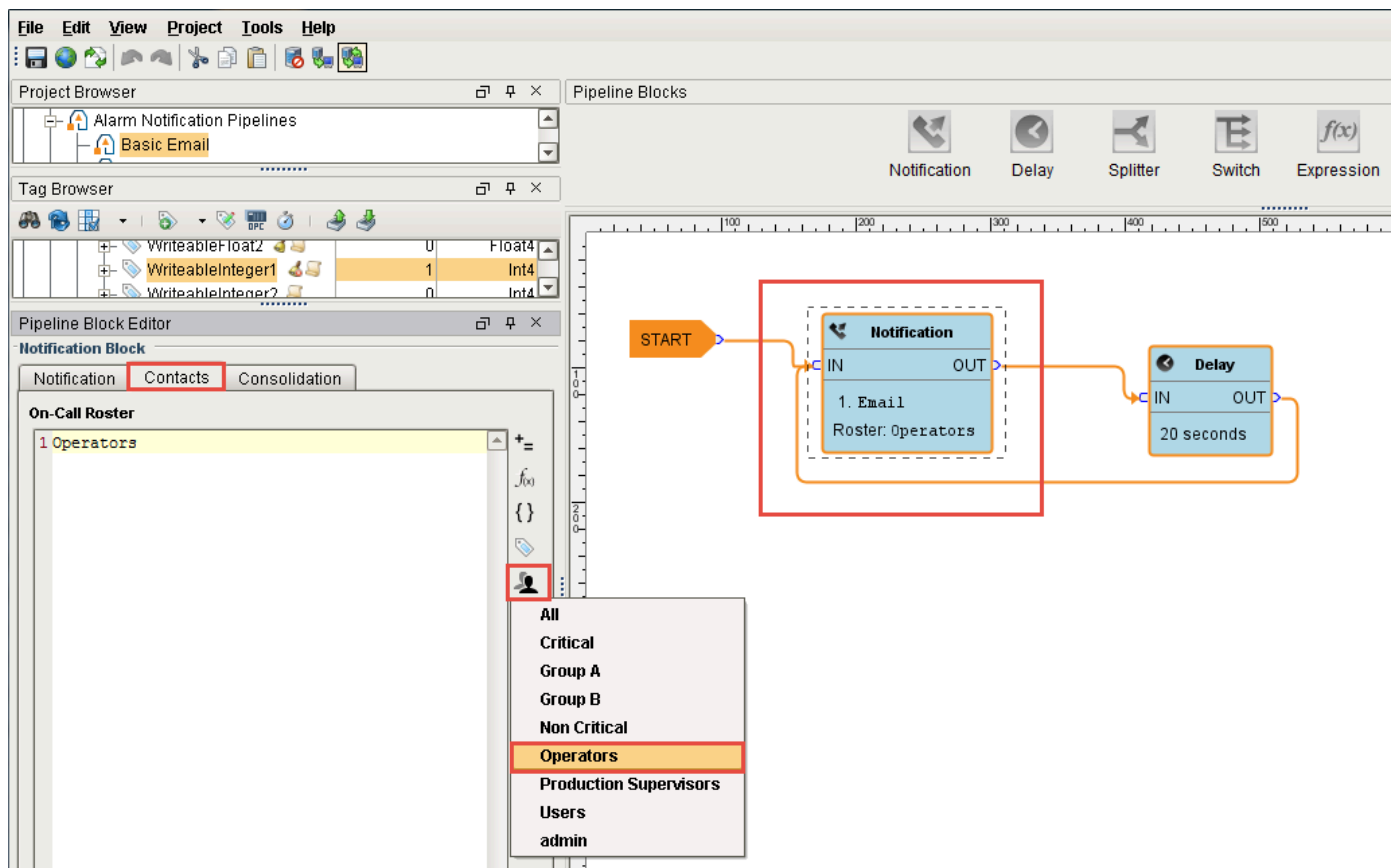
- If you want to use **Consolidation**, you can use the default **Consolidated Message** script or create a custom script using the alarm properties. Click on the **Link** icon to expand the Consolidated Message script.



- If required, go to the **Consolidation** tab, and enable **Consolidation**. When enabled, multiple alarm events are sent together in one email, which results in fewer messages being sent out. For more information on the Delay and Frequency settings, refer to [Consolidation](#).



- Next, from the **Contacts** tab, choose the **On-Call Roster** that you want to send email notifications too.

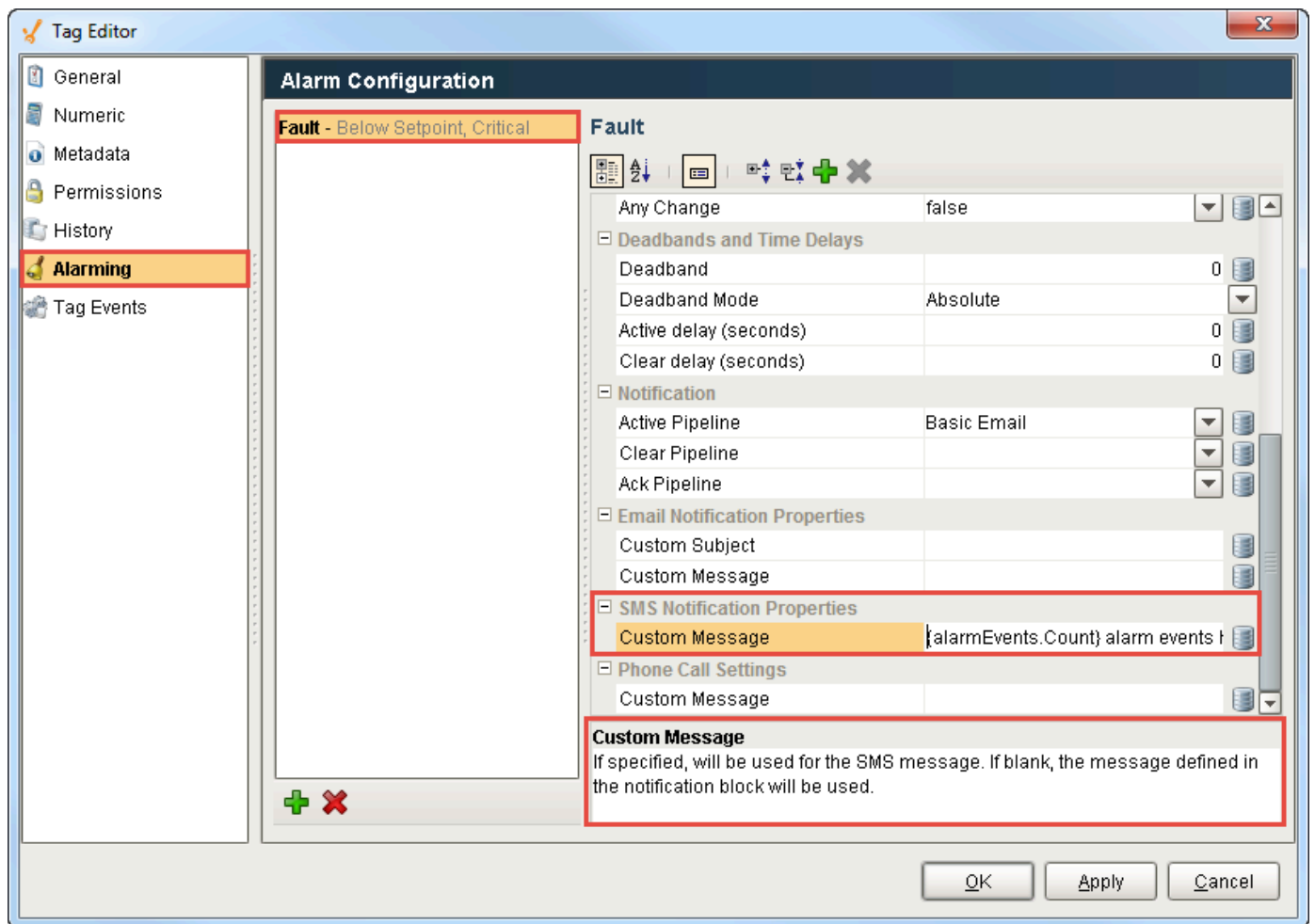


SMS

SMS works the same way as Email except for a few subtle differences.

Customizing SMS Message Scripts in the Alarm Configuration of the Tag Editor

1. To customize an SMS alarm notification, select your Tag, and go to the **Alarm Configuration**. Scroll down to **SMS Notification Properties**.
2. In the **Notification** section, specify the name of the alarm pipeline that you want to use when the alarm becomes Active, Cleared and Acknowledged. In this example, the name of the pipeline is '**Basic Email**' and is triggered on the '**Active**' state.
3. Enter your **Custom Message** script or use the default script. You will notice there is no Subject, only a Custom Message.
4. When you're finished customizing the alarm configuration, click **OK**.

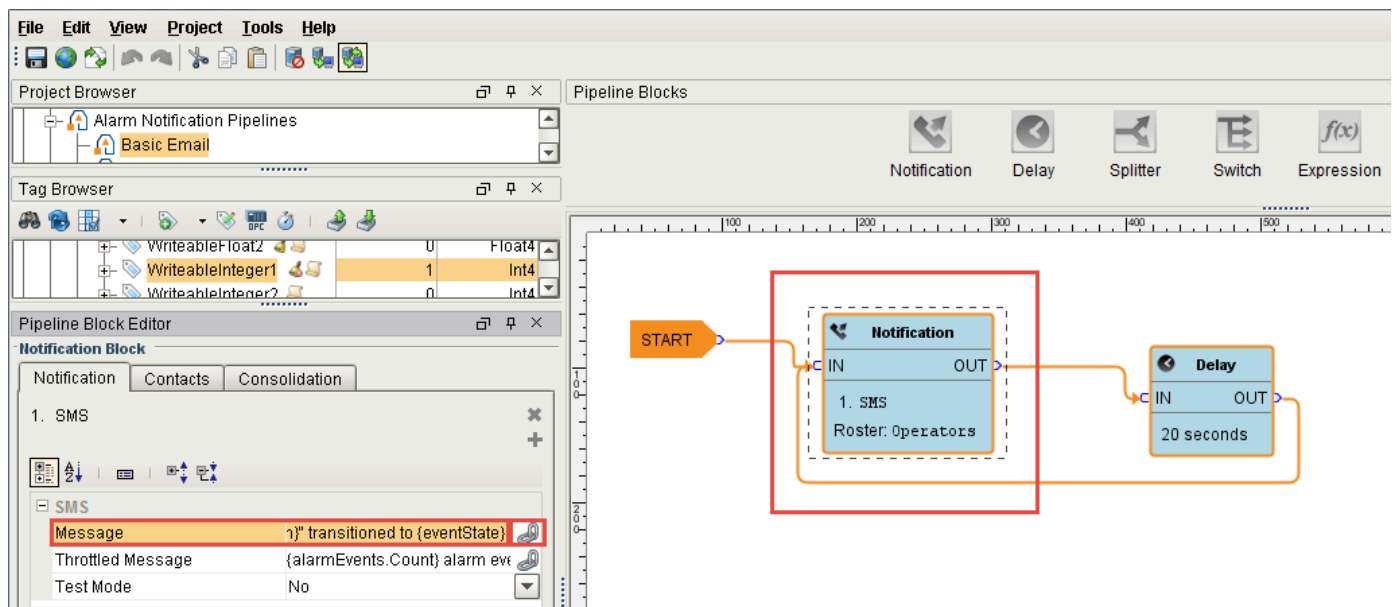


5. Go to the **Pipeline Notification Block** and from the **Contacts** tab, choose the **On-Call Roster** that you want to send alarm notifications too.
6. If required, go to the **Consolidation** tab, and **enable Consolidation**. You can use the default Consolidated Message script or customize the script. Set the **Delay** and **Frequency** settings.

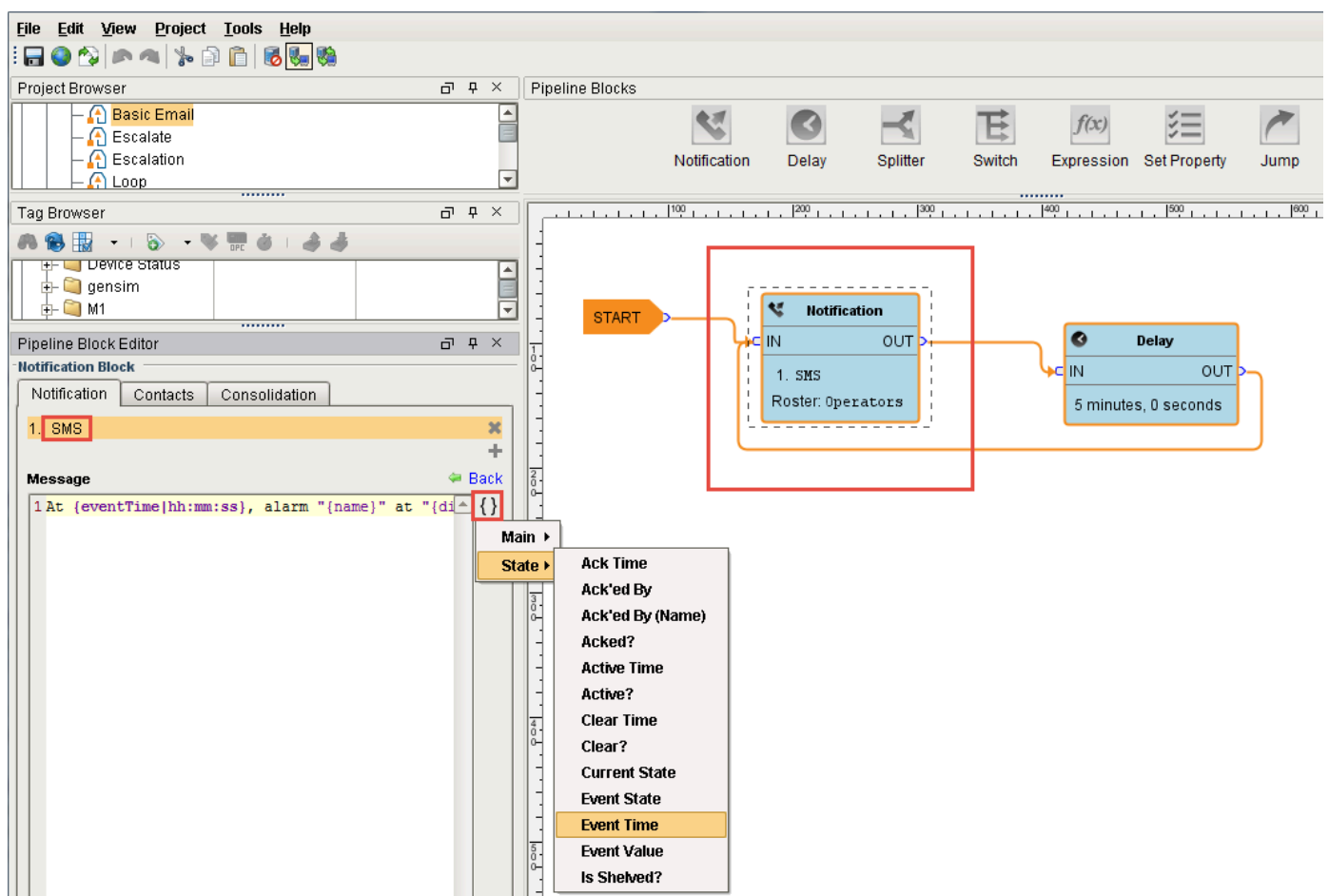
Customizing SMS Message Scripts in the Pipeline Notification Block

Another way to customize the default SMS alarm notification is to build your custom message in the Pipeline Notification Block.

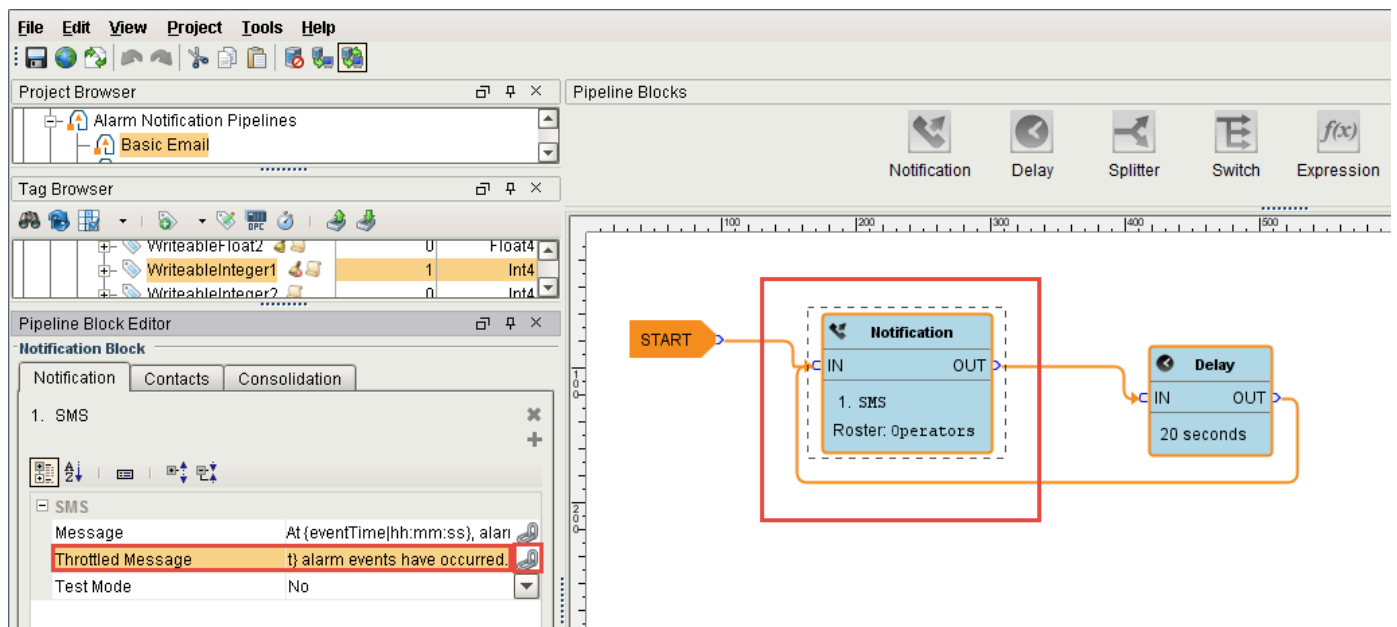
1. Click on the **Notification Block** of your pipeline, and choose **SMS**. You can see all the default **SMS settings** in the Notification Block. Click on the **Message link** icon to expand the script.



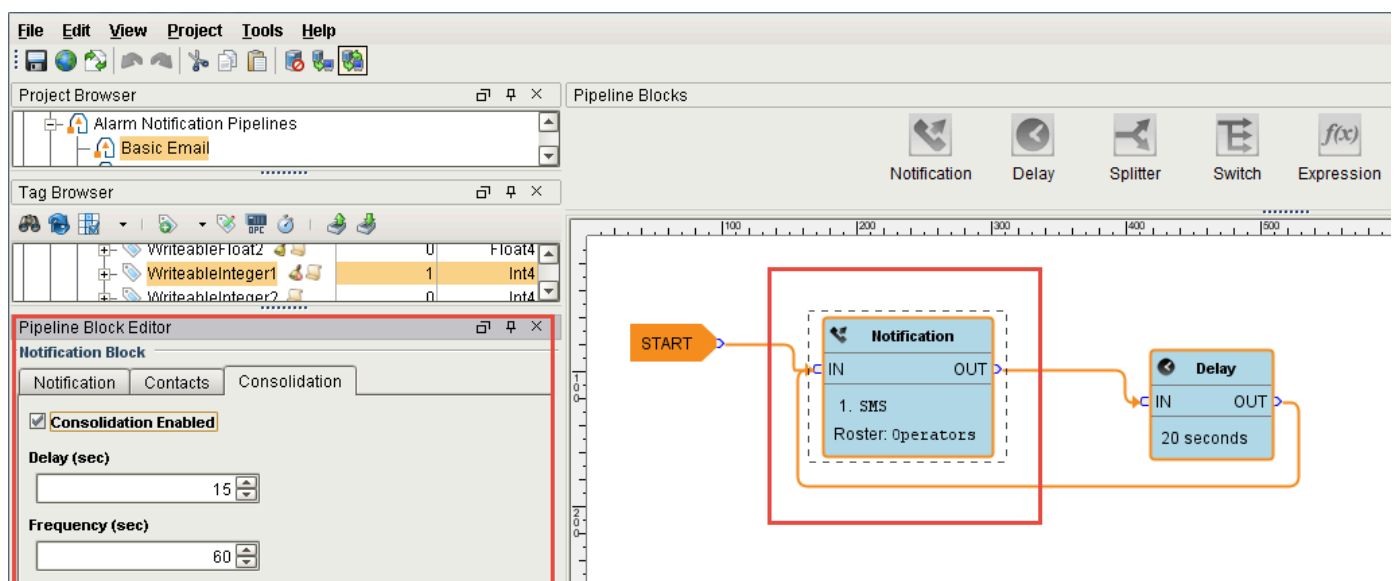
2. You can either use the default message script or customize one using the alarm properties. Click on the **Curly Braces** to see a list of available properties.



3. Like Email, use the Consolidated Message when multiple alarm events are sent together. In **SMS**, click on the **Throttled Message** link icon to expand the consolidated script. You can either use the default script or create your own custom message using the alarm properties.



4. From the **Contacts** tab, choose the **On-Call Roster** that you want to send alarm notifications too.
5. If required, go to the Consolidation tab, and **enable Consolidation**. Enter your **Delay** and **Frequency** settings.



Voice

The Voice Notification Module lets you deliver alarm notifications to users via telephone.

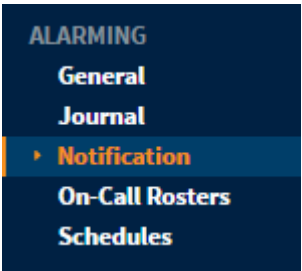
SIP and VOIP

Voice calls are delivered through any SIP compatible phone system. No dedicated hardware is required. Using a hosted service like Skype is the quickest way to implement the Voice Notification Module. For more information on how to configure any SIP compatible phone system, refer to the [Voice Notification Profile](#).

Voice Call Script

The voice message played to the the user during the phone call is defined in the call script which is located in the **Voice Notification Profile**. The voice message is built off a message template that can reference properties in the alarm. Be aware that the voice message is shared across all voice notification profiles. If you define a custom message, it will be used instead of the default message in the call script. Use the following steps to access the default call message template and create your own voice call script.

- 1. Go to the **Configure** page and select **Alarming > Notification**.



- 2. This opens the Alarm Notification Profiles. Next to the Voice Notification Profile, click **More** and select **Manage Scripts** from the dropdown.

Alarm Notification Profiles

Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	deleteedit
Email 1		true	Email Notification	Running	deleteedit
SMS		true	SMS Notification	Running	deleteedit
Voice		true	VOIP Voice Notification	Registered with VOIP Host	manage scriptsdelete More▼edit

→ Create new Alarm Notification Profile...

→ Test Pipelines and Notification Profiles...

- 3. Select your preferred language script and click **Edit**.

scripts

Language	Country	
English	General	edit
Italian	General	delete edit
French	General	delete edit
Spanish	General	delete edit

[→ Create new script...](#)

4. Edit the voice message script and click **Save Changes**.

New in 7.9.12 You can stop any of these Call Scripts from being read by removing all text in that field.

Edit script

Main

Language

English (General)

(default: en)

Call Script

Greeting

Hello, this is an Ignition alarm notification.

PIN Challenge

Please enter your pin, followed by the pound sign.

A message asking the user to enter their PIN number. Must instruct them to press '#' when finished.

Non-PIN
Challenge

Please press any key to continue.

When a PIN is not required, this message is played to instruct the users to press any key to continue.

Invalid PIN

Invalid pin number.

Summary

There are %d events.

When there are multiple alarms, this message will be played to tell the user how many to expect. It must include the placeholder '%d' somewhere in the message.

Active Message

At {eventTime|hh:mm:ss}, alarm named {name} became {eventState} with a value of {eventValue}

The message to play for active alarms. This message can reference properties of alarms by using syntax like "{propertyName}".

Clear Message

At {eventTime|hh:mm:ss}, alarm named {name} became {eventState} with a value of {eventValue}

The message to play for cleared alarms. This message can reference properties of alarms by using syntax like "{propertyName}".

Option 1:
Acknowledge

Press 1 to acknowledge

Tells the user to press '1' to acknowledge.

Acknowledge
Confirmation

Alarm has been acknowledged.

Tells the user that the alarm event has been successfully acknowledged.

Option 2: Ignore

Press 2 to ignore

Tells the user to press '2' to ignore/bypass the alarm.

Ignore
Confirmation

Alarm has been ignored.

Option 3: Repeat

Press 3 to repeat

Tells the user to press '3' to repeat the last message.

Closing

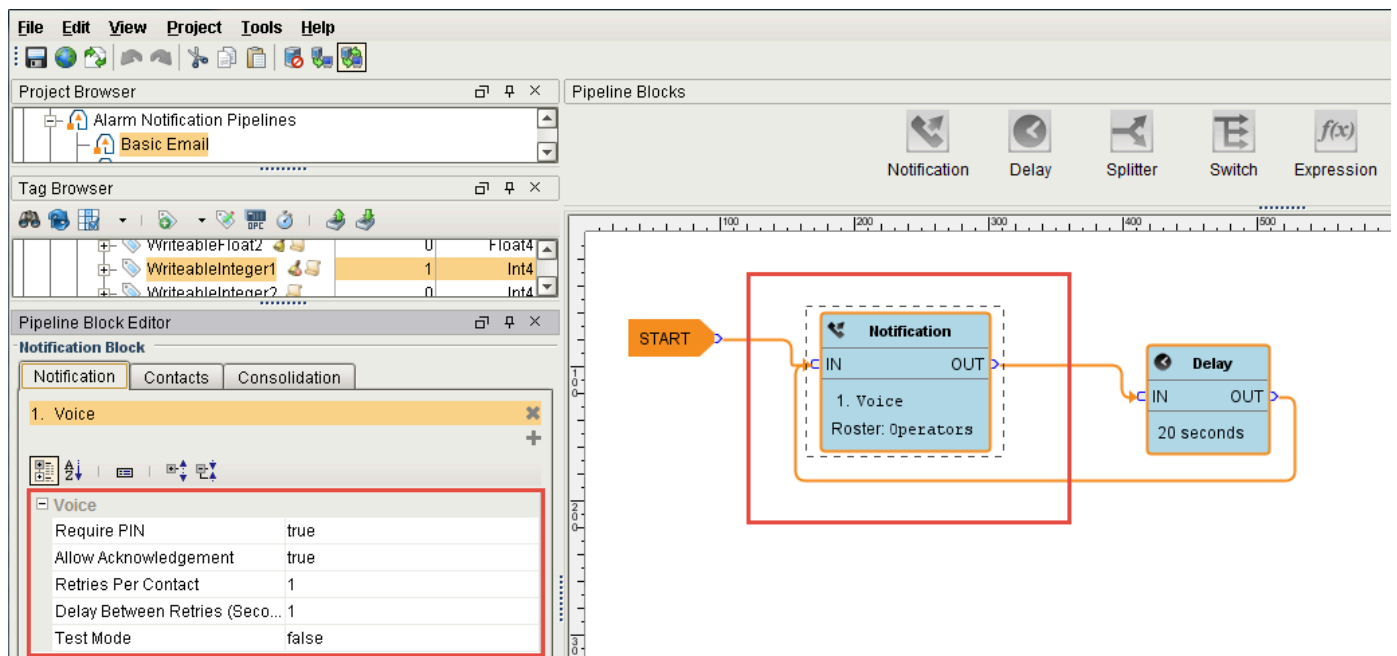
Thank you, goodbye.

Save Changes

Configuring the Voice Call Script

The Voice Notification system is primarily accessed through the Notification Block. The only settings you can configure in the Alarm Configuration section is the Custom Message, and the Alarm Pipeline which defines what state you want to trigger the alarm. All **Voice settings** that you need to configure are in the Notification Block.

1. Click on the **Notification Block** of your pipeline, and choose **Voice**.
2. Configure each setting for your Voice alarm notification, as required.
 - **PIN Required** - If required, the user will need a PIN to hear the alarm notification. PINs are defined in the user's profile.
 - **Allow Acknowledgement** - If Acknowledgement is not required, users will only be allowed to listen to an alarm and not acknowledge them.
 - **Retries per Contact** - Number attempts to contact user by phone.
 - **Delay between Calls** - Creates a delay between each contact for each alarm. For most phone systems, a delay of 1 second is recommended.
 - **Test Mode** - If enabled, messages will be logged to the console indicating when and to whom calls would have been made without actually making a call.
3. This example shows how you might configure your Voice settings, but it depends primarily on your business requirements.
 - PIN Required - **True**
 - Allow Acknowledgement - **True**
 - Retries per Contact - **1**
 - Delay between Retries - **1**
 - Test Mode - **False**



4. From the **Contacts** tab, specify your **On-Call Roster**. Users in the Voice On-Call Roster must have a telephone number, and an established PIN in their user profile.
5. Voice notification supports message consolidation. If required, enable **Consolidation** and enter **Delay** and **Frequency** settings.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

User PIN for Voice

You can force an operator to enter a personal identification number (PIN) in order to hear a Voice alarm notification message. It's a good idea to validate that the person who is receiving the Voice alarm is the person they say they are by entering their PIN.

When alarms are sent out via Voice, you typically want an operator to acknowledge the alarm by pressing '1' on their phone. Requiring a PIN also prevents other people who might be sharing a phone from acknowledging important alarms.

It is easy to setup Security PINs in Ignition in a couple simple steps. First, a PIN number will need to be created in each user's profile to listen to Voice messages. The second step is to configure the Voice.

User Profile

Let's begin by configuring a Security PIN for each user that will be receiving a Voice message and acknowledging SMS alarm messages.

1. Go to the **Gateway Configure** page, **Security > Users, Roles**, and under **default user sources**, select the **manage users** link.

User Sources

Name	Type	Description	
default	Internal	This is the default and always present internal authentication profile.	manage users edit
opcua-module	Internal	OPC-UA clients will authenticate against this profile by default.	More ▾ edit

2. Here you will see list of all your users. For every user that is using Voice and/or SMS alarm notifications, you will need to edit their user profile to create a PIN. Select the **Edit** link for one of your users.

Manage Users and Roles for Profile 'default'

Users		Roles		
Username	Name	Roles	Contact Info	Schedule
admin	Maria	Administrators, Operator, Administrator	email: admin@inductiveautomation.com, phone: 800-266-7798	Always
guest		Users, Prod_Sups		Always
janedoe	Jane Doe	Operator	email: jane_doe@mailinator.com	Weekday Day Shift

3. The user profile screen will open. Scroll down to the bottom of the screen, and make sure the contact information for Email and Phone is up-to-date.

Create a 4-digit **Security Pin**, and press **Save Changes**.

Now, this PIN is associated with this user.

Contact Info

Type	Value	
E-Mail	admin@inductiveautomation.com	Up / Down Delete
Phone	800-266-7798	Up / Down Delete

Add Contact Info

Extended Properties

Security PIN	1234
--------------	------

< Cancel

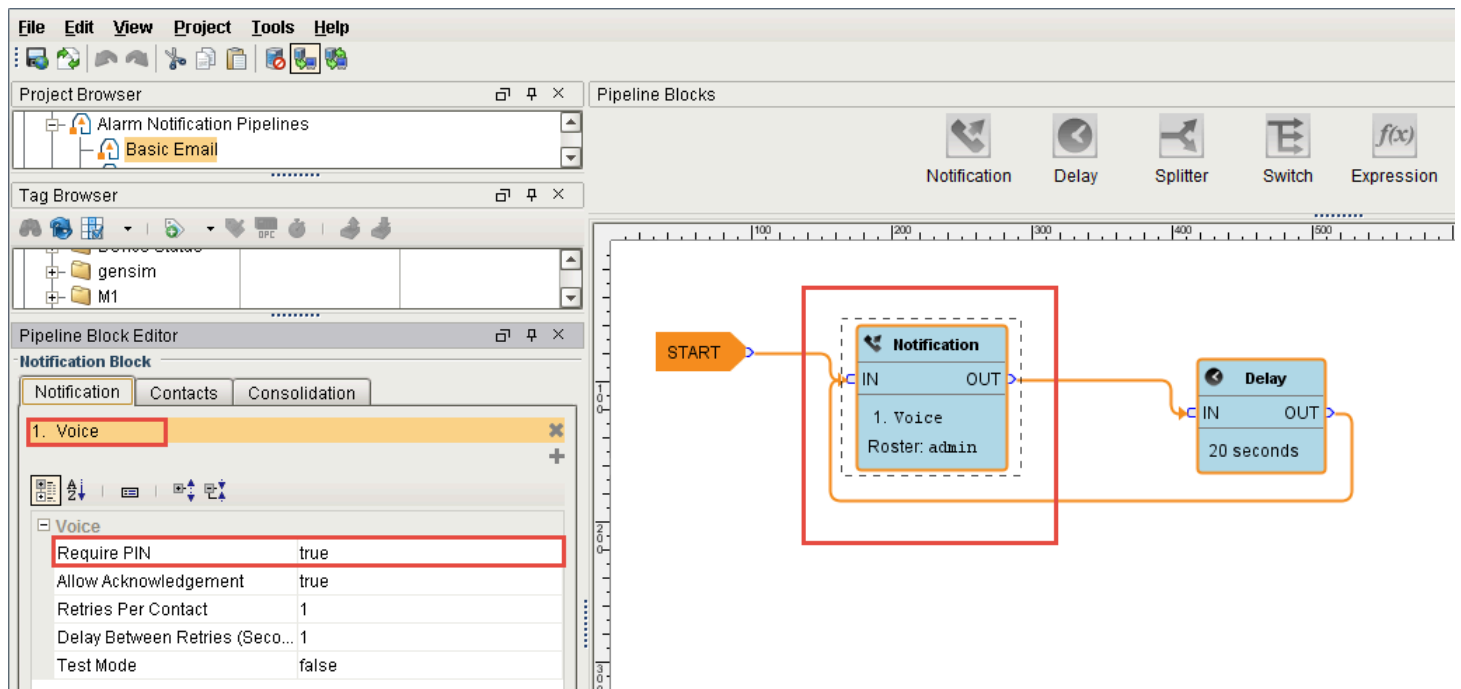
Save Changes

Voice Notification Profile

The next step is to configure the Voice Notification Profile to require a PIN.

1. In **Designer**, go to your **Alarm Notification Pipelines** and open your pipeline. This example uses the **Basic Email** pipeline. Select the **Notification Block** in the workspace.
2. Click on the **Notification** tab, and choose **Voice**.
3. You'll see several settings for Voice. Set the **Required PIN** to 'true'.

Now, when the operator receives a call, they will have to type in their PIN number followed by the pound sign to hear the message.



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Voice Languages



INDUCTIVE
UNIVERSITY

Voice Languages

[Watch the video](#)

Voice Languages Overview

When alarm events happen, Ignition uses the Voice Notification Module and the voice language modules to send alarm messages. The Voice Notification Module supports making calls using male and female voices in multiple languages. The language modules (i.e., English, Spanish, Italian, etc.) contain the text to speech engine that is specific for that language and gender.

You can download these modules from [Inductive Automation's Downloads page](#) on the website, and install them in your Gateway.

Home / Downloads



Try Ignition now.

Download and start using the world's most powerful SCADA software in minutes.

[Download Ignition](#)
[Get Started with IIoT](#)
[What is IIoT?](#)

Ignition-7.8.2-windows-x64-installer.exe

Not the file you were looking for? Check out the rest of our [system installers](#)

What's new in
Ignition 7.8

Learn Ignition for free with
Inductive University

Have a Problem?
Get Support

Notification

Alarm Notification Module

Alarm Notification-module.modl (325KB)

Version: 3.8.2.2016030813

SMS Notification Module

SMS Notification-module.modl (196KB)

Version: 3.8.2.2016030813

Voice Notification Module

Voice Notification-module.modl (14MB)

Version: 3.8.2.2016030813

Voice Module Downloads

Looking for additional Text-to-Speech voices? Take a look at this [Knowledgebase article](#).

The following table shows a list of the current Text-to-Speech voices that you can download after you have the Voice Notification Module installed. To download Text-to-Speech voices, go to the Knowledge Base article on [Voice Notification Downloads](#) where you can also find some good information.

Voices	Languages
Katherine	English
Laura	Italian
Sara	Spanish
Suzanne	French
Alex	German

Once you have [installed the voice modules](#), you can configure the voice scripts for any language in the voice notification module. For example, you may want to have a different script that gets sent out when you're calling someone in English verses someone that you're calling in Spanish.

NOTE

Important!

On this page we mention "**scripts**" a lot. These scripts have **nothing** to do with the scripting language (Python) that you write code with. When we say "script" on this page, we are talking about a speech or a dialog for Ignition to read to your users over the phone.

Modifying a Voice Notification Script

From the Voice Notification Profiles page, you can manage your voice scripts. Two examples are provided here; one in English, and the other in Spanish.

1. Go to your **Gateway Configure** page, and select **Alarming > Notification**. To the right of the Voice Notification Profile, click on the **manage scripts** link.

Alarm Notification Profiles

Name	Description	Enabled	Type	Status	
Email		true	Email Notification	Running	deleteedit
Email 1		true	Email Notification	Running	deleteedit
SMS		true	SMS Notification	Running	deleteedit
Voice		true	VOIP Voice Notification	Registered	manage scriptsdelete Moreedit

→ Create new Alarm Notification Profile...

→ Test Pipelines and Notification Profiles...

2. Here, you can manage all the scripts for every language, and can even create new scripts. The languages that come with Ignition are English, Italian, French and Spanish. There are a host of other languages that Ignition supports. For this example, click on the **edit** link for the **English** language.

scripts

Language	Country	
English	General	edit
Italian	General	delete edit
French	General	delete edit
Spanish	General	delete edit

[→ Create new script...](#)

3. Default voice scripts are the same in all supported languages. This is the default script for English. It consists of a number of little scripts. Each script is what the recipient of the call will hear over the phone. These are instructions informing the recipient of one or more the alarm(s) depending on how the alarm notification is configured. When you receive a voice notification, you will hear each script or instruction one at a time.

For example, the first script you will hear is “Hello this is an Ignition alarm notification” which, of course, is the greeting. It will then go through each script and inform the recipient of the alarm name, date and time of the alarm, alarm status, and how to acknowledge the alarm. You can edit this default script to your requirements. When you're finished editing, click **Save Changes**.

Edit script

Main

Language

English (General)

(default: en)

Call Script

Greeting

Hello, this is an Ignition alarm notification.

PIN Challenge

Please enter your pin, followed by the pound sign.

A message asking the user to enter their PIN number. Must instruct them to press '#' when finished.

Non-PIN Challenge

Please press any key to continue.

When a PIN is not required, this message is played to instruct the users to press any key to continue.

Invalid PIN

Invalid pin number.

Summary

There are %d events.

When there are multiple alarms, this message will be played to tell the user how many to expect. It must include the placeholder '%d' somewhere in the message.

Active Message

At {eventTime|hh:mm:ss}, alarm named {name} became {eventState} with a value of {eventValue}

The message to play for active alarms. This message can reference properties of alarms by using syntax like "{propertyName}".

Clear Message

At {eventTime|hh:mm:ss}, alarm named {name} became {eventState} with a value of {eventValue}

The message to play for cleared alarms. This message can reference properties of alarms by using syntax like "{propertyName}".

Option 1: Acknowledge

Press 1 to acknowledge

Tells the user to press '1' to acknowledge.

Acknowledge Confirmation

Alarm has been acknowledged.

Tells the user that the alarm event has been successfully acknowledged.

Option 2: Ignore

Press 2 to ignore

Tells the user to press '2' to ignore/bypass the alarm.

Ignore Confirmation

Alarm has been ignored.

Option 3: Repeat

Press 3 to repeat

Tells the user to press '3' to repeat the last message.

Closing

Thank you, goodbye.

Save Changes

4. Here is an example of the default scripts in Spanish. Once again, you can edit the scripts to your requirements. When editing is complete, click **Save Changes**.

Edit script

Main	
Language	Spanish (General) ▼ (default: en)

Call Script	
Greeting	Hola esta es una notificación de alarma.
PIN Challenge	Por favor, introduzca su número de PIN seguido de la tecla numeral. A message asking the user to enter their PIN number. Must instruct them to press '#' when finished.
Non-PIN Challenge	Presione cualquier número para continuar. When a PIN is not required, this message is played to instruct the users to press any key to continue.
Invalid PIN	Ese número PIN es incorrecto.
Summary	Hay %d alarmas. When there are multiple alarms, this message will be played to tell the user how many to expect. It must include the placeholder '%d' somewhere in the message.
Active Message	A la hora {eventTime hh:mm:ss} la alarma llamada {name} se convirtió {eventState} con un valor de {eventValue}. The message to play for active alarms. This message can reference properties of alarms by using syntax like "{propertyName}".
Clear Message	A la hora {eventTime hh:mm:ss} la alarma llamada {name} se convirtió {eventState} con un valor de {eventValue}. The message to play for cleared alarms. This message can reference properties of alarms by using syntax like "{propertyName}".
Option 1: Acknowledge	Presione 1 para confirmar la alarma. Tells the user to press '1' to acknowledge.
Acknowledge Confirmation	La alarma ha sido confirmada. Tells the user that the alarm event has been successfully acknowledged.
Option 2: Ignore	Presione 2 para ignorar la alarma. Tells the user to press '2' to ignore/bypass the alarm.
Ignore Confirmation	La alarma ha sido ignorada.
Option 3: Repeat	Presione 3 para repetir el mensaje. Tells the user to press '3' to repeat the last message.
Closing	Gracias, adiós.

You can attach a different script for any language that Ignition supports depending on the user, and whatever that user is setup to do. So, if the script is in Spanish, the user will receive the voice notification in Spanish assuming the Spanish voice module is installed in Ignition.

5. Once you have the appropriate voice module and language module(s) installed, go to **Security > Users and Roles**. Select the **manage users** link for the default user source.

User Sources

Name	Type	Description	
default	Internal	This is the default and always present internal authentication profile.	manage users edit
opcua-module	Internal	OPC-UA clients will authenticate against this profile by default.	More ▾ edit

6. You can edit every single user, and put each user into a different language, as long as the language that you select for each user has the voice module for the language installed. In this example, let's edit a user named Sally. Put Sally into the Spanish language. If the voice module is installed for the Spanish language, Ignition will automatically choose the Spanish script.

Manage Users and Roles for Profile 'default'

Users

Roles

Edit user "admin"

User Properties

Username

admin

Change Password?

☐ Check this box to change the existing password.

Password

Latviesu (Latvija)
Lietuvių
Lietuvių (Lietuva)
Malti
Malti (Malta)
Nederlands
Nederlands (België)
Nederlands (Nederland)
Slovenčina
Slovenčina (Slovenská republika)
Slovenščina
Slovenščina (Slovenija)
Srpski (Latin)
Srpski (Latin,Bosna i Hercegovina)
Srpski (Latin,Crna Gora)
Srpski (Latin,Srbija)
Tiếng Việt
Tiếng Việt (Việt Nam)
Türkçe
Türkçe (Türkiye)
català
català (Espanya)
español

First Name

Last Name

Roles

☒ Operator

☐ Prod_Sups

☐ Test_Group

☐ Users

Schedule

Language

español

Notes

Now, when the alarm notification is received, the voice message will be in Spanish. You can also change a user's language preferences from the **User Management** component.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Journal

By default, current alarm data is only stored in memory, and a finite number of events are retained for each alarm. Fortunately, Ignition can easily be configured to store alarm data into a SQL database with an Alarm Journal Profile. The journal can store basic data about alarms that have occurred, such as their source and timestamp, associated data on the alarm, and the values of the alarm's properties at the time the event occurred. The alarm journal is used by the **Alarm Journal Table** component, and can be accessed through scripting functions and direct database queries.

The Gateway can have more than one Alarm Journal. Alarm Journals have options to filter which Alarms are stored in the journal, therefore by having more than one alarm journal configured on the Gateway, it is possible to store some alarms in one journal, and different alarms in another journal.

Once configured, the Journal can be accessed by the **Alarm Journal Table** component, scripting functions, or direct database queries.

Store Alarms

Create a single Alarm Journal to store all of your alarms, or create multiple Journals to store alarms across multiple databases. Each Journal stores alarms based on the filters you set up, and can prune data automatically after a set time limit.

Gateway-Name

USER MANUALSUPPORT

adminSign Out

Launch Designer

Ignition
by inductive automation

HOMESTATUSCONFIGURE

Search...

SYSTEM

OverviewBackup/RestoreLicensingModulesProjectsRedundancyGateway Settings

NETWORKING

Gateway NetworkEmail Settings

SECURITY

Alarm Journal Profiles

Name	Datasource		
Administrative_Alarms	Test	delete	edit
Journal	Test	delete	edit
Packaging_Area	Test	delete	edit

→ Create new Alarm Journal Profile...

Third Party Accessibility

Because the Alarm Journal uses a SQL database to log alarm events, any application that has access to the database can retrieve journal data. Alarm events can be made freely available outside of Ignition, and integrated into other software packages. Additionally, other applications can write to the same tables, so Ignition applications can monitor activity in other systems.

Alarm Journal Component

While a SQL query will return data from the journal, the Alarm Journal Table component will automatically do so without manually writing a query. The component can filter on both **Display Path and Source Path**, as well as **Date Range**. The component can be configured to a single Journal Profile, so multiple instances of the component in the same project may look at different profiles.

Event Time	Display Path	Event State	Event Value
8/4/16 8:22 AM	Gen Sim/Sine/Sine0/Positive Value	Clear	-3.1725
8/4/16 8:21 AM	Gen Sim/Sine/Sine0/Positive Value	Active	4.2085
8/4/16 8:21 AM	Gen Sim/Sine/Sine0/Positive Value	Ack	
8/4/16 8:21 AM	Gen Sim/Sine/Sine0/Positive Value	Clear	-3.1515
8/4/16 8:20 AM	Gen Sim/Sine/Sine0/Positive Value	Active	2.0942

93 events

More information on this component can be found on the [Alarm Journal Table Component](#) page.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Journal Properties and Tables



INDUCTIVE
UNIVERSITY

Create Alarm Journal Profile

[Watch the video](#)

Alarm Journal

In Ignition, an Alarm Journal stores historical information about alarms in a database. It can store basic data about alarms that have occurred, such as their source and timestamp, along with associated data on the alarm, and the values of the alarm's properties at the time the event occurred. The Alarm Journal is used by the **Alarm Journal Table** component that can be accessed through scripting functions and direct database queries.

To Create a New Alarm Journal Profile

1. Go to the **Configure** section of the Gateway.
2. Choose **Alarming** > **Journal** from the menu on the left. The Alarm Journal Profiles page is displayed.
3. Look for the blue arrow and click on **Create new Alarm Journal Profile....** The New Alarm Journal Profiles page is displayed.

Controller

USER MANUAL SUPPORT

admin Sign Out

Launch Designer

HOME STATUS CONFIGURE

Search...

License Incomplete ? 0:00:00 Reset Trial View Modules

SYSTEM

- Overview
- Backup/Restore
- Licensing
- Modules
- Projects
- Redundancy
- Gateway Settings

NETWORKING

Alarm Journal Profiles

Name	Datasource
Journal	Mysql

delete edit

→ Create new Alarm Journal Profile...

4. **New in 7.9.13** Ignition Gateways can now create an Internal Alarm Journal Profile, which stores journal entries in the Gateway's internal database as well as an external SQL database. You can store `alarm_events_data` and `alarm_events` in the SQLite table.

To create an Internal Alarm Journal Profile, select **Internal** as shown in the image below. Click **Next**.

Add Alarm Journal Profile Step 1: Choose Type

☐ **Database**
Alarm journal events will be logged into an external database.

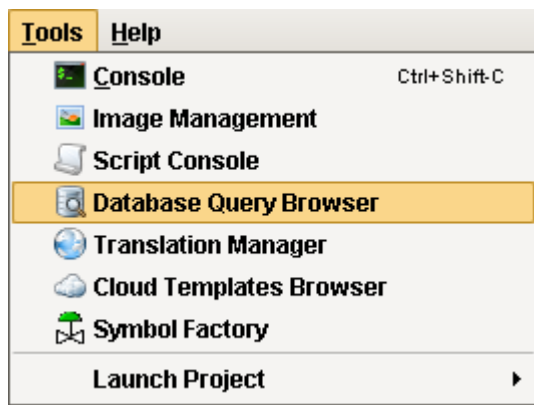
☒ **Internal**
Alarm journal events will be logged into a SQLite database file, local to the gateway.

Next >

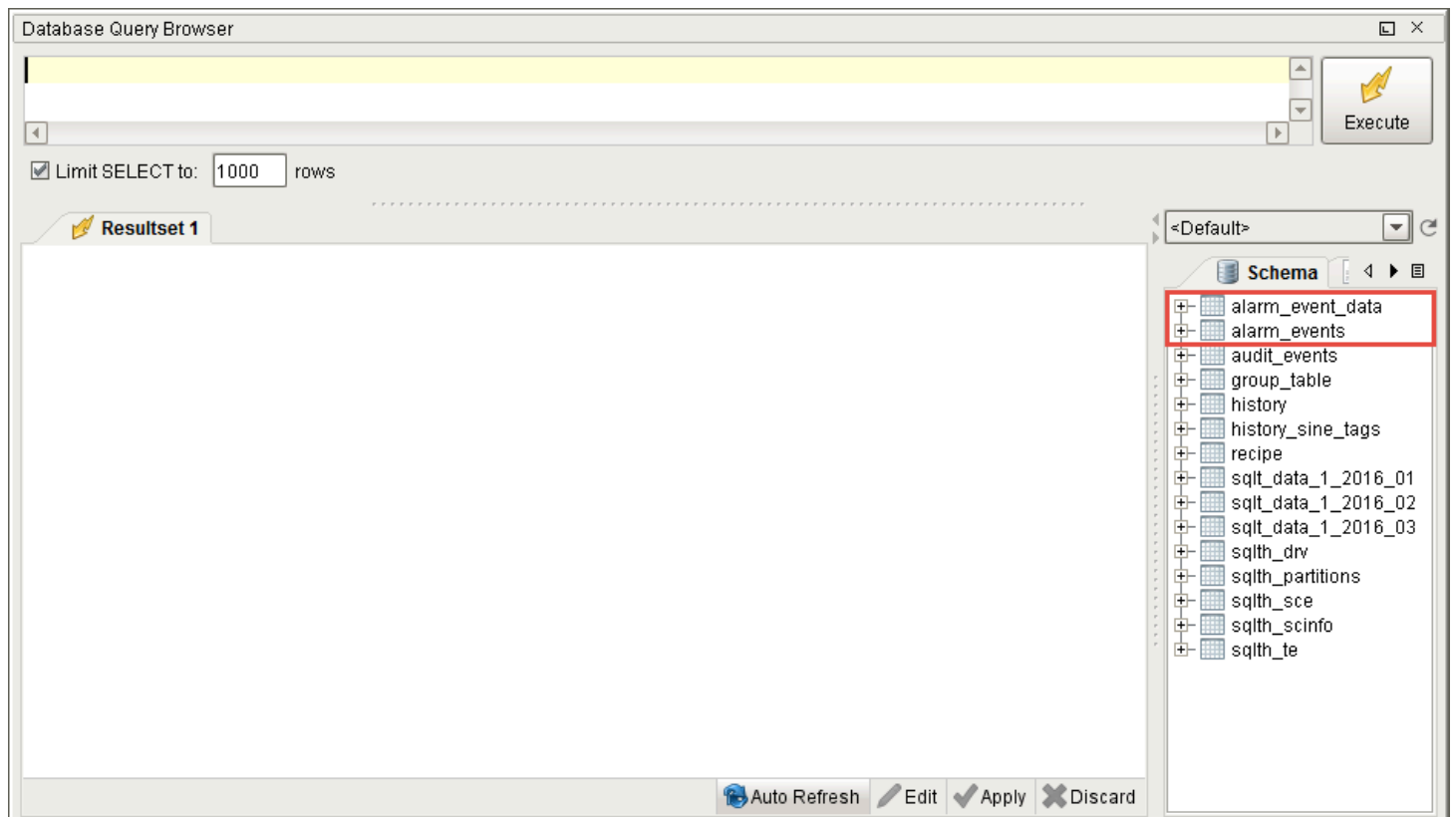
5. Fill in the fields and click the **Create New Alarm Journal Profile** button at the bottom of the page. Once completed, the tables will be created for you once an alarm event occurs. It's a good idea to trigger a test alarm and verify that Ignition automatically created the tables in the database after your Alarm Journal Profile was created.

Verify the Alarm Journal tables were created.

You can easily verify the Alarm Journal tables from the Ignition **Designer**. Open it and go to the menu bar and select **Tools > Database Query Browser**.



The two tables that Ignition automatically creates are the `alarm_events` and `alarm_event_data`. Alarm events consist of two main types of data: the primary information about the alarm, such as transition state, time, etc., and the event data. If you don't see these two tables in your database, you need to make an alarm occur after the Alarm Journal Profile is created. Make the alarm Active, take it back to Clear, Acknowledge it, and you will start seeing information come into these tables.



Double click on the **alarm_events**, to see all the events that occurred when the alarms are Active, Cleared, and Acknowledged.

Database Query Browser

```
SELECT * FROM alarm_events
```

Limit SELECT to: 1000 rows

Resultset 1

id	eventid	source	disp
813	903a6940-b171-4278-9f70-2aec1...	prov.default/tag:gensim/Realistic/Realistic2/alm:Low Alarm	
817	903a6940-b171-4278-9f70-2aec1...	prov.default/tag:gensim/Realistic/Realistic2/alm:Low Alarm	
990	a003a838-c4b2-4ea7-83b6-0566...	prov.default/tag:gensim/Realistic/Realistic2/alm:Low Alarm	
991	dc37348-795d-4a33-9a87-97f36...	prov.default/tag:gensim/Realistic/Realistic2/alm:Low Alarm	
996	dc37348-795d-4a33-9a87-97f36...	prov.default/tag:gensim/Realistic/Realistic2/alm:Low Alarm	
999	37fb22af-69fa-4246-bfc4-f9ace26...	prov.default/tag:gensim/Realistic/Realistic2/alm:Low Alarm	
1000	742d7068-09e8-47cb-b056-acd7f...	prov.default/tag:gensim/Realistic/Realistic2/alm:Low Alarm	
9	27202126-8d8e-4b0a-ad8d-faa72...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
10	052d7871-c94d-4f35-9ecd-38701...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
37	052d7871-c94d-4f35-9ecd-38701...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
39	82693c81-2f06-42f6-806a-8072e...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
40	63ccee97-3735-4b02-9ac0-e9622...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
41	63ccee97-3735-4b02-9ac0-e9622...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
44	1e62c97e-6e07-4e3d-a21e-7418...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
45	72a55b04-1c9c-4481-a7d7-80a4...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
57	72a55b04-1c9c-4481-a7d7-80a4...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
81	00cf730b-e986-49b2-8998-16c04...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	
82	02e0e0e6-52f6-4615-02e0-000f551...	prov.default/tag:gensim/Realistic/Realistic3/alm:High Alarm Low	

1000 rows fetched in 0.108s

Schema

- alarm_event_data
 - id (INT)
 - displaypath (VARCHAR)
 - eventflags (INT)
 - eventid (VARCHAR)
 - eventtime (DATE)
 - eventtype (INT)
 - priority (INT)
 - source (VARCHAR)
- audit_events
- group_table
- history
- history_sine_tags
- recipe
- sqlt_data_1_2016_0
- sqlt_data_1_2016_0
- sqlt_data_1_2016_0
- sqlth_drv
- sqlth_partitions

The **alarm_event_data** is all the associated data that is associated with each alarm.

Database Query Browser

```
SELECT * FROM alarm_event_data
```

Limit SELECT to: 1000 rows

Resultset 1

id	propname	dtype	intvalue	floatvalue	strvalue
308	eventValue	1	NULL	41.222	NULL
309	eventValue	1	NULL	33.851	NULL
311	eventValue	1	NULL	42.752	NULL
312	eventValue	1	NULL	28.581	NULL
313	eventValue	1	NULL	-33.045	NULL
315	eventValue	1	NULL	-36.957	NULL
316	eventValue	1	NULL	-23.09	NULL
318	eventValue	1	NULL	-35.412	NULL
319	eventValue	1	NULL	-33.814	NULL
320	eventValue	1	NULL	33.651	NULL
322	eventValue	1	NULL	-35.714	NULL
324	eventValue	1	NULL	-37.787	NULL
325	eventValue	1	NULL	-27.669	NULL
327	eventValue	1	NULL	37.966	NULL
329	eventValue	1	NULL	38.391	NULL
330	eventValue	1	NULL	-32.495	NULL
332	eventValue	1	NULL	37.733	NULL
333	eventValue	1	NULL	34.137	NULL

1000 rows fetched in 0.077s

Schema

- alarm_event_data
 - dtype (INT)
 - floatvalue (DOUBLE)
 - id (INT)
 - intvalue (BIGINT)
 - propname (VARCHAR)
 - strvalue (VARCHAR)
- alarm_events
- audit_events
- group_table
- history
- history_sine_tags
- recipe
- sqlt_data_1_2016_0
- sqlt_data_1_2016_0
- sqlt_data_1_2016_0
- sqlth_drv
- sqlth_partitions
- sqlth_sce
- sqlth_sinfo

You will want to look into the Database Query Browser to simply verify that the information is there. Once the information is there, close the Database Query Browser.

NOTE

If the tables were not created, check the Gateway console page for any errors.

UDT Parameter Binding on a Label Alarm Property

When using UDT parameters in an alarm binding on a Label property, values are recorded in the Alarm Journal Table when the Static Config property is enabled. If there are not any other types of property references in the binding, then the Alarm Journal will not record UDT parameter changes unless Static Config is enabled.

Reference

Journal Options

Most Journal Options are in both the external Database and Internal SQLite databases, but there are few that are not. Be sure to check the Database and Internal columns in the table below to see if that journal option is used.

Main

Property	Description	Database	Internal
Name	The default name is Journal.	X	X
Enabled	By default the journal profile is enabled.	X	X
Datasource	This is the ONLY required setting which must be a valid database connection, events are stored to this datasource.	X	
Description	Description of the Alarm Journal Profile.	X	X
Minimum Priority	Only events equal to or greater than the specified priority will be stored. The default is Low. You can set the priority to be: Diagnostic, Low, Medium, High, and Critical.	X	X
Store Shelved Events	Not enabled by default. If enabled, events generated by "shelved" alarms will still be written to the journal system.	X	X
Use Store and Forward	Enabled by default, which means the alarm journaled events will be stored through the Store and Forward system. If not enabled, they will be stored directly against the database. This system protects data from being lost due to temporary database connectivity issues.	X	

Stored Event Data

Alarm events consist of two main types of data: the primary information about the alarm, such as transition state and time, and the event data. The following settings specify what type of event data is stored:

Property	Description	Database	Internal
Static Config	By default, it is not selected. If selected, will store the values of static alarm configuration. That is, the alarm properties that are not bound. These do not change during evaluation, only when a user modifies them in the Designer, and so they are not stored by default.	X	X
Dynamic Config	If selected, will store the values of dynamically bound alarm configuration properties. The value of these properties can change at any time, and the values at the time of the alarm are captured on the alarm event.	X	X
Static Associated Data	If selected, will store the values of non-bound associated data (properties created by the user) properties on alarm that do not change during execution.	X	X
Dynamic Associated Data	If selected, will store the values of dynamically bound associated data (properties created by the user) properties.	X	X

Data Filters

! INFO

The following three properties interact via logical AND, meaning an alarm must meet the criteria for all three for it to be logged in the Journal. Thus, if you supply values for all three Data Filter properties, then an alarm must match the **Filter by Alarm Source**, **Filter by Display Path**, and **Filter by Display Path or Source** properties.

Example: if a Journal has values for all three properties, and an alarm only meets the requirements for **Filter by Alarm Source** and **Filter by Display Path or Source**, but not **Filter by Display Path**, then the alarm will not be logged to the Journal.

If you want to filter on both the Display Path and Source Path, you would want to use only one of the two following approaches:

- **Filter by Alarm Source** and **Filter by Display Path**

- Only use **Filter by Display Path or Source**

It is recommended to avoid using all three properties simultaneously on the same Journal.

Property	Description	Database	Internal
Filter by Alarm Source	Only events matching the source will be stored. Multiple sources to match can be comma separated. Leave blank to store events from all sources.	X	X
Filter by Display Path	Only events matching the display path will be stored. Multiple display paths to match can be comma separated. Leave blank to store events from all display paths.	X	X
Filter by Display Path or Source	Only events matching the display path, if defined, will be stored. Multiple matches can be comma separated. If no display path is defined, only events matching the source will be stored. Leave blank to store all events.	X	X

Data Pruning

Property	Description	Database	Internal
Enable Data Pruning	Default is not enabled. If selected, data will be deleted after the specified time period as set by the Prune Age and Units below. Note that since the data is stored directly in a database, an administrator is free to manually delete data at any time.	X	X
Prune Age	The number of Prune Age Units to store data for. IE: 1 year, 5 hours, etc. The default is 1.	X	X
Prune Age Units	The default is YEAR. You can choose the unit to be MS, SEC, MIN, HOUR, DAY, WEEK, MONTH, and YEAR.	X	X

Advanced

These settings let you specify your own table names. This is especially useful when trying to use multiple alarm profiles within a single database (not common, but can happen, especially with multiple systems sharing a single database).

Property	Description	Database	Internal
Table Name	The default is <code>alarm_events</code> . The table name for the core event table.	X	
Event Data Table Name	The default is <code>alarm_event_data</code> . The table name for event data associated with alarms.	X	

Table Definitions

The alarm journal system will automatically create the necessary tables for you, and scripting functions can be used to query the system without having to know about the table structure. However, understanding the structure of the alarm journal tables can be useful for accessing the data in situations where SQL queries are more convenient.

Alarm Events (`alarm_events`)

This table stores the core data for each event that occurs. An event is a transition for an alarm between active, cleared, or acknowledged. Additionally, other events may be stored in this table that aren't directly related to an alarm, such as a system shutdown event. This table defines a primary key "id", that is used as a foreign key by the Alarm Event Data table, which stores additional information for each event.

Column Name	Data Type	Description
id	integer	A unique integer id for each event entry event.
eventid	string	The UUID of the alarm event that this individual event is related to. Each alarm event (one particular active/clear/ack cycle of a defined alarm) receives a unique id in order to distinguish it from other events from the same source.
source	string	The qualified path of the entity that generated the alarm event. See below for more information about qualified paths.
display path	string	The value set for the "Display Path" of the alarm. Generally a user defined, friendlier version of the source.
priority	integer	The priority or severity of the alarm: - 0: Diagnostic - 1: Low - 2: Medium - 3: High - 4: Critical
eventtype	integer	The type of transition represented by this event: - 0: Active - 1: Clear - 2: Acknowledgement
eventflags	integer	A numeric bitmask flag field providing additional information about the event. - Bit 0: System Event - One of the designated system events. (System Startup, System Shutdown)

Column Name	Data Type	Description
		- Bit 1: Shelved Event - The alarm was "shelved" at the time that the event occurred. Shelving alarms does not prevent execution, so if the journal is configured to store shelved events, they will be stored even if they're not sent to the notification system, or shown to users. - Bit 2: System Acknowledgement - When the "live event limit" (defined in general alarm settings) is exceeded, the system will automatically acknowledge overflow events, and the acknowledgment event will have this flag set. - Bit 3: Acknowledge Event - The event was acknowledged at the time of the event. For events that are cleared after being acknowledged. - Bit 4: Cleared Event - The event was cleared at the time of the event. For alarms that are acknowledged after being cleared.
eventtime	datetime	The time of the event.

Alarm Event Data (alarm_event_data)

This table stores the properties associated with an alarm event. The individual event is referenced through the ID column, against the alarm event table.

Column Name	Data Type	Description
id	integer	The id that corresponds to the alarm event in the alarm_events table.
propname	string	The name of the property. May be one of the common alarm properties (a configuration setting), or the name of an associated data property.
dtype	integer	The data type of the property, indicating which data column should be used: - 0: Integer - 1: Float - 2: String
intvalue/floatvalue/strvalue	integer/float (double)/string	The corresponding value columns for the property. Unused columns will receive "null" values.

Qualified Paths

A qualified path in Ignition is a path to an object, described by various annotated components. Each component has a type identifier and a value, separated by a colon (:), and each component is separated by colon-forward slash (:/). For example, an alarm is identified by `alm:Alarm Name`. It usually exists under a Tag, in which case its fuller path would be `tag:Path/To/Tag:/alm:Alarm Name`. Paths can be built up further depending on the level of specificity required by the situation.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Journal Table Component



INDUCTIVE
UNIVERSITY

Alarm Journal Component

[Watch the video](#)

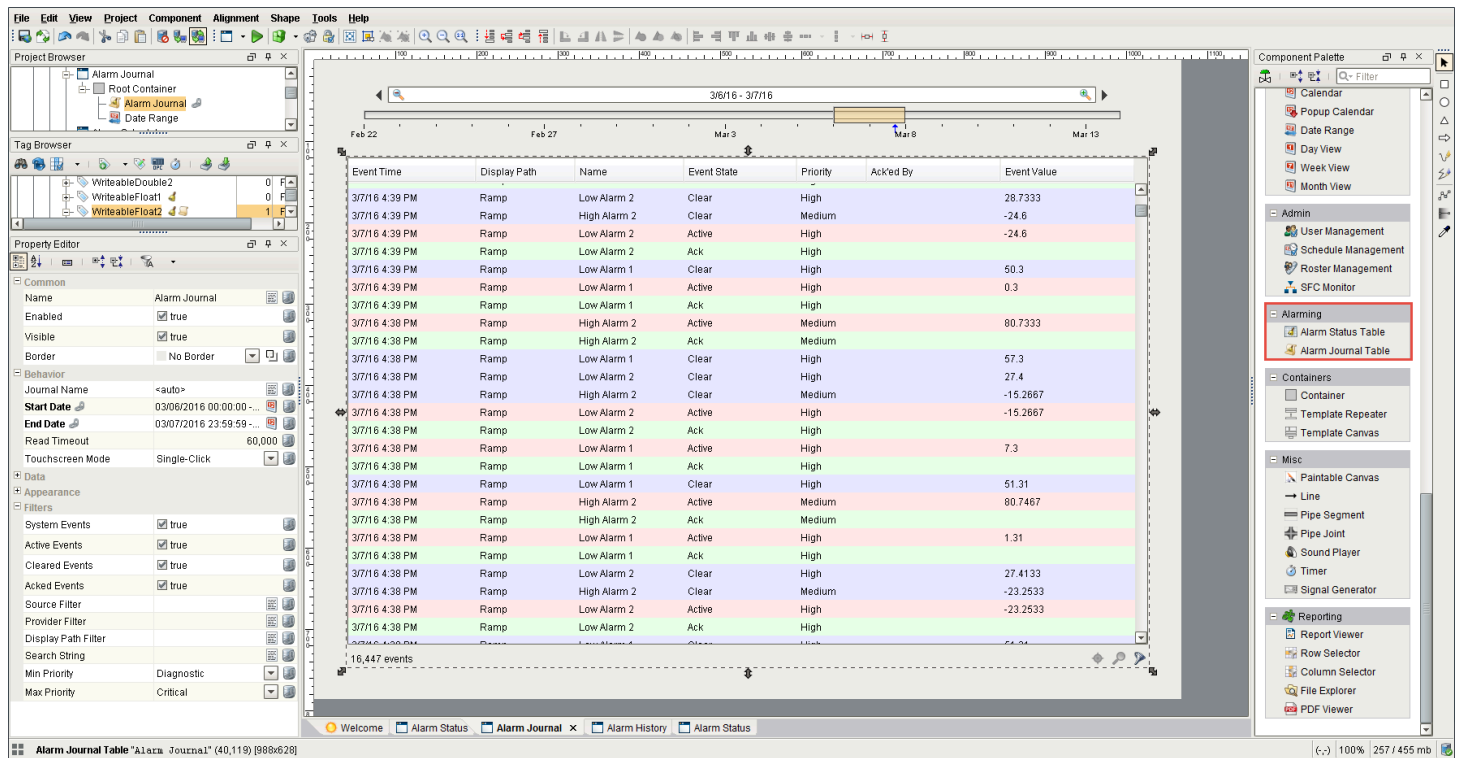
The **Alarm Journal Table** component provides a built-in view to explore alarm history that has been stored in your database. Before viewing alarm history, you must first tell Ignition to log alarms in your database by creating your [Alarm Journal Profile](#).

NOTE

To view alarm history, an Alarm Journal Profile and alarms must first be configured. To learn more, go to [Configuring Alarms](#), and the [Alarm Journal Table](#) in the Appendix.

Alarm Journal Table Component

There is a pre-built component in Ignition called the **Alarm Journal Table** that allows the database to see alarm history. In the **Component Palette**, under Alarming, drag the **Alarm Journal Table** component to your window and you'll see the history of all the alarms that occurred.



By default, the Alarm Journal Table does not filter out any of the alarms, it simply shows you all the alarms stored in the table for the time range set. When you add an Alarm Journal component to a window and have no start and end dates selected, it defaults to show the previous 8 hours of alarms and will not refresh itself until the start or end date properties change. See [Alarm Journal - Filter on Date Range](#) for more information.

The alarms are color coded so you know what each of the states represent. In Preview Mode, you can right click in the header to hide or show columns. You can also move columns by clicking them and dragging them around. You can sort by a column by left-clicking it, or sort by multiple columns by holding the **Ctrl** key while clicking the column headers. You will see small numbers next to the column header to indicate the sort order and direction (ascending vs descending).

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Alarm Journal
 - Root Container
 - Alarm Journal
 - Date Range

Tag Browser

- WritableDouble2
 - WritableFloat1
 - WritableFloat2

Property Editor

Common

Name Alarm Journal

Enabled ☒ true

Visible ☒ true

Border ☐ No Border

Behavior

Journal Name <auto>

Start Date 03/06/2016 00:00:00 ...

End Date 03/07/2016 23:59:59 ...

Read Timeout 60,000

Touchscreen Mode Single-Click

Data

Appearance

Filters

System Events ☒ true

Active Events ☒ true

Cleared Events ☒ true

Acked Events ☒ true

Source Filter

Provider Filter

Display Path Filter

Search String

Min Priority Diagnostic

Max Priority Critical

Event Time Event State Priority Ack'ed By Event Value

3/7/16 4:39 PM Clear High 28.7333

3/7/16 4:39 PM Clear Medium -24.6

3/7/16 4:39 PM Active High -24.6

3/7/16 4:39 PM Ack High 50.3

3/7/16 4:39 PM Active High 0.3

3/7/16 4:38 PM Active Medium 80.7333

3/7/16 4:38 PM Ack Medium 57.3

3/7/16 4:38 PM Clear High 27.4

3/7/16 4:38 PM Clear Medium -15.2667

3/7/16 4:38 PM Active High -15.2667

3/7/16 4:38 PM Ack High 7.3

3/7/16 4:38 PM Ack High 51.31

3/7/16 4:38 PM Clear High 80.7467

3/7/16 4:38 PM Ramp High Alarm 2 Active Medium 1.31

3/7/16 4:38 PM Ramp High Alarm 1 Active High 27.4133

3/7/16 4:38 PM Ramp Low Alarm 2 Clear Medium -23.2533

3/7/16 4:38 PM Ramp Low Alarm 2 Active High -23.2533

3/7/16 4:38 PM Ramp Low Alarm 2 Ack High 54.34

16,447 events

Auto Resize This Column

Auto Resize All Columns

Hide This Column "Event Time"

Show All Hidden Columns

☐ eventId

☐ Source Path

☒ Display Path

☒ Name

☒ Event Time

☒ Event State

☒ Event Value

☒ Priority

☐ isSystemEvent

☒ Ack'ed By

☐ Ack Notes

Reset to Default

More...

To view an alarm's **Details** and **Notes**, select an individual alarm, then click the **Magnifying**  icon in the lower right corner of the table to see all the information that is associated with the individual alarm. To close the Details and Notes tabs, click the **Magnifying** icon again or click the **Expand**  button.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Alarm Journal
 - Root Container
 - Alarm Journal
 - Date Range

Tag Browser

- WritableDouble2
 - WritableFloat1
 - WritableFloat2

Property Editor

Common

Name Alarm Journal

Enabled ☒ true

Visible ☒ true

Border ☐ No Border

Behavior

Journal Name <auto>

Start Date 03/06/2016 00:00:00 ...

End Date 03/07/2016 23:59:59 ...

Read Timeout 60,000

Touchscreen Mode Single-Click

Data

Appearance

Filters

System Events ☒ true

Active Events ☒ true

Cleared Events ☒ true

Acked Events ☒ true

Source Filter

Provider Filter

Display Path Filter

Search String

Min Priority Diagnostic

Max Priority Critical

Event Time Display Path Name Event State Priority Ack'ed By Event Value

3/7/16 4:39 PM Ramp Low Alarm 2 Clear High 28.7333

3/7/16 4:39 PM Ramp High Alarm 2 Clear Medium -24.6

3/7/16 4:39 PM Ramp Low Alarm 2 Active High -24.6

3/7/16 4:39 PM Ramp Low Alarm 2 Ack High 50.3

3/7/16 4:39 PM Ramp Low Alarm 1 Active High 0.3

3/7/16 4:38 PM Ramp High Alarm 2 Active Medium 80.7333

3/7/16 4:38 PM Ramp High Alarm 2 Ack Medium 57.3

3/7/16 4:38 PM Ramp Low Alarm 2 Clear High 27.4

3/7/16 4:38 PM Ramp High Alarm 2 Clear Medium -15.2667

3/7/16 4:38 PM Ramp Low Alarm 2 Active High -15.2667

16,447 events

Details Notes

Config Properties

Source Path prov.default/tag.gensim/Ramp/Ramp1:/alm:Low Alarm 1

Display Path Ramp

Name Low Alarm 1

Mode Equal

Priority High

Current State Cleared, Unacknowledged

Ack'ed By

Active Pipeline

Clear Pipeline

Deadband 0

Active delay (seconds) 0

Clear delay (seconds) 0

Timestamp Source System

Ack Mode Manual

Notes

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Journal - Filter on Date Range



INDUCTIVE
UNIVERSITY

Alarm Journal - Filter on Date Range

[Watch the video](#)

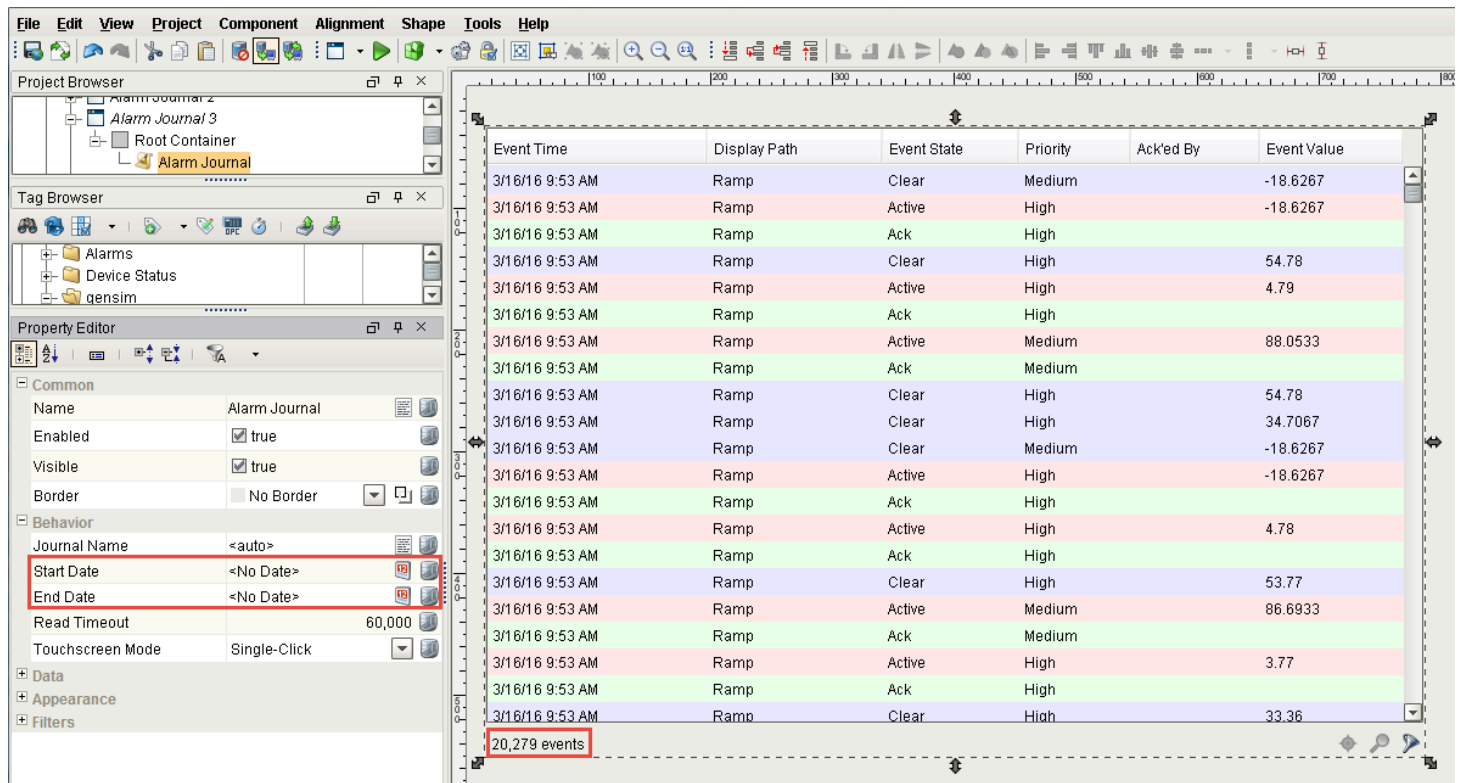
NOTE

This section assumes that an alarm journal profile and alarms are configured. To learn more, go to the [Configuring Alarms](#) and [Journal Properties and Tables](#) pages.

Alarm Journal Table

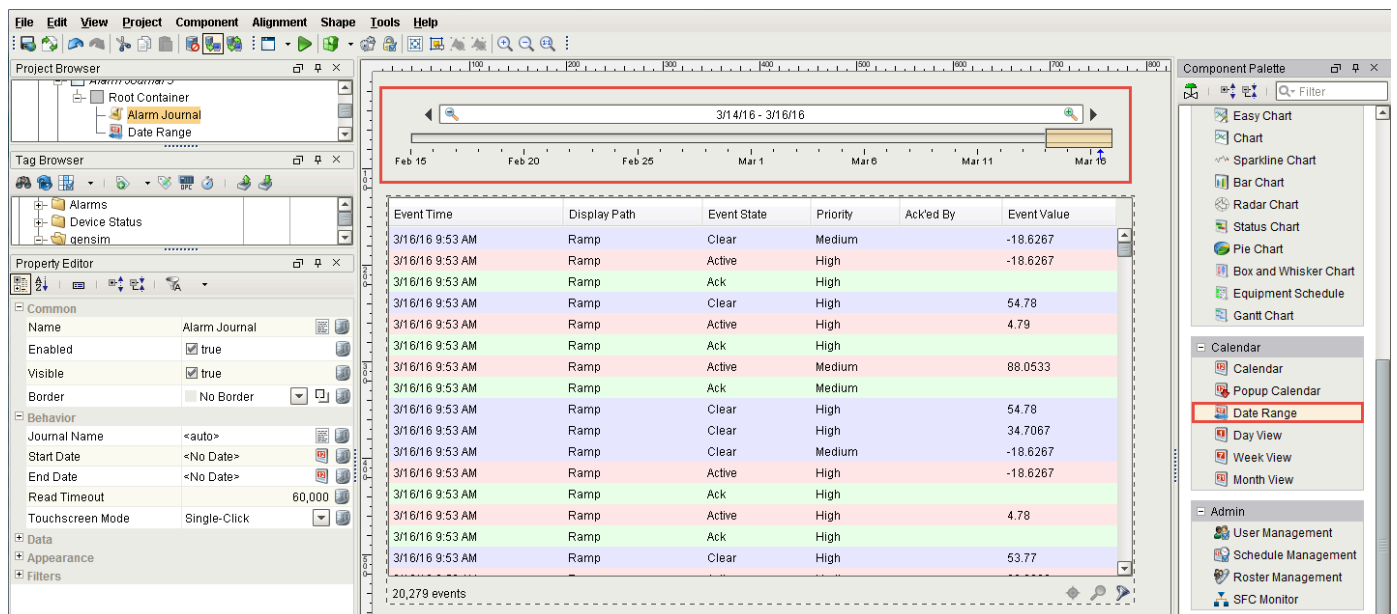
The alarm history displayed in the **Alarm Journal Table** component can be filtered by start date and end date.

When the Alarm Journal Table component is initially **placed on a window**, the component shows the last 8 hours of alarm events stored in the **Alarm Journal Profile**. You typically want to filter for alarm events in a specific time period rather than scroll through a list of events. You don't want operators to see everything, but to choose a start date and end date so they can see the alarm history for a specific time period. This is very easy to setup on the Date Range component.

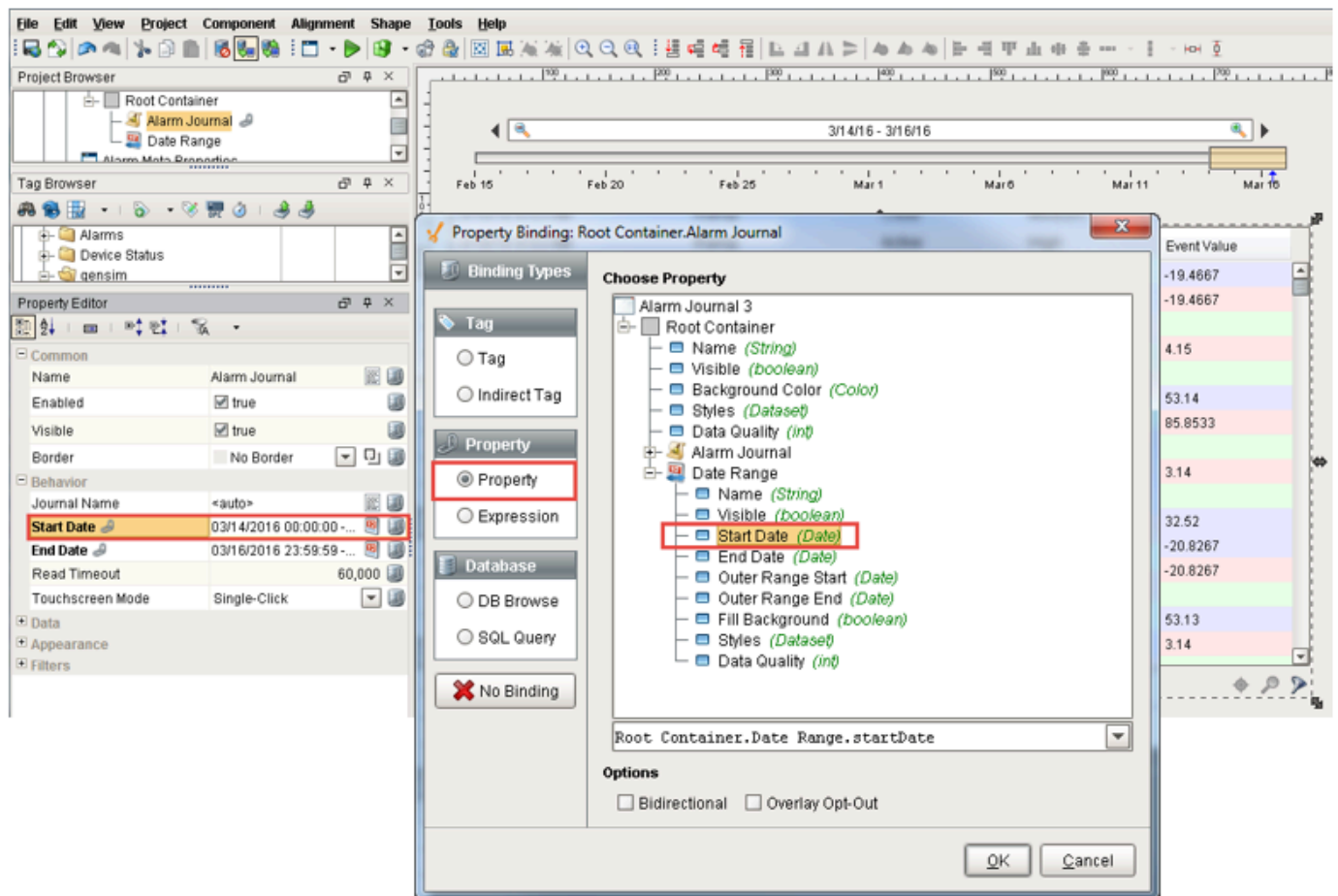


Setting up Alarm Journal Table to Filter on Date Range

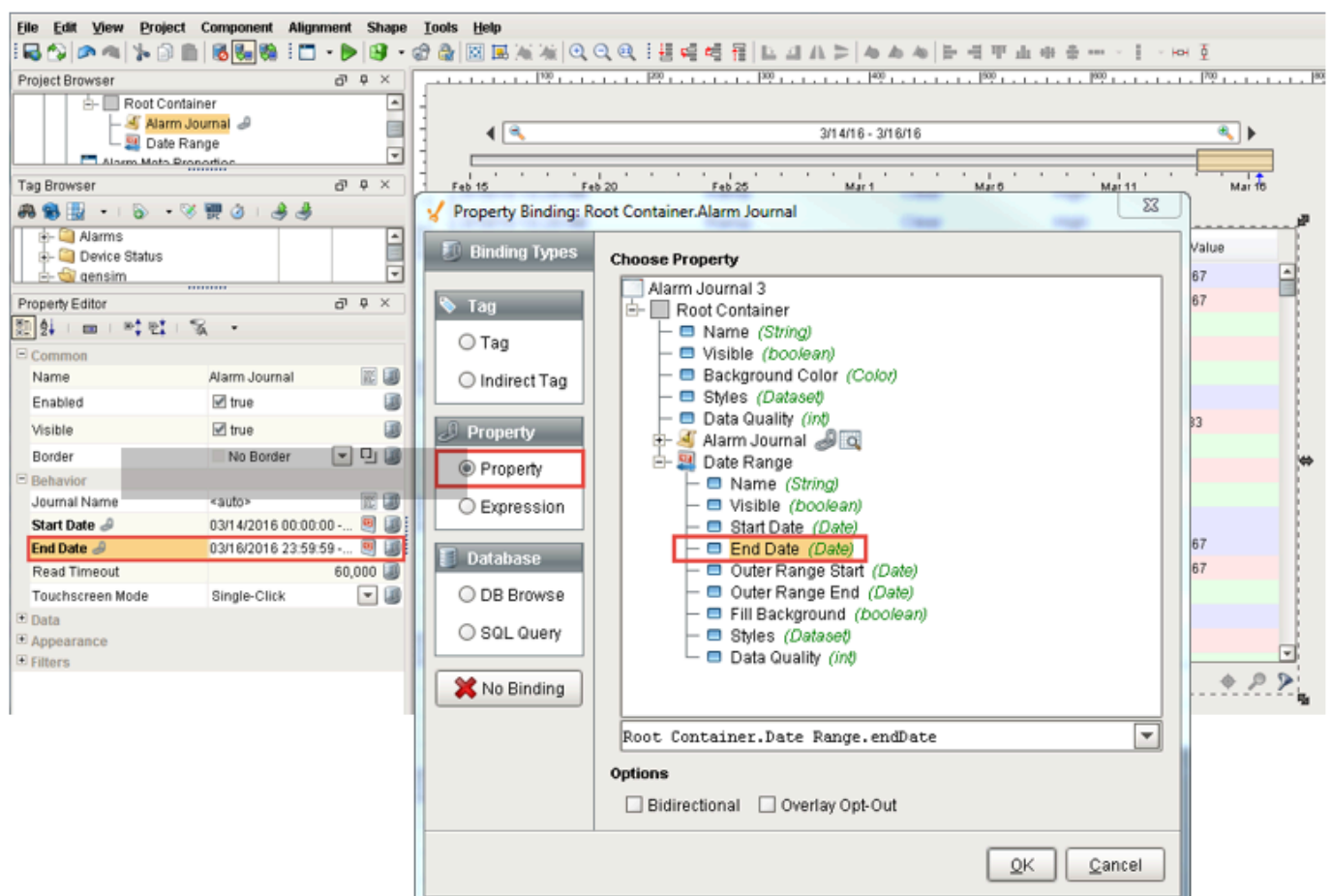
1. Ignition offers several types of date pickers under the Calendar tab of the Component Palette. This example uses the **Date Range component**. The start date and end date are built into the Date Range component so you can select the Start Date and End Date all in one place. Drag the **Date Range component** from the Component Palette to your window above the Alarm Journal Table.



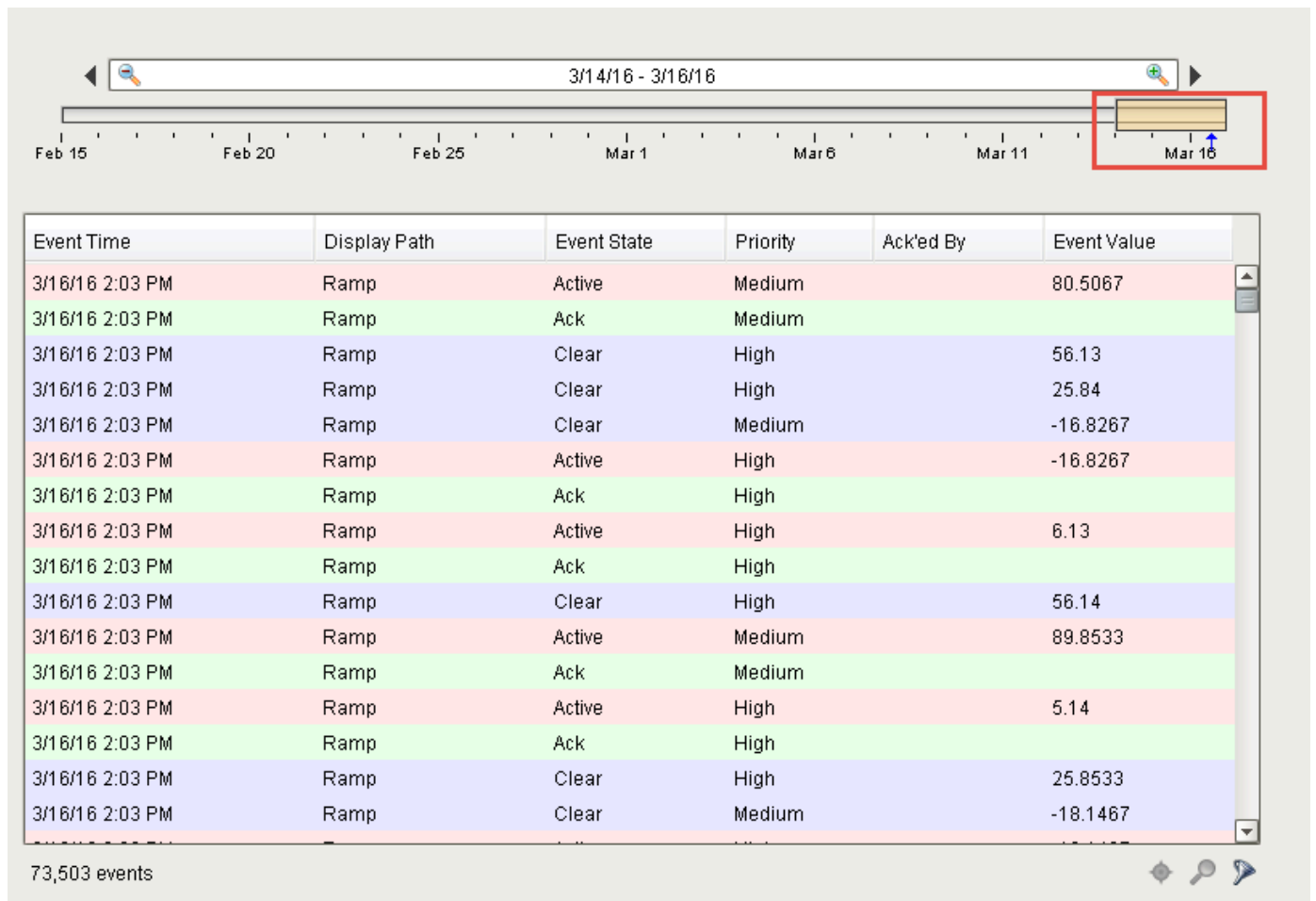
2. To link the Alarm Journal Table component to the Date Range component, select the **Alarm Journal Table** component, click the **Start Date** binding icon. Click the **Property** binding type, and select the **Start Date** for the **Date Range** component. Click **OK**.



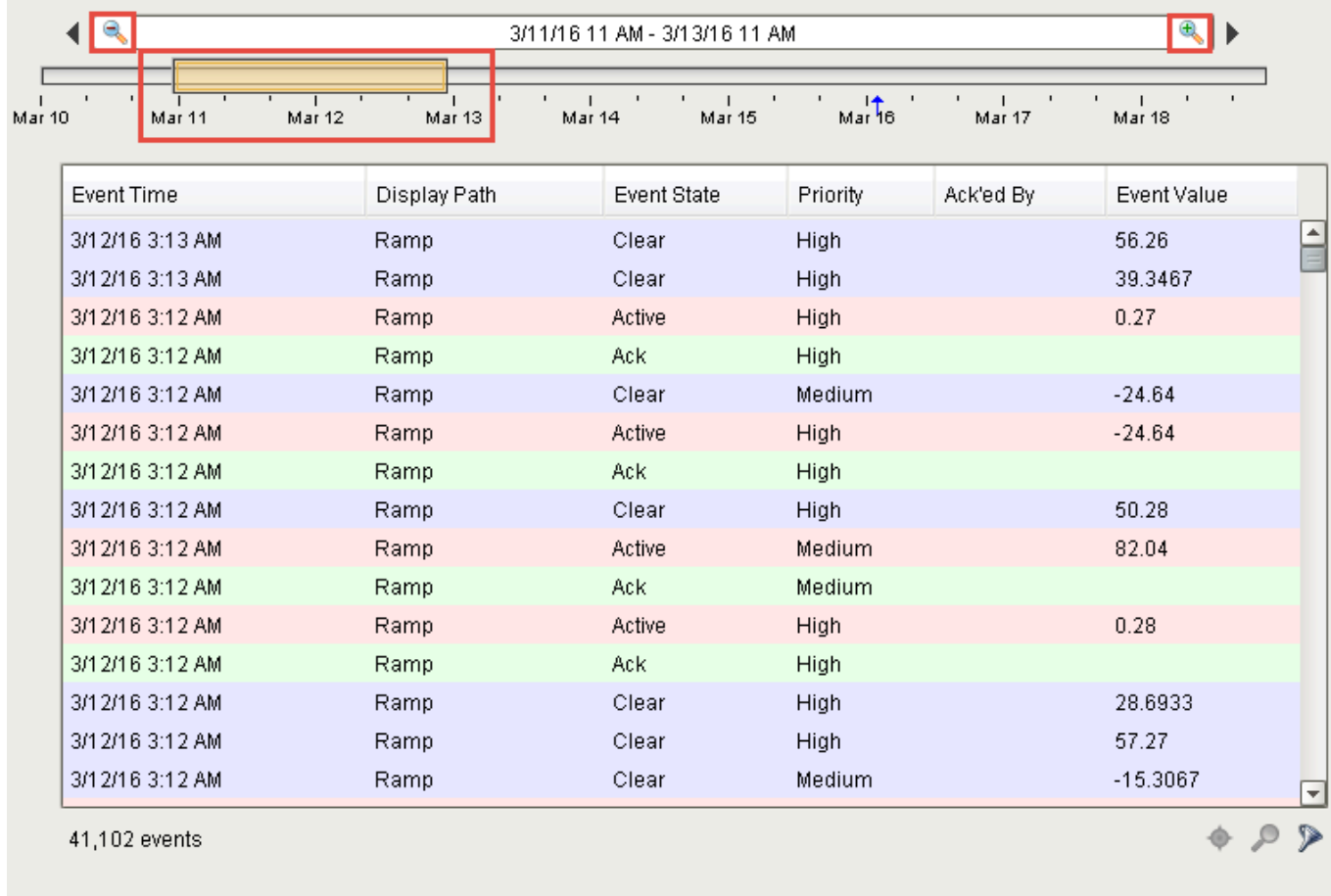
3. Now, do the same for the **End Date**. Click the **End Date** binding icon, bind it to the **Property** binding type, and select the **End Date** for the **Date Range** component. Click **OK**.



4. By linking the Alarm Journal Table and the Date Range components together, as soon as the operator changes the Start Date and End Date on the Date Range component, Ignition automatically updates the Alarm Journal Table and all the alarm events within that time period.
5. Click on **Preview Mode** and you will see alarm events within that time period.



6. You can also zoom in or out on specific time periods to see the alarm events that exist in the alarm history table.



It is very easy to filter alarm events based on time. It's important to note that the Alarm Journal Table fetches data only when the window is first opened and when the start or end date change. To see new alarm events come into the table, use the slider to change the dates.

NOTE

You can make the alarm events show as red bars in your [Date Range component](#) component if you bind the Data Density property.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Journal - General Filtering



INDUCTIVE
UNIVERSITY

**Alarm Journal - General
Filtering**

[Watch the video](#)

Alarm Journal Table Filtering

Ignition provides a host of filters to view alarm history data. Not only can you filter the Alarm Journal Table on a date range to see alarm history data within a specific time period, but there are also a number of other filters you can use as well.

In the Alarm Journal Table component you can filter on alarms in two different areas in the table; filter properties in the Property Editor and the filter icons located at the bottom of the Alarm Journal Table component.

Setting up General Filtering on the Alarm Journal Table

In the Designer, drag an Alarm Journal Table component into your window. By default, Ignition filters on four different type of alarm events: **System Events**, **Active Events**, **Cleared Events** and **Acked Events**. In the Property Editor you can see each type of alarm event.

For example, if you uncheck the Cleared Events and Acked Events, the Alarm Journal Table will show only Active and the System events. If you want to show only Active events, check Active Events and leave the other events unchecked. It's easy to configure the event filters on the type of alarms you want to see.

You can also filter on **Source Path**, **Priority**, **Display Path** and even create your own **Search String**. This example shows a basic filter displaying all the alarm events. Each color represents a different

alarm state. In Preview Mode, you can right click in the column header to show or hide columns, reorganize the columns, and resize the columns.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

Tag Browser

Property Editor

Appearance

Filters

3/3/16 - 3/4/16

Feb 22 Feb 27 Mar 3 Mar 8 Mar 13

Event Time	Display Path	Name	Event State	Priority	Ack'd By	Event Value
3/4/16 12:57 PM	Ramp	Low Alarm 1	Ack	High		
3/4/16 12:57 PM	Ramp	High Alarm 2	Active	Medium		86.84
3/4/16 12:57 PM	Ramp	High Alarm 2	Ack	Medium		
3/4/16 12:57 PM	Ramp	Low Alarm 1	Clear	High		52.88
3/4/16 12:57 PM	Machine C	Low Alarm	Clear	Critical		101
3/4/16 12:57 PM	Machine D	High Alarm	Active	Critical		101
3/4/16 12:57 PM	Machine D	High Alarm	Ack	Critical		
3/4/16 12:57 PM	Ramp	Low Alarm 2	Clear	High		32.1733
3/4/16 12:57 PM	Machine C	Low Alarm	Active	Critical		10
3/4/16 12:57 PM	Machine D	High Alarm	Clear	Critical		10
3/4/16 12:57 PM	Ramp	High Alarm 2	Clear	Medium		-21.1733
3/4/16 12:57 PM	Ramp	Low Alarm 2	Active	High		-21.1733
3/4/16 12:57 PM	Ramp	Low Alarm 2	Ack	High		
3/4/16 12:57 PM	Ramp	Low Alarm 1	Active	High		2.87
3/4/16 12:57 PM	Ramp	Low Alarm 1	Ack	High		
3/4/16 12:57 PM	Machine C	Low Alarm	Clear	Critical		125
3/4/16 12:57 PM	Machine D	High Alarm	Active	Critical		125
3/4/16 12:57 PM	Ramp	Low Alarm 1	Clear	High		52.88
3/4/16 12:57 PM	Ramp	High Alarm 2	Active	Medium		85.4933
3/4/16 12:57 PM	Ramp	High Alarm 2	Ack	Medium		
3/4/16 12:57 PM	Ramp	Low Alarm 1	Active	High		2.87
3/4/16 12:57 PM	Ramp	Low Alarm 1	Ack	High		
3/4/16 12:57 PM	Ramp	Low Alarm 2	Clear	High		32.16
3/4/16 12:56 PM	Ramp	High Alarm 2	Clear	Medium		-21.1733
3/4/16 12:56 PM	Ramp	Low Alarm 2	Active	High		-21.1733

6,576 events

There is also a way to filter alarms based on a Tag Path, Tag Provider and alarm Name. If you configure alarms in particular folders in Ignition, you can use the **Source Filter** to find alarms by Tag Path. In this example, ***Machine*** was entered using wildcards in the Source Path field of the Property Editor to find all the 'Machine' alarms in all the Machine folders.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

Tag Browser

Property Editor

Appearance

Filters

2/29/16 7 AM - 3/3/16 12 PM

Feb 28 Feb 29 Mar 1 Mar 2 Mar 3 Mar 4

Event Time	Display Path	Name	Event State	Priority
3/1/16 4:23 PM	Machine B	Low Alarm	Active	Medium
3/1/16 4:23 PM	Machine B	High Alarm	Active	Critical
3/1/16 4:23 PM	Machine B	Low Alarm	Active	Medium
3/1/16 4:23 PM	Machine B	Low Alarm	Clear	Medium
3/1/16 4:23 PM	Machine B	High Alarm	Clear	Critical
3/1/16 4:23 PM	Machine B	Low Alarm	Clear	Medium
3/1/16 4:47 PM	Machine A	Out of Range	Clear	Medium
3/1/16 4:47 PM	Machine A	Out of Range	Clear	Critical
3/1/16 4:47 PM	Machine A	Out of Range	Active	Medium
3/1/16 4:47 PM	Machine A	Out of Range	Ack	Medium
3/1/16 4:47 PM	Machine A	Out of Range	Active	Critical
3/1/16 4:47 PM	Machine A	Out of Range	Ack	Critical
3/1/16 4:47 PM	Machine A	Out of Range	Clear	Medium
3/1/16 4:47 PM	Machine A	Out of Range	Clear	Critical
3/1/16 4:47 PM	Machine A	Out of Range	Active	Medium
3/1/16 4:47 PM	Machine A	Out of Range	Ack	Medium
3/1/16 4:47 PM	Machine A	Out of Range	Active	Critical
3/1/16 4:47 PM	Machine A	Out of Range	Ack	Critical
3/1/16 4:47 PM	Machine A	Out of Range	Clear	Medium
3/1/16 4:47 PM	Machine A	Out of Range	Clear	Critical
3/1/16 4:45 PM	Machine A	Out of Range	Active	Medium
3/1/16 4:45 PM	Machine A	Out of Range	Ack	Medium
3/1/16 4:45 PM	Machine A	Out of Range	Active	Critical
3/1/16 4:45 PM	Machine A	Out of Range	Ack	Critical

203 events

You can also filter on alarms by name using the **Display Path Filter** which was setup in the [alarm configuration](#). In this example, ***Ramp*** was entered in the Display Path field to find all the alarms in the system with a 'Ramp' name.

The screenshot shows the Alarm Journal configuration interface. On the left, the 'Property Editor' is open, showing the 'Filters' section. The 'Display Path Filter' is highlighted with a red box, and its value is '*Ramp*'. The 'Search String' is empty. The 'Min Priority' is set to 'Diagnostic' and the 'Max Priority' is set to 'Critical'. The 'Date Range' is set to '3/2/16 - 3/4/16'. The 'Tag Browser' shows the 'Alarm Journal' tag. The 'Project Browser' shows the 'Alarm Journal' project. The main area displays a table of events.

Event Time	Display Path	Event State	Priority	Ack'ed By	Event Value
3/4/16 10:43 AM	Ramp	Ack	High		
3/4/16 10:43 AM	Ramp	Clear	High		56.66
3/4/16 10:43 AM	Ramp	Active	Medium		83.88
3/4/16 10:43 AM	Ramp	Ack	Medium		
3/4/16 10:43 AM	Ramp	Active	High		1.66
3/4/16 10:43 AM	Ramp	Ack	High		
3/4/16 10:43 AM	Ramp	Clear	High		26.5467
3/4/16 10:43 AM	Ramp	Clear	Medium		-24.12
3/4/16 10:43 AM	Ramp	Active	High		-24.12
3/4/16 10:43 AM	Ramp	Ack	High		
3/4/16 10:43 AM	Ramp	Clear	High		50.66
3/4/16 10:43 AM	Ramp	Active	High		0.66
3/4/16 10:43 AM	Ramp	Ack	High		
3/4/16 10:43 AM	Ramp	Active	Medium		82.5733
3/4/16 10:43 AM	Ramp	Ack	Medium		
3/4/16 10:43 AM	Ramp	Clear	High		50.66
3/4/16 10:43 AM	Ramp	Clear	High		25.2267
3/4/16 10:43 AM	Ramp	Clear	Medium		-14.7733
3/4/16 10:43 AM	Ramp	Active	High		-14.7733
3/4/16 10:43 AM	Ramp	Ack	High		
3/4/16 10:43 AM	Ramp	Active	High		7.67
3/4/16 10:43 AM	Ramp	Ack	High		
3/4/16 10:43 AM	Ramp	Clear	High		57.67
3/4/16 10:43 AM	Ramp	Active	Medium		81.2267
3/4/16 10:43 AM	Ramp	Ack	Medium		

760 events

You can also create your own **Search String** to find specific alarms, events and conditions. In this example, ***Machine B*** was entered in the Search String field to find all Machine B alarms in the system.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Date Range
- Alarm Journal
 - Root Container
 - Alarm Journal

Tag Browser

- WriteableFloat1
- WriteableFloat2

Property Editor

Common

Name Alarm Journal

Enabled ☒ true

Visible ☒ true

Border No Border

Behavior

Journal Name <auto>

Start Date 03/02/2016 00:00:00 -...

End Date 03/04/2016 23:59:59 -...

Read Timeout 60,000

Touchscreen Mode Single-Click

Data

Appearance

Filters

System Events ☒ true

Active Events ☒ true

Cleared Events ☒ true

Acked Events ☒ true

Source Filter

Provider Filter

Display Path Filter

Search String *Machine B*

Min Priority Diagnostic

Max Priority Critical

3/2/16 - 3/4/16

Event Time	Display Path	Event State	Priority	Ack'd By	Event Value
3/4/16 10:38 AM	Machine B	Clear	Medium		100
3/4/16 9:38 AM	Machine B	Active	Medium		0
3/4/16 9:38 AM	Machine B	Active	Medium		0
3/4/16 9:38 AM	Machine B	Clear	Medium		0
3/4/16 8:34 AM	Machine B	Clear	Medium		99
3/4/16 8:34 AM	Machine B	Active	Medium		0
3/4/16 8:33 AM	Machine B	Clear	Medium		95
3/4/16 8:32 AM	Machine B	Active	Medium		0
3/4/16 8:32 AM	Machine B	Active	Medium		0
3/4/16 8:32 AM	Machine B	Clear	Medium		0
3/3/16 4:59 PM	Machine B	Clear	Medium		95

11 events

Lastly, you can filter alarms based on **Priority**. Typically, when you open the Alarm Journal Table, it displays every single alarm on all the priorities from Diagnostic to Critical. If all you want to do is display Critical alarms, select 'Critical' from the **Min Priority** field and 'Critical' from the **Max Priority** field. Now, only the Critical alarms will be displayed in your table.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Date Range
- Alarm Journal
 - Root Container
 - Alarm Journal

Tag Browser

- WriteableFloat1
- WriteableFloat2

Property Editor

Common

Name Alarm Journal

Enabled ☒ true

Visible ☒ true

Border No Border

Behavior

Journal Name <auto>

Start Date 03/03/2016 00:00:00 -...

End Date 03/04/2016 23:59:59 -...

Read Timeout 60,000

Touchscreen Mode Single-Click

Data

Appearance

Filters

System Events ☒ true

Active Events ☒ true

Cleared Events ☒ true

Acked Events ☒ true

Source Filter

Provider Filter

Display Path Filter

Search String

Min Priority Critical

Max Priority Critical

3/3/16 - 3/4/16

Event Time	Display Path	Event State	Priority	Ack'd By	Event Value
3/4/16 11:27 AM	Machine C	Clear	Critical		145
3/4/16 11:27 AM	Machine D	Active	Critical		145
3/4/16 11:27 AM	Machine C	Active	Critical		10
3/4/16 11:27 AM	Machine D	Clear	Critical		10
3/4/16 11:27 AM	Machine C	Clear	Critical		109
3/4/16 11:27 AM	Machine D	Active	Critical		109
3/4/16 11:27 AM	Machine C	Active	Critical		8
3/4/16 11:27 AM	Machine D	Clear	Critical		8
3/4/16 11:27 AM	Machine C	Clear	Critical		1,253
3/4/16 11:27 AM	Machine D	Active	Critical		1,253
3/4/16 11:27 AM	Machine C	Active	Critical		3
3/4/16 11:27 AM	Machine D	Clear	Critical		3
3/4/16 11:26 AM	Machine D	Active	Critical		122
3/4/16 11:25 AM	Machine D	Active	Critical		122
3/4/16 11:25 AM	Machine D	Ack	Critical		
3/4/16 11:25 AM	Machine D	Clear	Critical		15
3/4/16 11:24 AM	Machine D	Active	Critical		120
3/4/16 11:24 AM	Machine D	Ack	Critical		
3/4/16 11:24 AM	Machine D	Clear	Critical		10
3/4/16 11:24 AM	Machine D	Active	Critical		91
3/4/16 11:24 AM	Machine D	Ack	Critical		
3/4/16 10:39 AM	Machine D	Clear	Critical		10
3/4/16 10:39 AM	Machine D	Active	Critical		119
3/4/16 10:39 AM	Machine D	Ack	Critical		
3/4/16 10:39 AM	Machine D	Clear	Critical		2

45 events

All of the Alarm Journal Filter properties are bindable so you can design your windows and let the operator choose and filter the alarm data based on how they want to display it.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Journal - Filter on Associated Data



INDUCTIVE
UNIVERSITY

Alarm Journal - Filter on Associated Data

[Watch the video](#)



AN ALARM JOURNAL PROFILE, ALARM, AND ASSOCIATED DATA MUST CONFIGURED

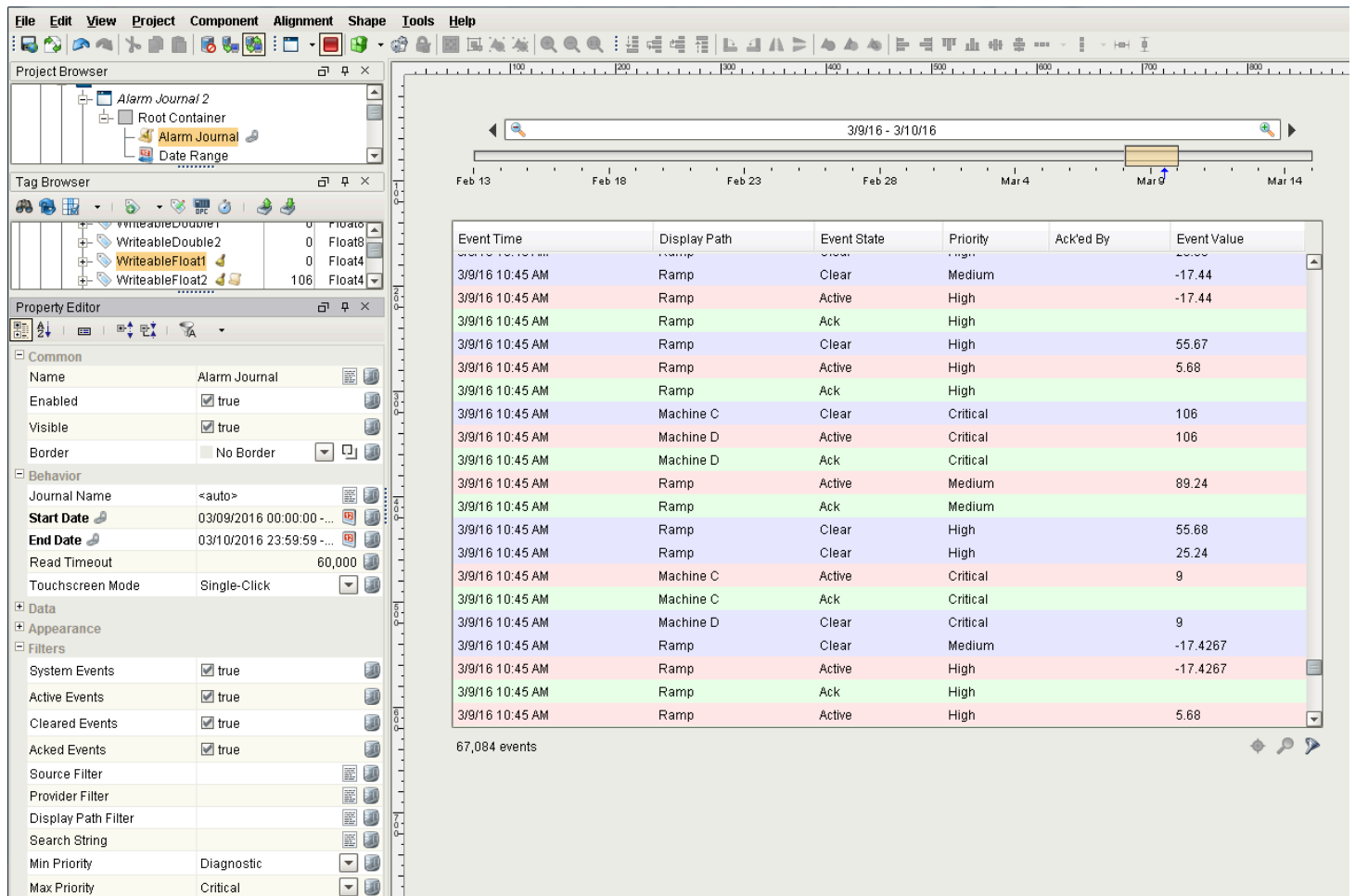
This section assumes that an Alarm Journal Profile, alarms, and associated data be configured.

To learn more, go to the [Configuring Alarms](#), [Alarm Associated Data](#), and [Journal Properties and Tables](#) pages.

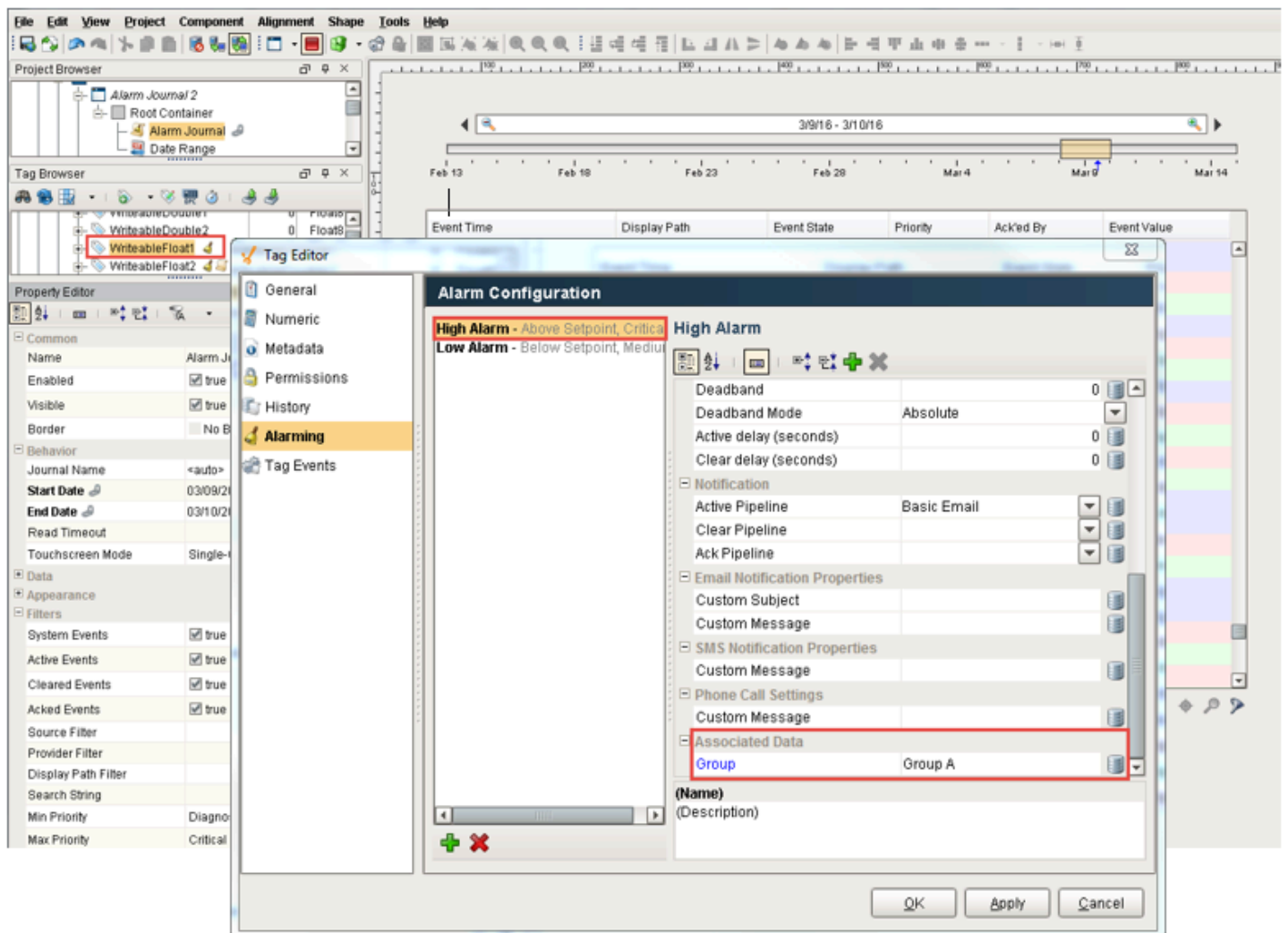
Just like the Alarm Status Table you can filter the Alarm Journal Table based on associated data that you added to your alarms.

How to set up the Alarm Journal to Filter on Associated Data

Once associated data is added to an alarm, you can easily filter for specific alarms in the [Alarm Journal Table](#) component. Typically, the Alarm Journal Table shows you the complete history of all alarms within a specific time period using the [Date Range](#) component.



In this example, a **WriteableFloat1** tag was used and added to an alarm. You can see that two alarms were configured for 'WriteableFloat1,' but only one has associated data. Associated data was added to the **High Alarm** called **Group** and was given a static value of **Group A**. You can filter on what is shown in the alarm list based on the associated data. You can search for all the Group A's or whatever associated data that was added to your alarm.



Once your associated data is configured in your alarm, the next thing you need to do is setup your script to filter for all the alarms in your history that have the associated data named 'Group' with a value of 'Group A'. Instead of using the properties in the Property Editor, go to the **Scripting area** and use an **Extension Function**.

1. Right-click on your Alarm Journal Table component and select **Scripting**. The component Scripting window will appear.
2. Under the **Extension Functions** folder, click **filterAlarm**.
3. Check **Enabled** and enter the following code at the bottom of the **filterAlarm** script.

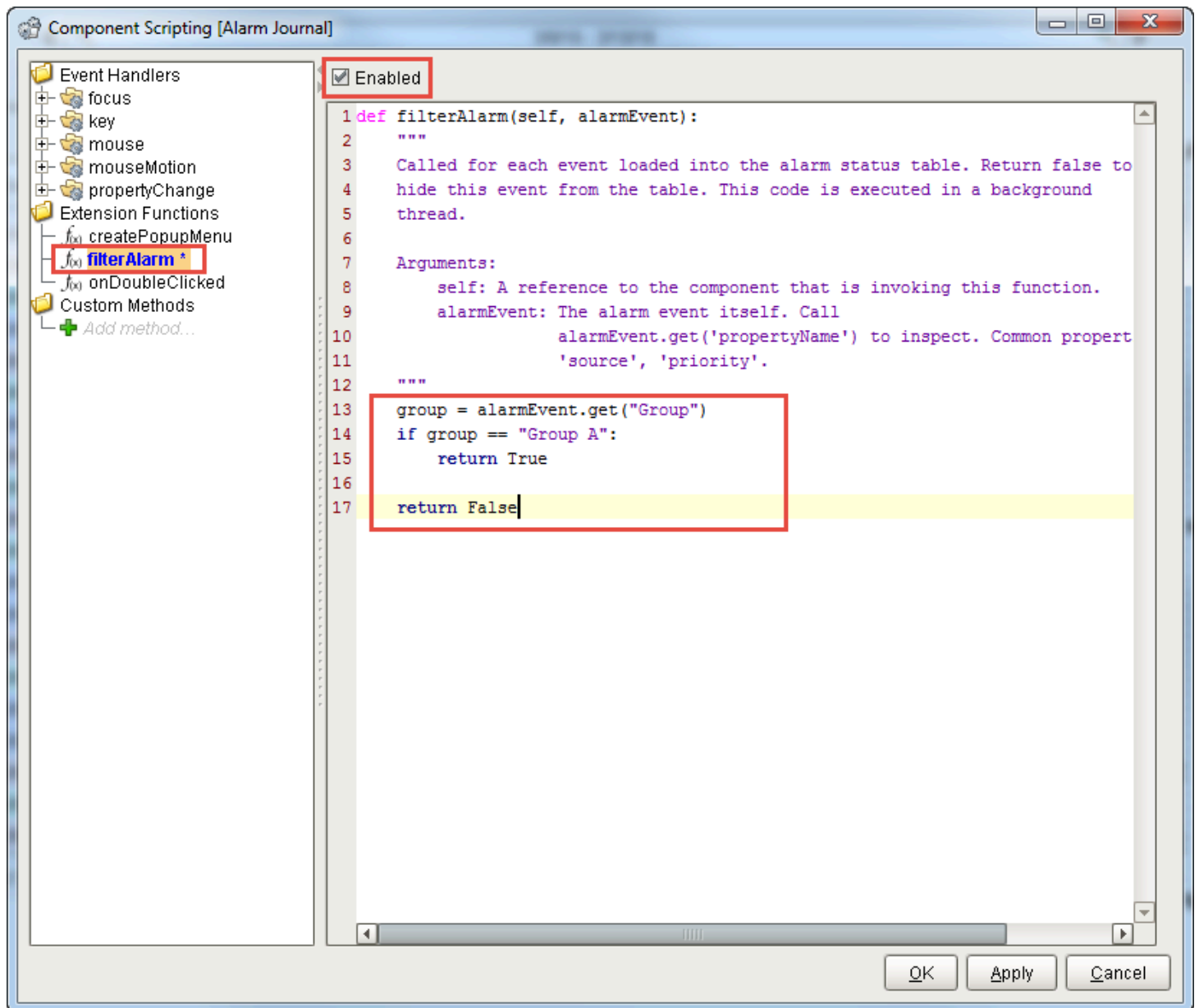
filterAlarm on associated data

```
group = alarmEvent.get("Group")
if group == "Group A":
    return True
return False
```

For every alarm matching 'Group A', the 'filterAlarm' script will return 'True,' and show the alarm in the table. If the associated data does not match 'Group A', it will return 'False,' and the alarm will not show up in the table.

Scripting allows you to use the 'filterAlarm' function for associated data. This script is only going to show you the alarms matching the associated data. You can add other filters as well depending on your requirements.

This example shows the updated 'filterAlarm' script. Press **OK**.



By default, the script is not going to automatically use your new filter until the data is updated. This happens automatically when a window is opened, but to trigger the associated data filter immediately, change the **Start Date** or **End Date**. The easiest way to change the start and end dates is by moving the Date Range slider if you have one already bound, as shown in the example below.

Click **Preview Mode**, and move the **Date Range** slider to initiate the associated data filter.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Alarm Journal 2
 - Root Container
 - Alarm Journal
 - Date Range

Tag Browser

- WriteableDouble1
- WriteableDouble2
- WriteableFloat1
- WriteableFloat2
- WriteableInteger1

Property Editor

Common

Name Alarm Journal

Enabled ☒ true

Visible ☒ true

Border No Border

Behavior

Journal Name <auto>

Start Date 03/08/2016 00:00:00 -...

End Date 03/10/2016 23:59:59 -...

Read Timeout 60,000

Touchscreen Mode Single-Click

Data

Appearance

Filters

3/8/16 - 3/10/16

Feb 9 Feb 14 Feb 19 Feb 24 Feb 29 Mar 5 Mar 10

Event Time	Display Path	Event State	Priority	Ack'd By	Event Value
3/10/16 4:40 PM	Ramp	Ack	Medium		
3/10/16 4:40 PM	Machine B	Active	Critical		110
3/10/16 4:40 PM	Ramp	Clear	High		55.01
3/10/16 4:40 PM	Ramp	Clear	High		25.68
3/10/16 4:40 PM	Ramp	Active	High		5.01
3/10/16 4:40 PM	Ramp	Ack	High		
3/10/16 4:40 PM	Ramp	Clear	Medium		-18.32
3/10/16 4:40 PM	Ramp	Active	High		-18.32
3/10/16 4:40 PM	Ramp	Ack	High		
3/10/16 4:40 PM	Ramp	Clear	High		55.01
3/10/16 4:40 PM	Machine B	Clear	Critical		12
3/10/16 4:40 PM	Ramp	Active	Medium		88.3467
3/10/16 4:40 PM	Ramp	Ack	Medium		
3/10/16 4:40 PM	Machine B	Active	Critical		166
3/10/16 4:40 PM	Machine B	Clear	Medium		166

1,519 events

Now, the Alarm Journal Table will show only alarms for '**Group A**'. So it is very easy to filter on associated data. The associated data is outside the folder structure and outside the Display Path. It allows you to filter any way you want so you can look at alarms in a particular area of the plant or for a certain set of alarms.

3/8/16 - 3/10/16

Feb 9 Feb 14 Feb 19 Feb 24 Feb 29 Mar 5 Mar 10

Event Time	Display Path	Event State	Priority	Ack'd By	Event Value
3/10/16 4:40 PM	Machine B	Active	Critical		189
3/10/16 4:40 PM	Machine B	Clear	Critical		97
3/10/16 4:40 PM	Machine B	Active	Critical		18,585
3/10/16 4:40 PM	Machine B	Clear	Critical		85
3/10/16 4:40 PM	Machine B	Active	Critical		110
3/10/16 4:40 PM	Machine B	Clear	Critical		12
3/10/16 4:40 PM	Machine B	Active	Critical		166
3/10/16 4:39 PM	Machine B	Clear	Critical		0
3/10/16 4:33 PM	Machine B	Active	Critical		101
3/10/16 4:33 PM	Machine B	Clear	Critical		99
3/10/16 4:28 PM	Machine B	Active	Critical		255
3/10/16 4:24 PM	Machine B	Clear	Critical		0
3/10/16 4:24 PM	Machine B	Active	Critical		200

13 events

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Journal - Row Styles

Row Styles

The Alarm Journal Table component allows you customize row styles for different states of alarm history.

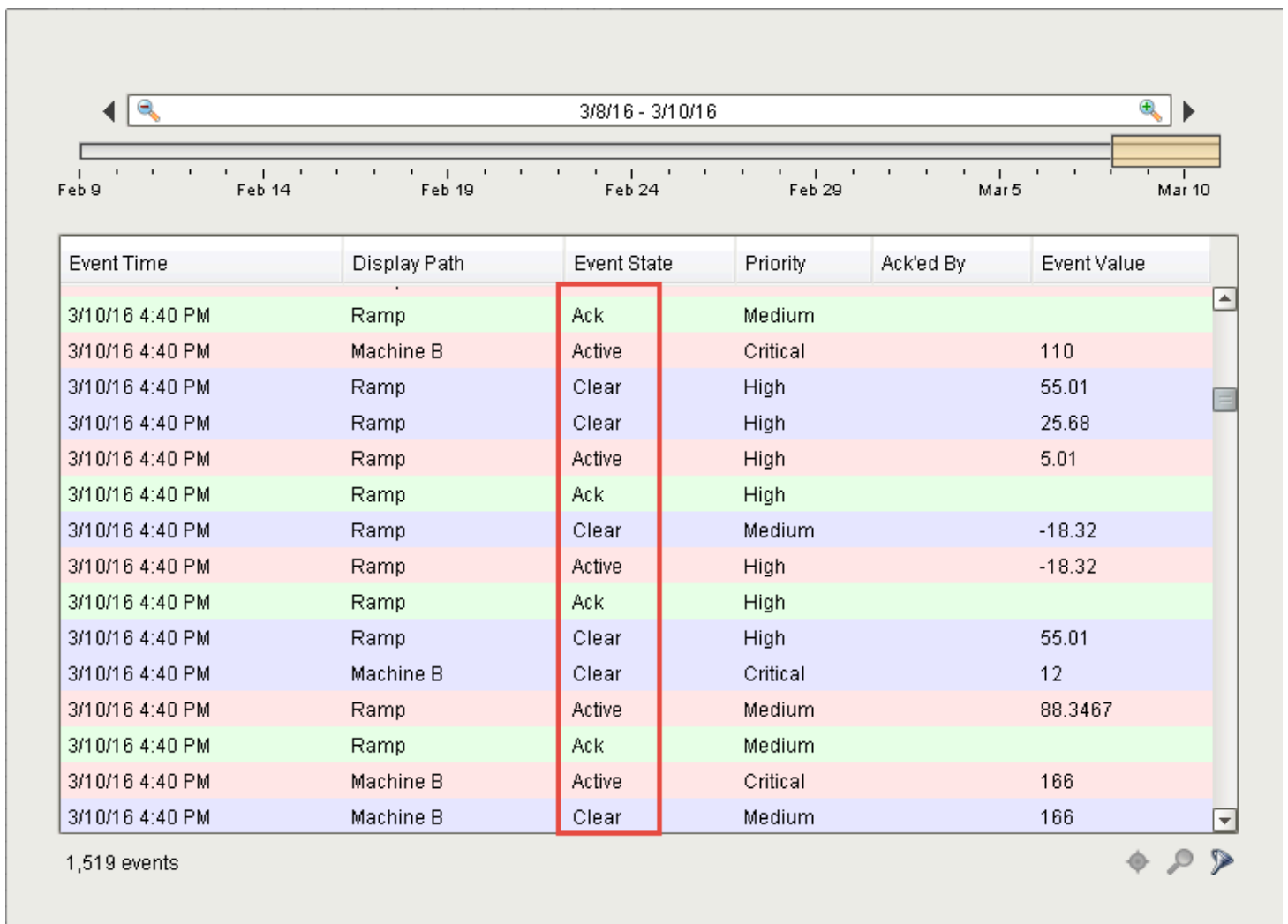


INDUCTIVE
UNIVERSITY

Alarm Journal - Row Styles

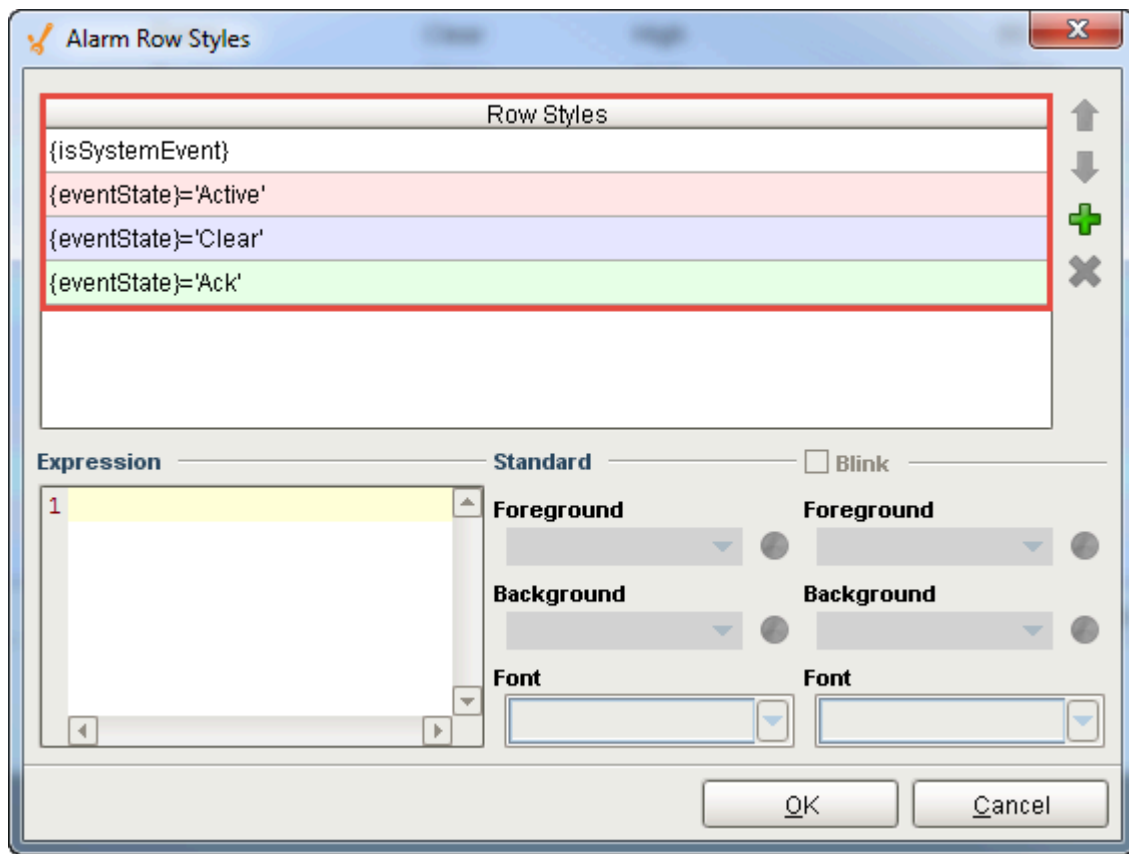
[Watch the video](#)

Just like the Alarm Status Table, the Alarm Journal Table comes with a particular set of colors associated with each of the alarm states as shown in this example. The Alarm Journal Table has a built-in feature to change the colors of each individual row based on the state of the alarm. You can change each of these colors for each of the alarm states in the **Alarm Rows Styles Customizer**.



Setting up Row Styles for Different Alarm States

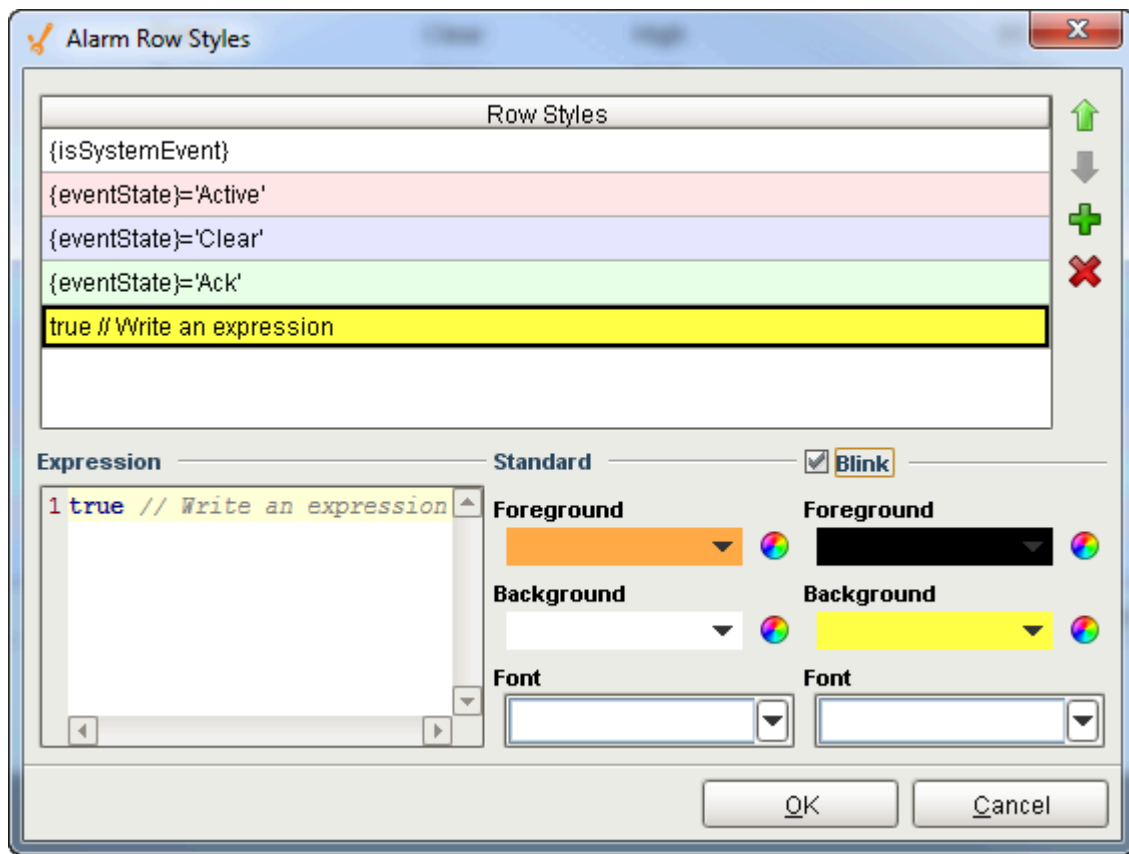
Right-click on the **Alarm Journal Table Component** and scroll down to **Customizers > Alarm Row Styles**. The Alarm Row Styles window will open. You can see the default styles, colors and states. Each style has an expression, a color, and if you want it to blink or not. In this example, the 'isSystemEvent' is white, and the event states for 'Active' is red, 'Clear' is blue, and 'Ack' is green.



You can add more row styles by clicking on the green Plus  icon or modify the expressions that currently exist.

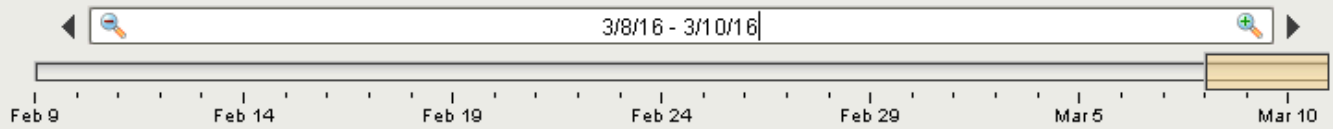
Row Styles gives you a head start for building your new expression. You can use Priority, State, Display Path, Active Time, Cleared Time, etc., in your expression. You can do any evaluations you want and use any parts of the alarm configuration for building your expression.

Alarms are evaluated in sequential order, from top to bottom. The first alarm that returns **True** is the style that will be used. If the alarm returns **False**, then that style will not be used.



Once the changes are made to row styles, you can see that all the red alarms are Active alarms, all the Cleared alarms are blue, and all the Ack alarms are green. So you can see how easy it is to add more alarm states, and change the colors of each of the individual rows in Alarm Journal Table using the Alarm Row Style Customizer.

Refer to the [Alarm Event Properties Reference](#) section to learn more about alarm properties and how they are used in the Alarm Row Style Customizer.



Event Time	Display Path	Event State	Priority	Ack'd By	Event Value
3/10/16 4:40 PM	Ramp	Ack	Medium		
3/10/16 4:40 PM	Machine B	Active	Critical		110
3/10/16 4:40 PM	Ramp	Clear	High		55.01
3/10/16 4:40 PM	Ramp	Clear	High		25.68
3/10/16 4:40 PM	Ramp	Active	High		5.01
3/10/16 4:40 PM	Ramp	Ack	High		
3/10/16 4:40 PM	Ramp	Clear	Medium		-18.32
3/10/16 4:40 PM	Ramp	Active	High		-18.32
3/10/16 4:40 PM	Ramp	Ack	High		
3/10/16 4:40 PM	Ramp	Clear	High		55.01
3/10/16 4:40 PM	Machine B	Clear	Critical		12
3/10/16 4:40 PM	Ramp	Active	Medium		88.3467
3/10/16 4:40 PM	Ramp	Ack	Medium		
3/10/16 4:40 PM	Machine B	Active	Critical		166
3/10/16 4:40 PM	Machine B	Clear	Medium		166

1,519 events



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Journal - Focusing on Alarms



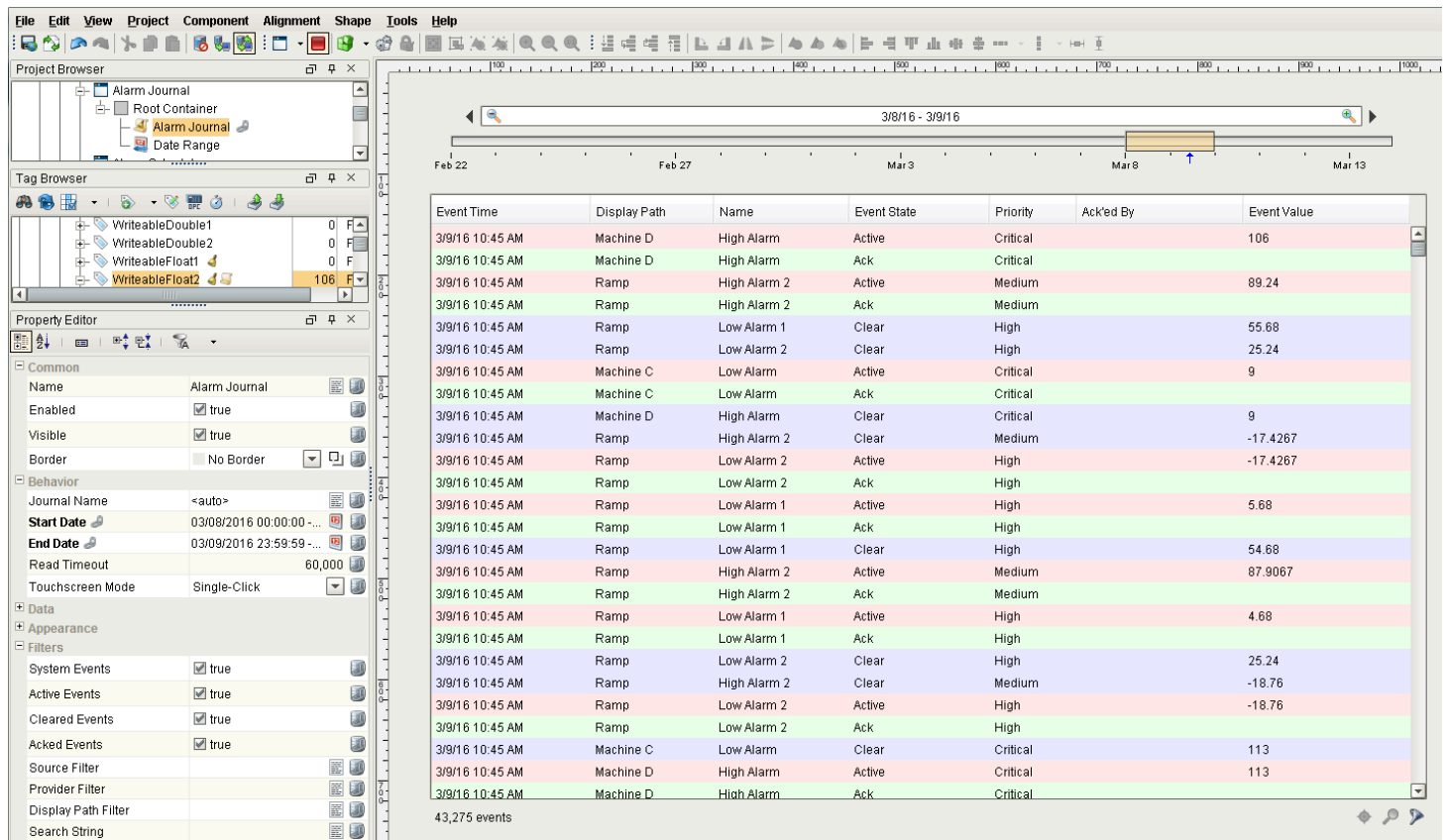
INDUCTIVE
UNIVERSITY

**Alarm Journal - Focusing on
Alarms**

[Watch the video](#)

Focusing on Alarms

The Alarm Journal Table can focus on a specific events in the alarm list. Focusing will filter out content in the table, allowing you to view all instances of a certain alarm. Additionally, alarms can be focused by an Event Id, which will show all values for a single alarm instance, such as when the alarm became active. In the following example, the Alarm Journal Table shows you a complete history of all the alarms within a specific time period using the Data Range component. You can see a lot of Machine C and Machine D alarms including quite a few Ramp alarms.

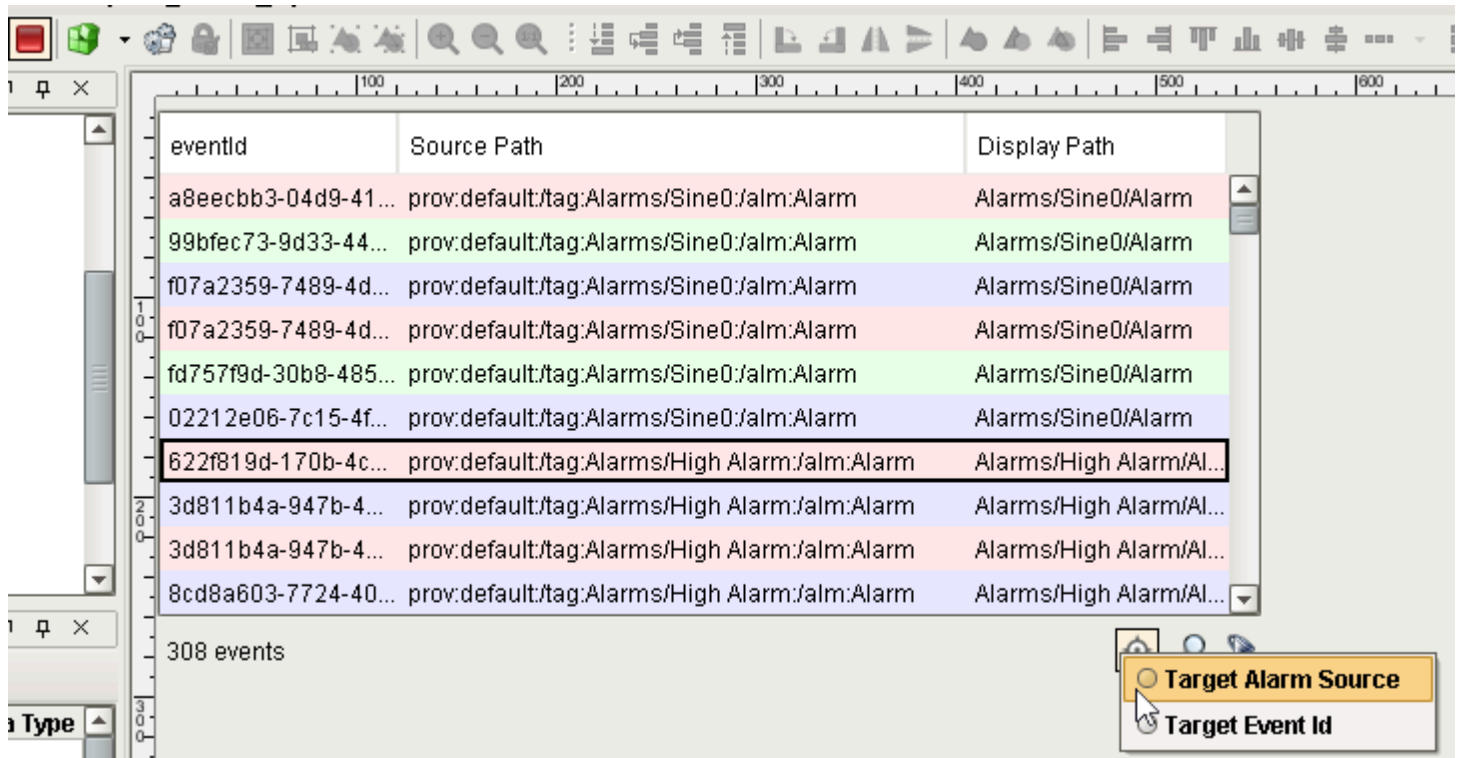


Using the Focus Feature

The **Focus** feature allows you to focus in on one specific alarm to view the history for that alarm.

Target Alarm Source

For example, if you want to see alarms from just a single Source Path, select any alarm event in the table, and click on the **Focus**  icon. This allows you to focus on the Target Alarm Source (the Source Path of the alarm) or the Event Id (a single instance of an alarm). Click on the Target Alarm Source option.



In this example, we are focusing on an alarm at source path `prov.default:/tag:Alarms/High Alarm:/alm:Alarm`. Once you click on the Target Alarm Source option you will only see events from the selected source path, including any changes to the state, as well as acknowledgments.

eventId	Source Path	Display Path
622f819d-170b-4cc...	prov.default:/tag:Alarms/High Alarm:/alm:Alarm	Alarms/High Alarm/Al...
3d811b4a-947b-48...	prov.default:/tag:Alarms/High Alarm:/alm:Alarm	Alarms/High Alarm/Al...
3d811b4a-947b-48...	prov.default:/tag:Alarms/High Alarm:/alm:Alarm	Alarms/High Alarm/Al...
8cd8a603-7724-40...	prov.default:/tag:Alarms/High Alarm:/alm:Alarm	Alarms/High Alarm/Al...
8cd8a603-7724-40...	prov.default:/tag:Alarms/High Alarm:/alm:Alarm	Alarms/High Alarm/Al...
105f40f2-ae7a-43a...	prov.default:/tag:Alarms/High Alarm:/alm:Alarm	Alarms/High Alarm/Al...

6 events

You can then remove the Focus by clicking on the **Focus**  icon again.

Target Event Id

You can also focus on a specific alarm event that already happened by selecting the event and click the **Target Event Id**.

The screenshot shows the Alarm Journal interface with a list of 7,507 events. The 'Event Time' column is highlighted, and the 'Target Alarm Source' and 'Target Event Id' buttons are visible in the bottom right corner.

Event Time	Display Path	Name	Event State	Priority	Ack'd By	Event Value
3/9/16 10:36 AM	Machine C	Low Alarm	Active	Critical		1
3/9/16 10:36 AM	Machine C	Low Alarm	Ack	Critical		
3/9/16 10:36 AM	Machine D	High Alarm	Clear	Critical		1
3/9/16 10:36 AM	Ramp	High Alarm 2	Clear	Medium		-23.44
3/9/16 10:36 AM	Ramp	Low Alarm 2	Active	High		-23.44
3/9/16 10:36 AM	Ramp	Low Alarm 2	Ack	High		
3/9/16 10:36 AM	Ramp	Low Alarm 1	Clear	High		51.17
3/9/16 10:36 AM	Machine C	Low Alarm	Clear	Critical		115
3/9/16 10:36 AM	Machine D	High Alarm	Active	Critical		115
3/9/16 10:36 AM	Machine D	High Alarm	Ack	Critical		
3/9/16 10:36 AM	Ramp	Low Alarm 1	Active	High		1.17
3/9/16 10:36 AM	Ramp	Low Alarm 1	Ack	High		
3/9/16 10:36 AM	Machine C	Low Alarm	Active	Critical		3
3/9/16 10:36 AM	Machine C	Low Alarm	Ack	Critical		
3/9/16 10:36 AM	Machine D	High Alarm	Clear	Critical		3
3/9/16 10:36 AM	Ramp	High Alarm 2	Active	Medium		83.2267
3/9/16 10:36 AM	Ramp	High Alarm 2	Ack	Medium		
3/9/16 10:36 AM	Ramp	Low Alarm 1	Clear	High		51.17
3/9/16 10:36 AM	Ramp	Low Alarm 2	Clear	High		29.8933
3/9/16 10:36 AM	Machine C	Low Alarm	Clear	Critical		112
3/9/16 10:36 AM	Machine D	High Alarm	Active	Critical		112
3/9/16 10:36 AM	Machine D	High Alarm	Ack	Critical		
3/9/16 10:36 AM	Ramp	High Alarm 2	Clear	Medium		-23.4267
3/9/16 10:36 AM	Ramp	Low Alarm 2	Active	High		-23.4267
3/9/16 10:36 AM	Ramp	Low Alarm 2	Ack	High		
3/9/16 10:36 AM	Ramp	Low Alarm 1	Active	High		1.18

7,507 events

Target Alarm Source
Target Event Id

Focusing on the Target Event Id will show you all the alarms occurring for that specific alarm event. You can see when the alarm went Active, when it was Cleared, and when it was Acknowledged. Click the **Focus** icon to remove the Focus and return to the complete list of all alarms.

The screenshot shows the Alarm Journal interface with a filtered list of 3 events. The 'Event Time' column is highlighted, and the 'Focus' icon is visible in the bottom right corner.

Event Time	Display Path	Name	Event State	Priority	Ack'd By	Event Value
3/9/16 10:37 AM	Ramp	Low Alarm 2	Ack	High		
3/9/16 10:36 AM	Ramp	Low Alarm 2	Clear	High		27.2267
3/9/16 10:36 AM	Ramp	Low Alarm 2	Active	High		-23.44

3 events

Focus

When you have a big list of all alarm events, the Active, the Cleared and the Acknowledged states are not going to be in sequential order depending on when the event times actually happened so it helps to use the Focus feature to filter on alarms. Using the Focus feature, you can actually focus on one

Target Alarm Source or the Target Event Id to look at when alarms occurred. It's a nice way of filtering through the Alarm Journal Table without having to scroll through the entire list of alarms to match up the Alarm Events, Times and States.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Journal - Searching



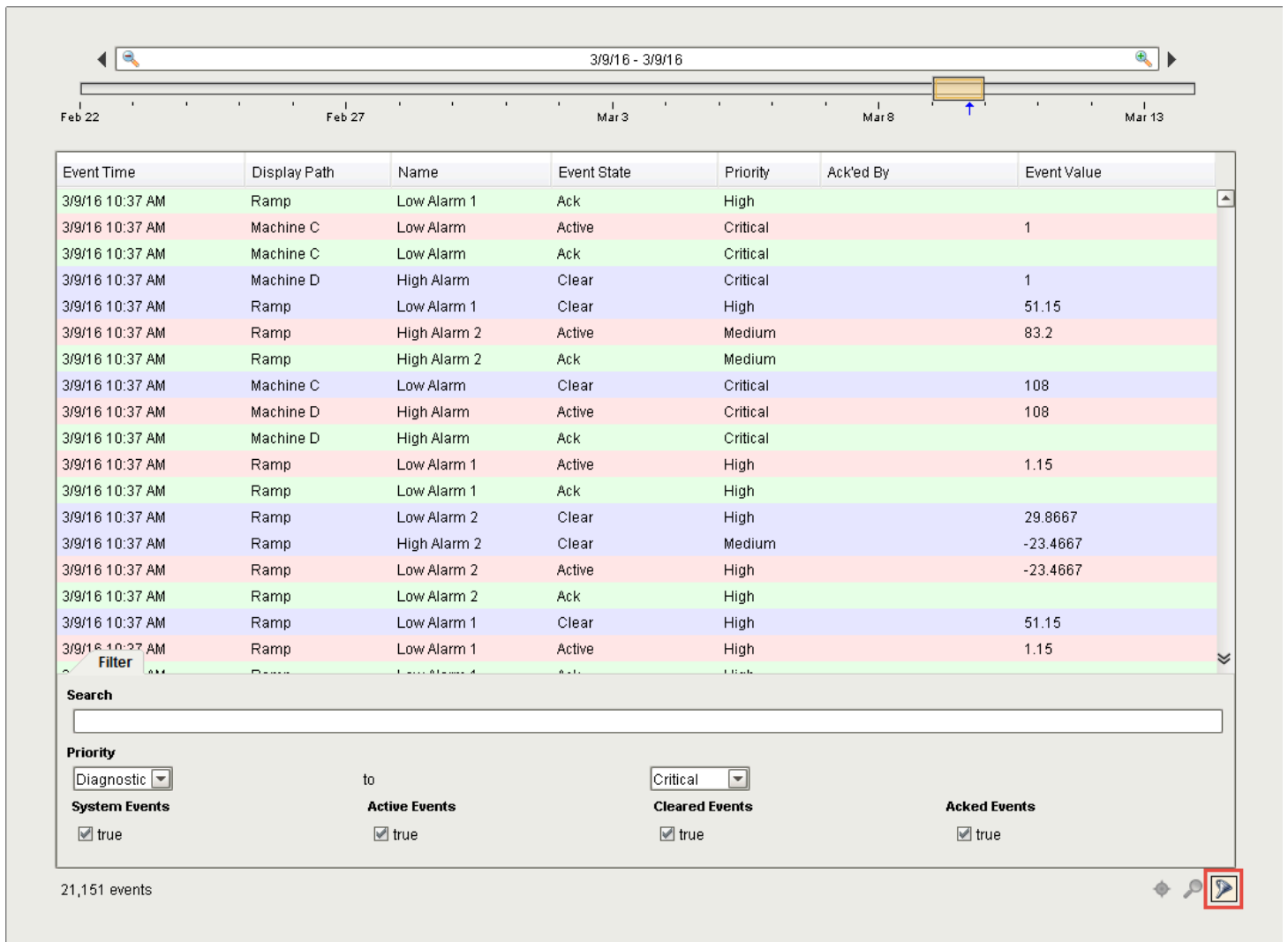
INDUCTIVE
UNIVERSITY

Alarm Journal - Searching

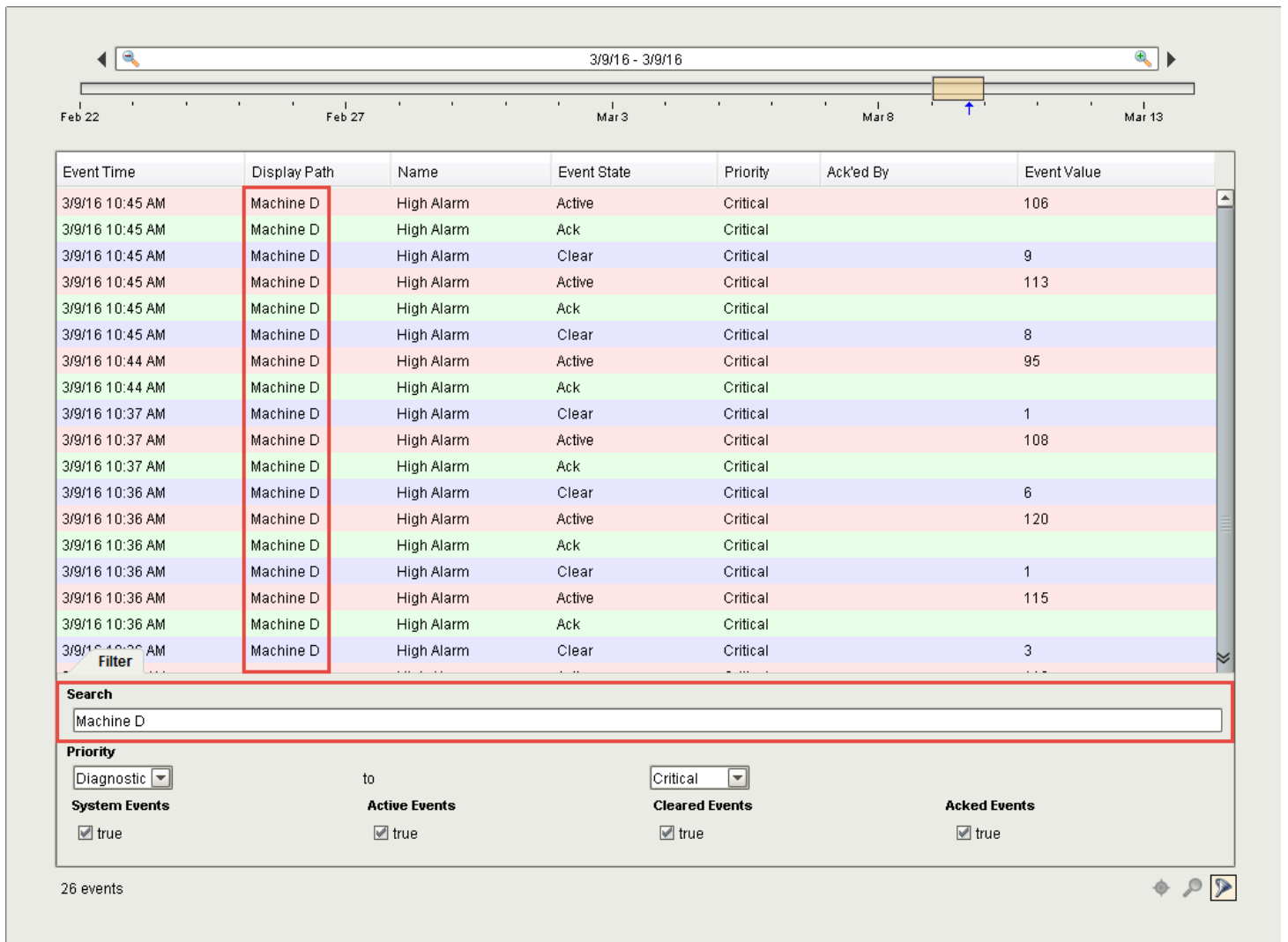
[Watch the video](#)

Using the Search Feature

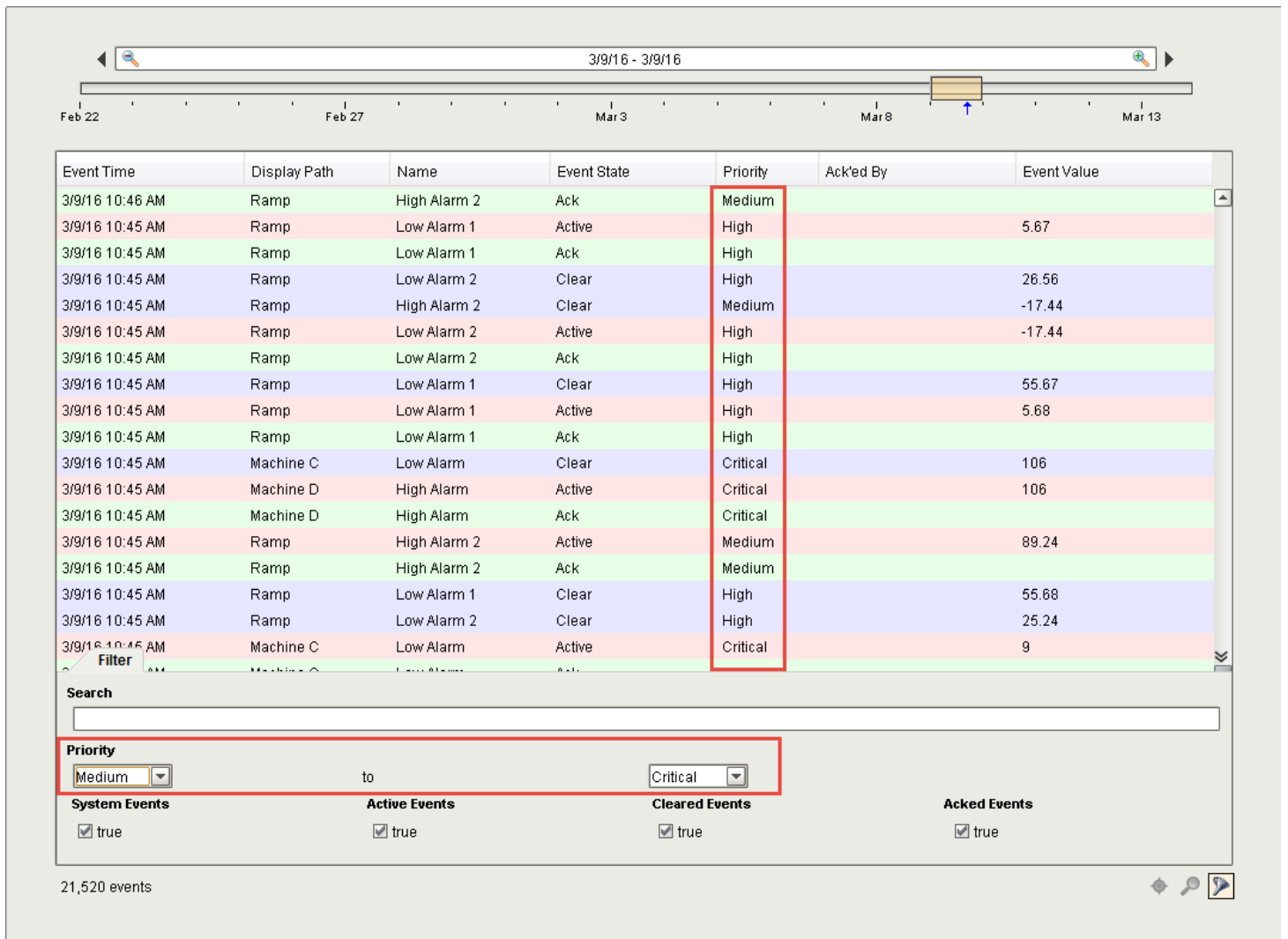
The Alarm Journal Table component can search through alarm history and filter for particular alarms with any criteria. The Search feature is built into the Alarm Journal Table component. On the bottom right side of the table, you can see the **Filter**  icon. Click on the Filter icon to start your search.



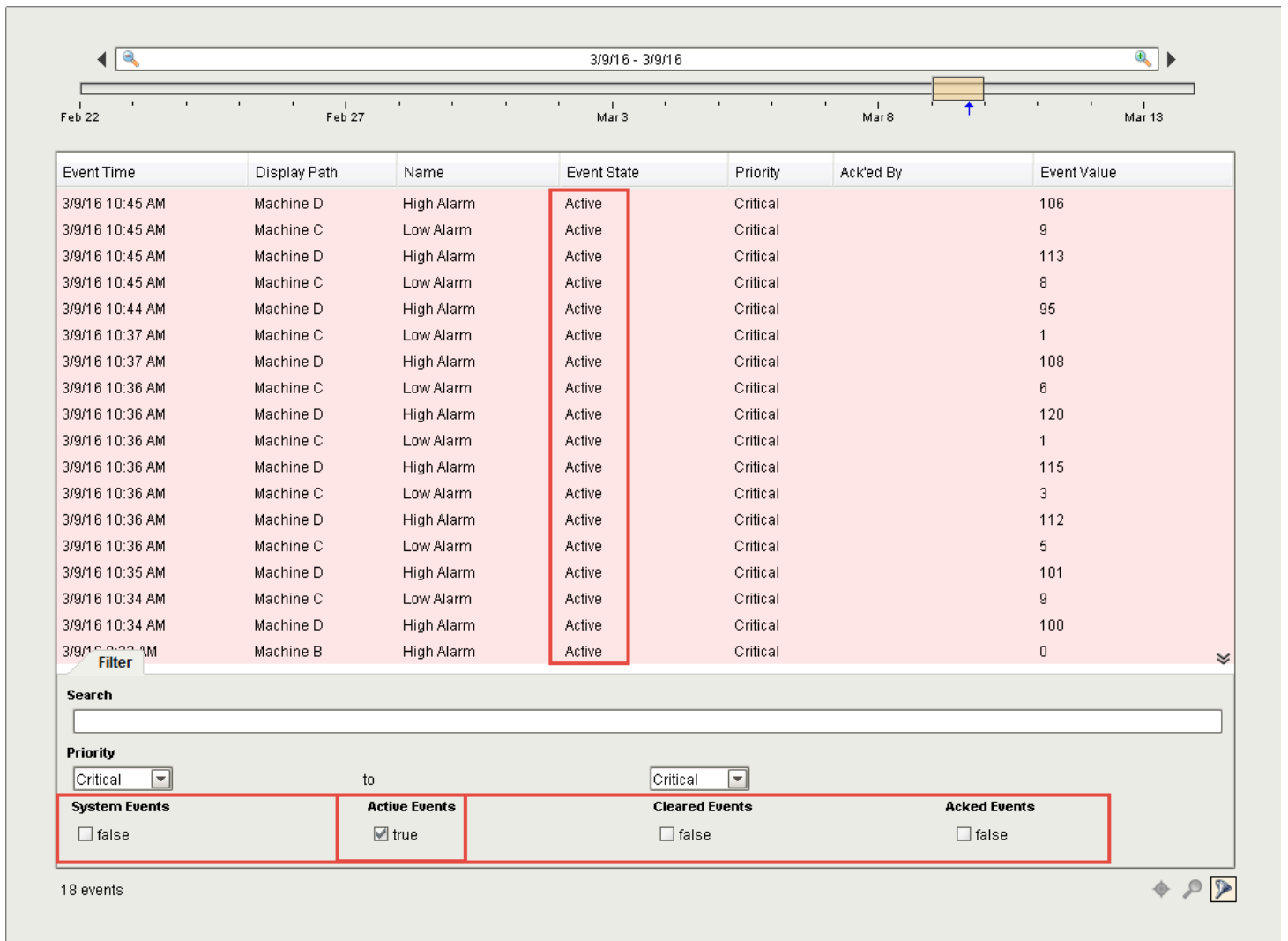
You can type in any search criteria you want as well as filter on **Priority** and **Event** types. This example shows how to find all Machine D alarms. Enter **Machine D** in the Search bar at the bottom of the table. The Alarm Journal Table will display only the results for Machine D.



This example filters on **Priority** and shows all the alarms with a priority from Medium to Critical.



You can also filter based on the type of **Events** you want to see. If you only want to look for just the Active alarms, you can uncheck all other events and simply leave the **Active Events** checked.



As you can see, the operator has the ability to search and filter on anything they like built right into the Alarm Journal Table component.