

1. Alarm Notification	2
1.1 Notification Contact Info	5
1.2 On-Call Rosters	8
1.3 Notification Profile Types	12
1.3.1 Email Notification Profile	13
1.3.1.1 Email Alarm Acknowledgement	18
1.3.2 Simple One-Way Email Notification Profile	20
1.3.3 SMS Notification Profile	23
1.3.4 Twilio SMS Notification	26
1.3.5 Voice Notification Profile	29
1.3.5.1 Notification Security PIN	34
1.3.5.2 Voice Notification Scripts	37
1.3.5.3 Registering Voice Alarm Module to CUCM	42
1.3.6 Remote Gateway Notification	45
1.4 Alarm Notification Pipelines	50
1.4.1 Alarm Pipeline Designer Interface	54
1.4.2 Simple Pipeline	57
1.4.3 Pipeline Blocks	61
1.4.3.1 Notification Block	69
1.4.4 Pipeline - Filter on Alarm Priority	82
1.4.5 Pipeline - Filter on Alarm Associated Data	85
1.4.6 Pipeline - Simple Loop	89
1.4.7 Pipeline - Escalation	91
1.4.8 Alarm Pipelines Status	95
1.5 Adding Pipelines to Tags	96

Alarm Notification

The Alarm Notification module enables you to send a message to a group of users when an alarm becomes active or cleared. This functionality is enabled by having the Alarm Notification module installed with alarm pipelines and Email notification set up. Additional modules can be installed which add [SMS](#) notification and [Voice](#) notification capabilities

On this page ...

- [Five Steps to Alarm Notification](#)
 - [Notification User](#)
 - [On-Call Roster](#)
 - [Alarm Notification Profile](#)
 - [Alarm Notification Pipeline](#)
 - [Adding Pipelines to Tags](#)
- [Notification Testing](#)

Five Steps to Alarm Notification

Regardless of what type of notification you decide to use, there are five simple steps to getting alarm notification set up on an Ignition Gateway.

Notification User

Setting up [users with contact information](#) is the first step to sending out alarm notifications. Most users will have already added contact information when they set up their username and password. This step is typically done in the Config section of the Gateway Webpage.

On-Call Roster

The second step to sending out alarm notifications is creating an [On-Call Roster](#). An On-Call Roster is a collection of users that are notified when an alarm occurs. 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. The On-Call Rosters are set up in the Config Section of the Gateway Webpage.

Alarm Notification Profile

The [Notification Profile](#) is the third step, and determines what type of notification (Email, SMS, or Voice) is sent out. The notification profile typically connects to an outside system that can do the actual notification. Each alarm notification type can be configured for both one-way and two-way notifications. A notification profile will be set up in the Config section of the Gateway Webpage.

Alarm Notification Pipeline

The next step is setting up an [Alarm Notification Pipeline](#). The Alarm Notification Pipeline creates a list of actions that the alarm will do. A pipeline can send out notification using pre configured Notification Profiles and On-Call Rosters, and can have logic to determine who to notify and when. Alarm Notification Pipelines have a section in the Ignition Designer where they are created.

Adding Pipelines to Tags

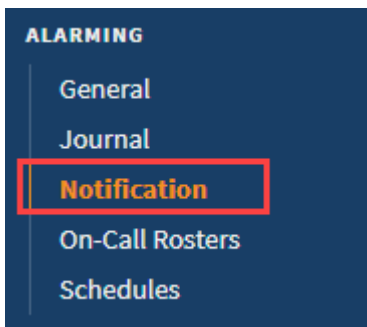
The last step of the alarm notification process is to [add an Alarm Notification Pipeline to a Tag](#). Every alarm on a Tag can have a pipeline set up that the alarm will use when the alarm becomes active, which is most common, but can also use a pipeline to notify users when an alarm becomes acknowledged or gets cleared. This last step of the process is configured on individual alarms on Tags.

Notification Testing

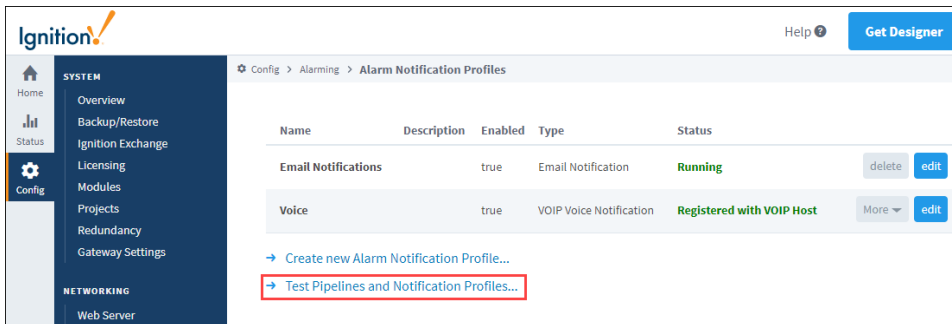
Before assigning Pipelines to alarms, the notification system can inject test alarms to help troubleshoot any problems. A notification will be sent out to any On-Call Rosters in the selected pipeline, so it is recommended to either enable test mode on any Notification blocks, or create a testing-only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

To test alarms, follow these simple steps:

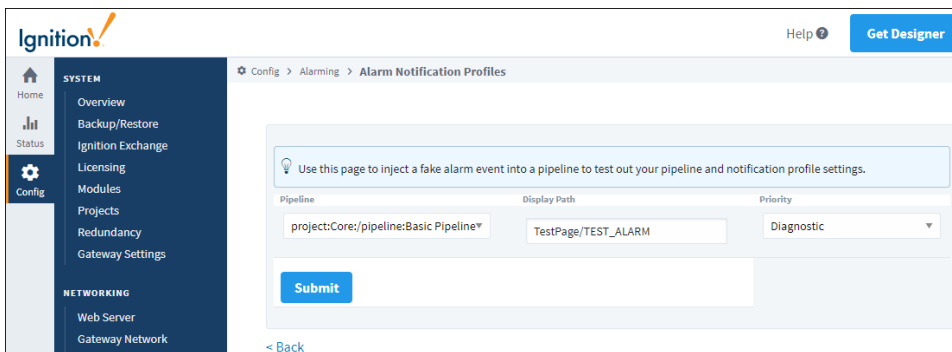
1. Go to the Gateway webpage and select the **Config** section.
2. From the menu on the left, select **Alarming > Notification**. The Alarm Notification Profiles page will open.



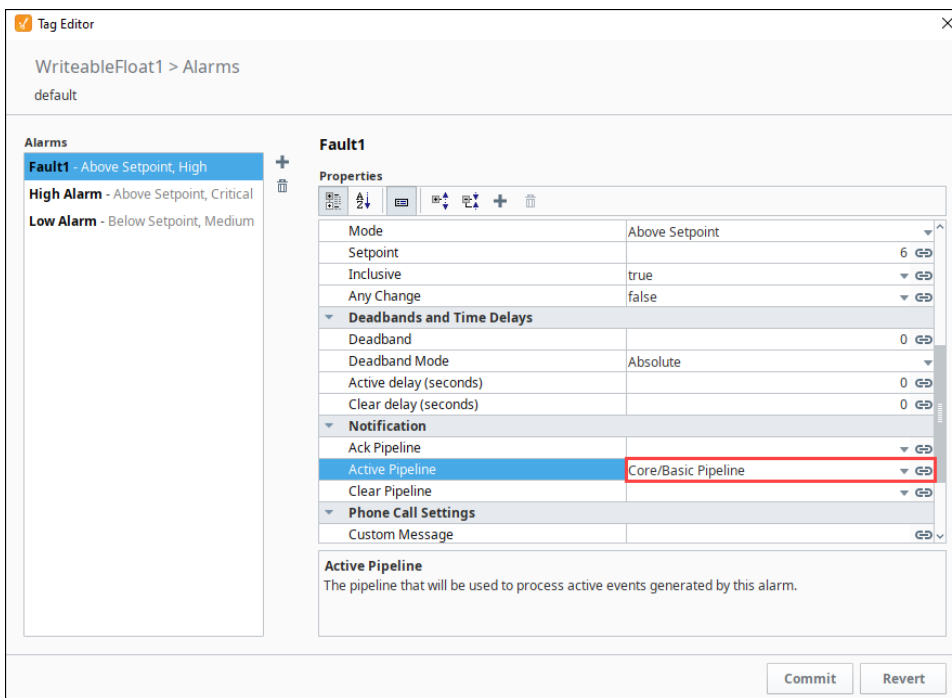
3. Click on **Test Pipelines and Notification Profiles**.



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



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.
6. The **Tag Editor** will open. Click on **Alarming**.
7. Select an alarm and go to the Notification section, and choose a pipeline (i.e., Escalate) to assign to your alarm. Click **OK**.



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

In This Section ...

Notification Contact Info

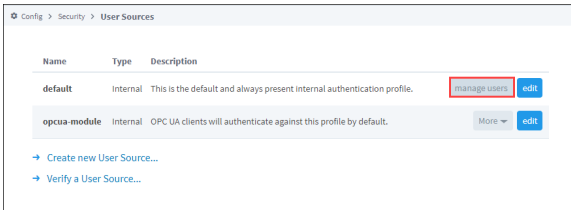
Ignition provides the capability to send out alarm notifications to a list of people in Email, SMS, and Phone (voice dial-out). There are several steps to get alarm notifications sent out to a list of people. The first thing to do is create an [Alarm Notification Profile](#) for [Email](#), [SMS](#) 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.

Add a User

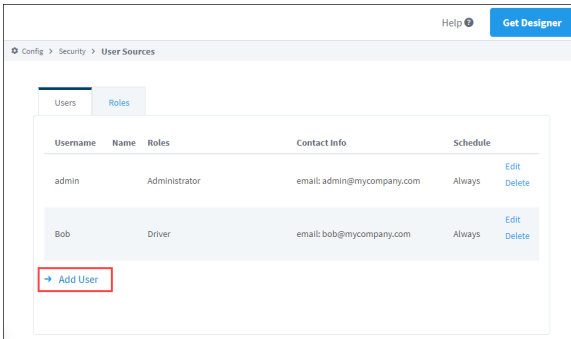
1. Go to the **Config** section of the Gateway Webpage.
2. Scroll down to **Security > Users, Role**.



3. Select **manage users** from the appropriate user source.




4. Look for the blue arrow and select the **Add User** link to add a user.



5. Complete the fields that correspond to the user's information. for example:

On this page ...

- [Add a User](#)
- [User Sources Settings](#)



INDUCTIVE UNIVERSITY

Notification Users

[Watch the Video](#)

6. In the **Contact Info** section, click **Add Contact info**.
7. Choose the delivery method from the dropdown list, and enter the appropriate Value such as an email address.

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

User Sources Settings

The following table describes all the settings on the User Source screen.

User Properties	
Setting	Description
Username	User name
Password	Password for this user.
Password	Re-type the password for verification
First Name	First name of the user (Optional).
Last Name	Last name of the user (Optional).
Roles	Select the roles this user will be assigned.
Schedule	Select the schedule for the user. Default is Always.
Language	Select the language for the user. Default is English.
Notes	Any notes about the user. (Optional).
Badge	Badge ID information (Optional).
Contact Info	
Type	Dropdown list of contact information types, such as Email.
Value	Contact information (i.e., email address).
Extended Properties	
Security PIN	Security PIN for the user (Optional).

Related Topics ...

- [Alarming Schedules](#)

On-Call Rosters

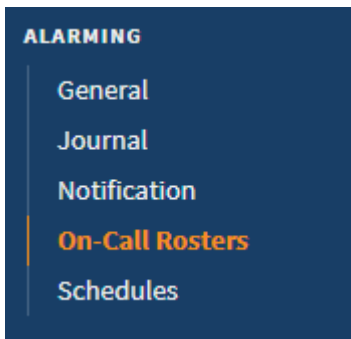
The On-Call Roster lets you create user groups to be notified when an alarm occurs. Each group includes a list of users in a specific order. Alarm pipeline's notification blocks must choose an on-call roster which identifies the users to notify for that notification block. When notifications are sent out by a pipeline, all users on the roster will be notified, save for those that are off shift (see [schedules](#)).

On-Call Rosters are statically defined, in that they can't be modified from a pipeline. However a [calculated roster](#) can be created on demand on a Notification block.

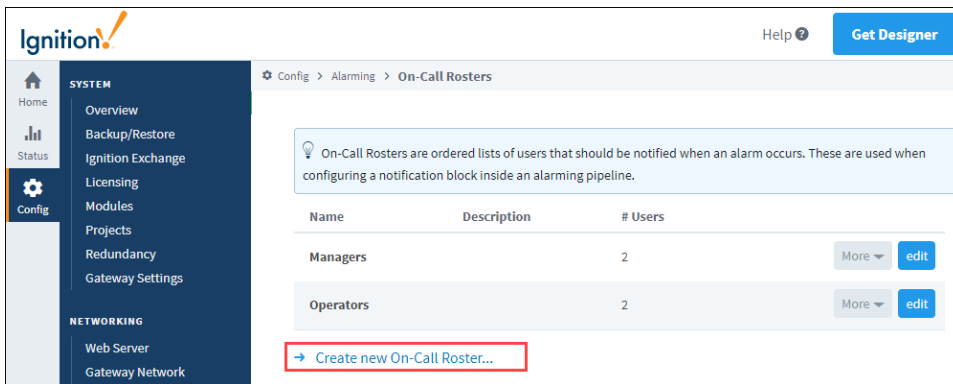


Create and Manage an On-Call Roster

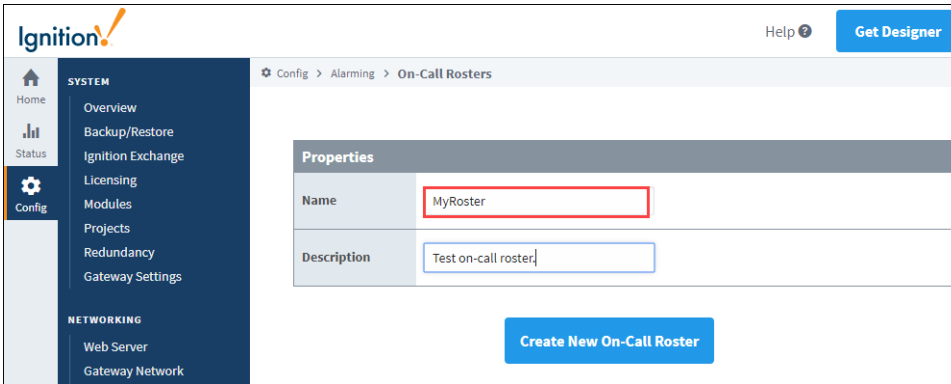
1. Go to the **Config** section of the Gateway Webpage.
2. Scroll down to **Alarming > On-Call Rosters**.



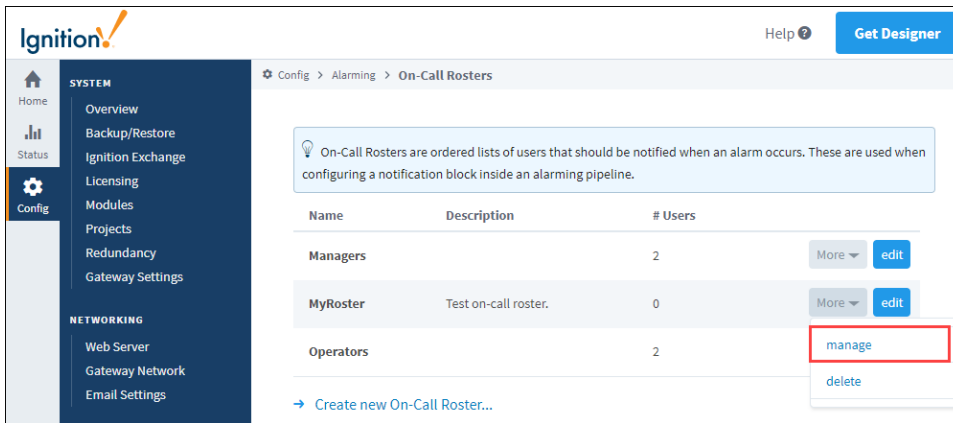
3. Click on **Create new On-Call Roster**.



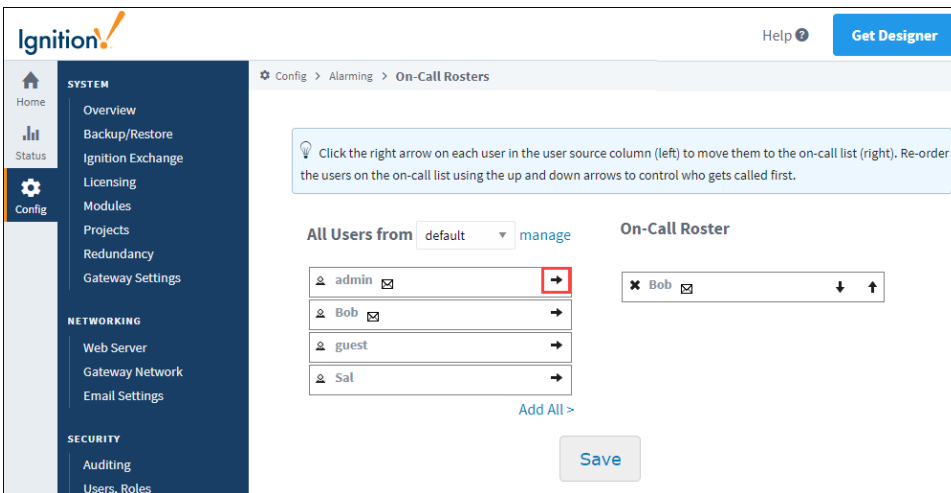
4. On the **New On-Call Roster**, enter a **Name**, and click **Create New On-Call Roster**.



- 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. 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.

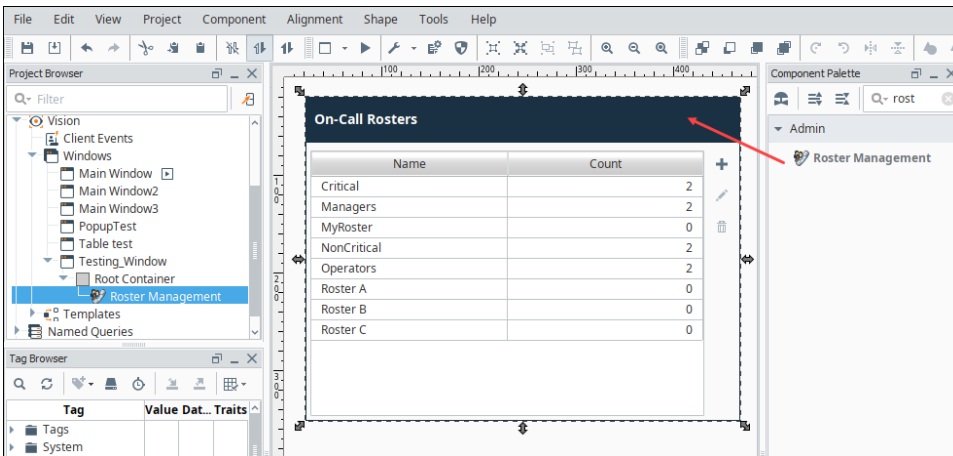



- Click on **Save**.

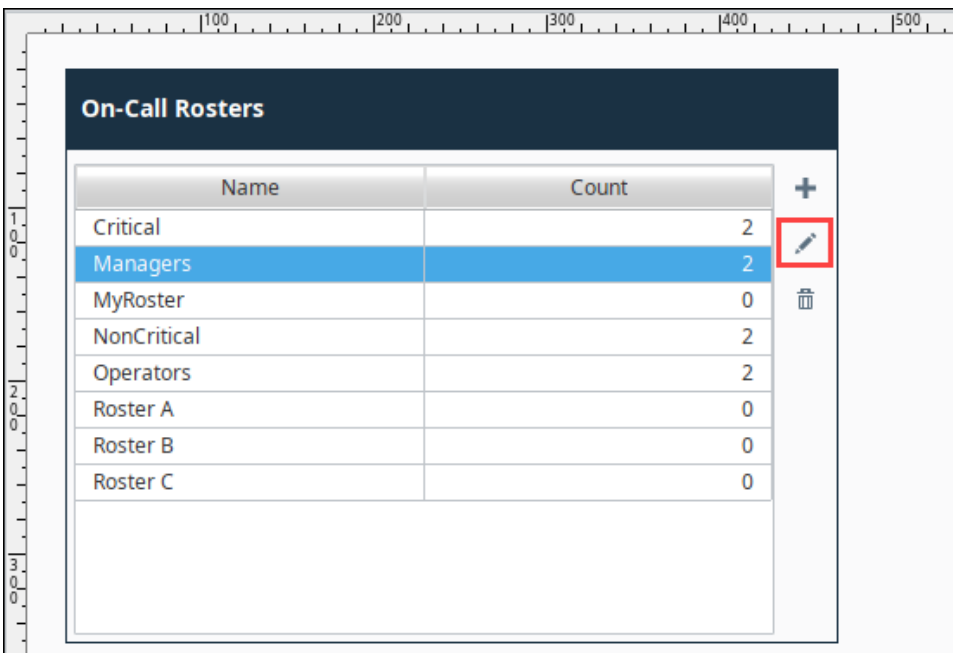
Manage the Roster from the Client



Roster management also can take place from within a Vision Client.

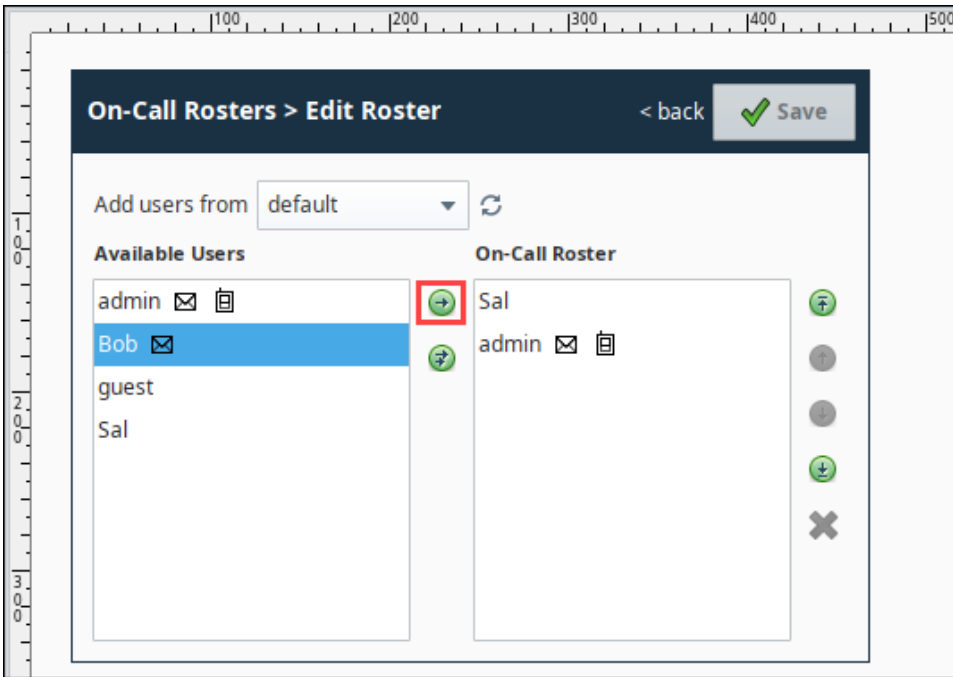
- Open your project in the **Designer** and navigate to a window the client has access to. If this is a new window, you will have to add it to your navigation.
- From the Component Palette, drag and drop a **Roster Management** component in your design window.




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** your project.
5. To interact with the roster, either launch your Vision Client or put your Designer into Preview mode.
6. To edit the roster, double-click any where on the roster or click the **Edit**  icon in the upper right.



7. On the Edit Roster screen, you can add an available user to the On-Call Roster, by selecting the user and clicking the **Add User**  icon. Alternatively you can add all users by clicking the **Add All Users**  icon.



8. To delete a user from the roster, select the user and click the **Delete**  icon.

For additional information, see the [Vision - Roster Management](#) component.

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, by text message, or by pressing 1 on the phone. Details on how to create and configure each of these profiles are found on the following sub-pages:

- **Email Notification Profile** - 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.

On this page ...

- [Module Requirements](#)
 - [Additional Requirements](#)

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 a 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.

[In This Section ...](#)

Email Notification Profile

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.

For notifications to be sent, Ignition must be able to access an SMTP server that accepts user credential (username and password).

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

On this page ...

- [Create an Email Notification Profile](#)
- [Email Notification Profile Settings](#)
 - [Email Settings](#)
 - [Two-Way Settings](#)
 - [POP3 Two-Way Settings](#)
 - [Auditing](#)
 - [Advanced](#)

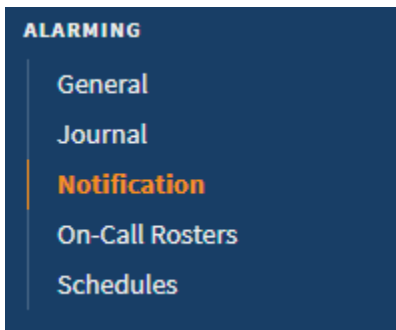


Email Notification Profile

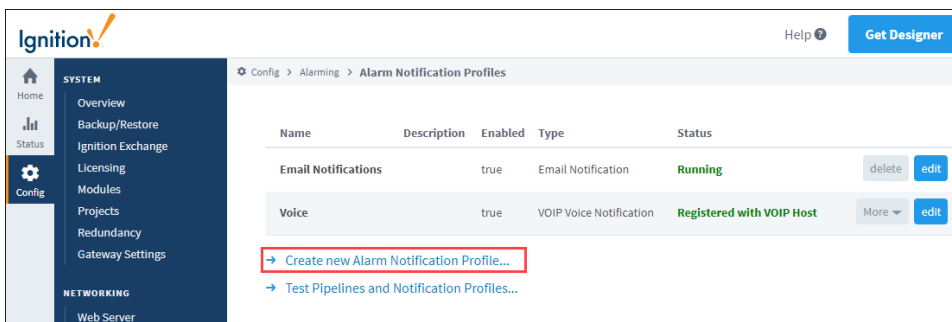
[Watch the Video](#)

Create an Email Notification Profile

1. Go to the **Config** section of the Gateway Webpage.
2. Scroll down and select **Alarming > Notification** from the menu on the left side.



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



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**.

Config > Alarming > Alarm Notification Profiles

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 settings for the Email 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 a **Username** and **Password** for the SMTP server, if they are required.

Config > Alarming > Alarm Notification Profiles

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.

6. Click **Create New Alarm Notification Profile**.

- 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.

Config > Alarming > Alarm Notification Profiles

✓ Successfully created new 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
SMS		true	SMS Notification	Running	delete edit

Note:

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 a 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. Note that this address must be reachable from wherever the notification emails will be accessed from. In some cases this means users acknowledging the alarms via the email must be connected to a VPN. If you're simply testing two-way notifications from the gateway server, you can use <code>localhost:8088</code> for this field.
Send HTTPS Link	Specifies if the link should use a HTTPS link for SSL.

POP3 Two-Way Settings

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 the 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 (<code>wrapper.log</code>).
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.

Related Topics ...

- [SMS Notification Profile](#)
- [Alarm Notification Pipelines](#)

In This Section ...

Email Alarm Acknowledgement

Email Alarm Acknowledgement allows operators to acknowledge alarms via a link that is included directly in the email alarm notification. The email alarm acknowledgement process a simple two-step process:

- The operator receives an email in their inbox alerting them to an alarm(s).
- The link in the email directs the operator to a landing page describing the alarm(s) where the operator can add notes and acknowledge the alarm(s).

Once you have an [Email Notification Profile](#) created in Ignition, operators can be notified when alarms occur and acknowledge alarms via email. In order to acknowledge alarms in an email, the **Two-way Enable** option must be set to 'true' in the [Email Notification Profile](#).

Note: This page assumes an [Email Notification Profile](#), an [On-Call Roster](#) containing users with valid email addresses who will be receiving and acknowledging alarms, and an [Alarm Notification Pipeline](#) are configured, including assigning the [alarm to a pipeline](#).

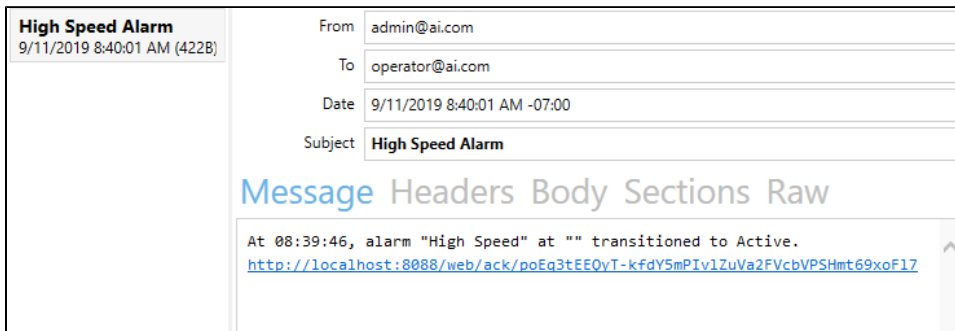
On this page ...

- [Using Email Alarm Acknowledgement](#)
 - [Using Email Acknowledgement with POP3](#)

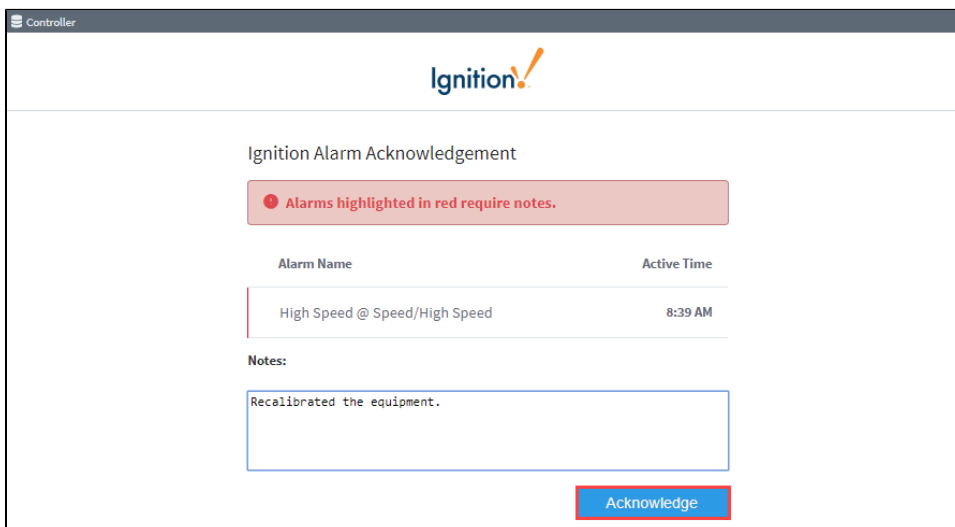
Using Email Alarm Acknowledgement

Acknowledging an alarm in an email is super simple!

1. Open the the alarm notification email message, and click the link.



2. A new window opens describing the alarm. This example shows one email for one alarm.
3. If notes are required as in the following example, enter notes about the alarm.
4. Click the **Acknowledge** button.



5. If [notification consolidation](#) is configured, you can limit the number of email notifications that will be sent, thus, consolidating multiple alarms into one email for operators to acknowledge.

Controller

Ignition!

Ignition Alarm Acknowledgement

Action	Alarm Name	Active Time
<input checked="" type="checkbox"/>	High Speed @ Speed/High Speed	11:09 AM
<input checked="" type="checkbox"/>	Low Amps @ Motor Plant/Motor 2/Amps/Low Amps	11:12 AM

[Select All](#)

Notes:

[Acknowledge](#)

Using Email Acknowledgement with POP3

When using POP3, you can reply back to the alarm event in order to acknowledge the event. To do this, reply back using the syntax `Pop3AckId: <systemId>: <ackId>` in the body, replacing `systemID` and `ackID` with the appropriate information. This should be done even in an empty reply.

Note: If you are not receiving any email alarm notifications, check **Status > Logs** for any error messages. If you are getting errors saying emails are "unreachable," then check your [Email Notification Profile](#).

If you run into any issues acknowledging email alarms, contact the [Support department](#).

Related Topics ...

- [Alarm Notification](#)
- [Alarm Notification Pipelines](#)

Simple One-Way Email Notification Profile

The Simple One-Way Email Notification profile is another type of Email Notification profile that can be set up 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.

It is important to be aware that there is no way to filter which 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.

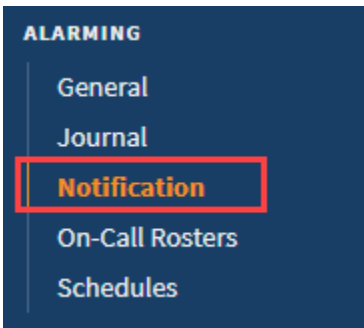
Note: The Simple One-way Email Notification requires the [Alarm Notification Module](#) to be installed.

On this page ...

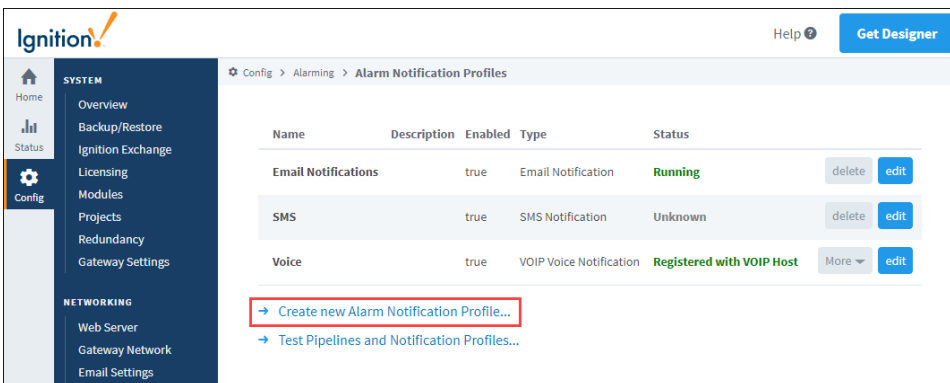
- [Create a Simple One-Way Email Notification Profile](#)
- [Simple One-Way Email Notification Profile Settings](#)

Create a Simple One-Way Email Notification Profile

1. Go to the **Config** 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.



4. A list of all the alarm notification profile types will appear.

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, go to the Config tab of the Gateway Webpage and click on **System>Modules**.)

5. Choose **Simple One-Way Email Notification**, and click **Next**.
6. Configure the profile settings for the Simple One-Way Email Notification Profile, and then click **Create New Alarm Notification Profile**.
 - **Name:** One-Way Email
 - **SMTP:** SMTP (Please see the [Email Settings](#) page for more information on creating an SMTP profile.)
 - **On-call Roster:** Managers (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**.

Config > Alarming > Alarm Notification Profiles

Main

Name	<input type="text" value="OneWay Email"/>
Description	<input type="text"/>
Enabled	<input checked="" type="checkbox"/> (default: true)

Email Settings

SMTP Profile	<input type="text" value="SMTP"/>
On-Call Roster	<input type="text" value="Managers"/>

Alarm Criteria

Minimum Priority	<input type="text" value="High"/> (default: Low)
Maximum Priority	<input type="text" value="Critical"/> (default: Critical)

[Create New Alarm Notification Profile](#)

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

Config > Alarming > 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

Simple One-Way Email Notification Profile Settings

Property Name	Description
Main	
Name	Name of the profile.
Description	Brief description of the profile. Optional.

Enabled	Whether the Simple One-Way Email Notification is enabled.
Email Settings	
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	
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. (Options are Diagnostic, Low, Medium, High, or 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. (Options are Diagnostic, Low, Medium, High, or Critical.)
Advanced	
From Address	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.29 Click here to check out the other new features</p> </div> <p>An optional address to use for the From field. If empty, the email session username will be used.</p>

Related Topics ...

- [Notification Profile Types](#)
- [On-Call Rosters](#)
- [Alarming](#)

SMS Notification Profile

SMS Notification Module

Short Message Service (SMS) is the protocol that is used by text messaging systems. 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 Notification 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/RV50X, RV55 and LX40. Most devices from most manufacturers that support the Airlink IP protocol 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. To learn more, refer to the Knowledge Base article by clicking the link on the right side of this page.

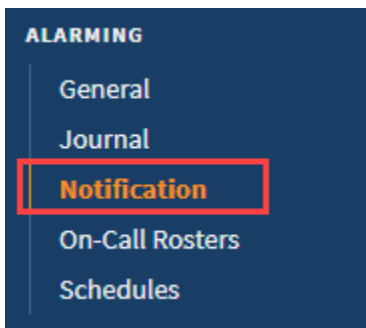
For more information about various AirLink® devices, see: <https://www.sierrawireless.com/router-solutions/airlink/>

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 **Config** section of the [Gateway](#) Webpage
2. Scroll down to **Alarming > Notification** from the menu on the left.



3. On the Alarm Notification Profiles page, click on **Create new Alarm Notification Profile**.
4. Select the **SMS Notification** profile type and click **Next**.

On this page ...

- [SMS Notification Module](#)
 - [Device Configuration](#)
 - [Multiple Systems with One Modem](#)
- [Create an SMS Notification Profile](#)
- [SMS Notification Profile Settings](#)

AirLink LS300 Setup Guide

[Link to Knowledge Base Article](#)

AirLink RV50 Setup Guide

[Link to Knowledge Base Article](#)



SMS Notification Profile

[Watch the Video](#)

Config > Alarming > Alarm Notification Profiles

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.

Twilio SMS Notification
 Send SMS alarm notifications via a Twilio account.

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.

Config > Alarming > Alarm Notification Profiles

Main

Name:

Description:

Enabled: (default: true)

SMS Settings

Airlink Host Address:

Send Port: (default: 17341)

Receive Port: (default: 17342)

Two-way Enabled: (default: false)

Numeric Only Ack Code: (default: false)

Auditing

Audit Profile:

[Create New Alarm Notification Profile](#)

Once you have the SMS Notification profile set up, you can add phone numbers to your contacts using the **SMS Contact** type. SMS notifications can also be sent with the [Twilio Module](#).

SMS Notification Profile Settings

The following table describes the settings on the Alarm Notification Profiles page for a Twilio SMS Notification profile.

Setting	Description
Main	
Name	Name for this alarm notification profile.
Description	Description of the profile.
Enabled	Whether the profile is enabled or disabled.
SMS Settings	
Airlink Host address	Enter the Airlink host address for this profile.
Send Port	The send port. (Default is 17,341.)
Receive Port	The receive port. (Default is 17,342.)
Two-Way Enabled	Whether or not two-way SMS is enabled.(Default is false.)
Numeric Only Ack Code	Whether to accept only numeric ack codes. (Default is false.)
Auditing	
Auditing Profile	Select an audit profile from the dropdown list.

Related Topics ...

- [Twilio Module](#)
- [Voice Notification Profile](#)
- [Alarm Notification Pipelines](#)

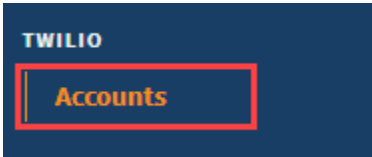
Twilio SMS Notification

SMS notifications can also be sent with the [Twilio Alarm Notification](#) 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.

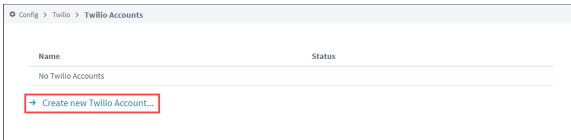
Note: The Twilio module is dependent on the [Alarm Notification module](#). You must have both modules installed to send alarms.

Define a Twilio Account

1. Navigate to the **Config** section of the Gateway Webpage.
2. Scroll down and choose **Twilio > Accounts**.



3. On the Twilio Accounts page, 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. The following feature is new in Ignition version **8.1.31**
[Click here](#) to check out the other new features

If desired, you can add an IP address of a local network to the **Local Address**, which will then be used instead of the default network interface.

6. 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.
7. Click the **Create New Twilio Account** button.

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

Twilio Accounts Settings

The following table describes the settings on the Twilio Accounts page.

Setting	Description
Twilio	
Name	Name for the Twilio Account.
Twilio Account Sid	Twilio account Sid: retrieve from the Account Settings menu in Twilio. (Required.)
Twilio Auth Token	Twilio auth token: retrieve from the Account Settings menu in Twilio. (Required.)
Local Address	<p>The following feature is new in Ignition version 8.1.31 Click here to check out the other new features</p>

On this page ...

- [Define a Twilio Account](#)
- [Twilio Accounts Settings](#)
- [Create a Twilio SMS Notification Profile](#)
 - [Twilio SMS Notification Profile Settings](#)
- [Receiving Alarm Acknowledgement via Twilio](#)

AirLink LS300 Setup Guide

[Link to Knowledge Base Article](#)

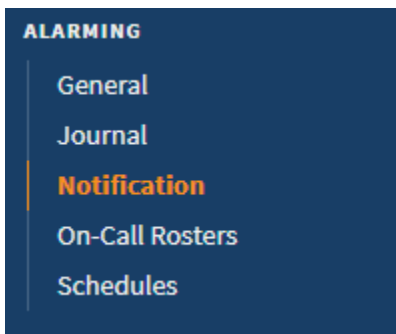
AirLink RV50 Setup Guide

[Link to Knowledge Base Article](#)

	The IP address of the local network interface to be used for Twilio communications. Leave blank to use the default interface.
Inbound	
Public Hostname	The publicly reachable hostname or IP address that is configured to forward inbound requests to this Ignition gateway. This must be configured in order to receive inbound SMS. Leave blank if inbound SMS are not needed.
Public Port	The port on the Public Hostname to make the connection to. Default is 80.
Backup Public Hostname	The publicly reachable hostname or IP address that is used when the backup of a redundant pair becomes active.
Backup Public Port	The port on the Public Hostname to make the connection to that is used when the backup of a redundant pair becomes active. Default is 80.
HTTPS Enabled	True if the callback URL should use HTTPS instead of HTTP. Default is false.
Inbound Validation Enabled	Cryptographically validate that inbound requests over HTTPS are from Twilio. Default is false.

Create a Twilio SMS Notification Profile

1. Go to the **Config** section of the Gateway Webpage.
2. Scroll down and select **Alarming > Notification** from the menu on the left side.



3. Click on the **Create new Alarm Notification Profile** link.
4. Select the **Twilio SMS Notification** profile type, and click **Next**.

The image shows a web interface for configuring alarm notification profiles. The breadcrumb trail is 'Config > Alarming > Alarm Notification Profiles'. There are seven radio button options, each with a description:

- 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.
- Twilio SMS Notification**: Send SMS alarm notifications via a Twilio account. (This option is selected and highlighted with a red border.)
- VOIP Voice Notification**: Telephone notification using VOIP, compatible with most SIP based telephony systems.

 A blue 'Next >' button is located at the bottom right of the form.

5. Provide a name for the profile, and select a **Twilio Account** from the dropdown.
6. Set the other profile settings as desired. (See table below for a description of all settings.)
7. Click **Create New Alarm Notification Profile**.

Twilio SMS Notification Profile Settings

The following table describes the settings on the Alarm Notification Profiles page for a Twilio SMS Notification profile.

Setting	Description
Name	Name for this alarm notification profile.
Description	Description of the profile.
Enabled	Whether the profile is enabled or disabled.
Twilio Account	Select a Twilio account from the dropdown list.
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.
Auditing Profile	Select an audit profile from the dropdown list.

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

Receiving Alarm Acknowledgement via Twilio

If you're using 2-way alarming, when a user acknowledges an alarm, Twilio's servers send the acknowledgement to your Ignition Gateway. This means your Ignition Gateway needs to be accessible from Twilio's servers. If you're running Ignition on the cloud, this is likely already the case. If Ignition is on-premise, you may need to explore one of the following options:

1. Make use of port forwarding
2. Put an Ignition alarming server in the cloud with a Gateway Network connection to your on-premise server.

Because there are often security implications of port forwarding, and a cost associated with adding an Ignition server in the cloud, often the SMS Notification module is chosen over the Twilio module for users running Ignition on-premise.

Related Topics ...

- [Voice Notification Profile](#)
- [Alarm Notification Pipelines](#)

Voice Notification Profile

Note: For the **Voice Notification Module** to work, a compatible [Voice Module](#) and Text-to-Speech (TTS) voices must be installed. See the [Voice Notification Downloads](#) Knowledge Base article for more information on TTS voices available to download.

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.

Hosted SIP Services

The voice notification module can be paired up with SIP-based services, although there may need to be some manual configuration. See our guide on integrating [Ignition with AWS Chime](#).

Gateway Configuration

To get started with voice notification, add a new profile by going to **Config-> 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.

Configuring Messages

On this page ...

- [Core Features](#)
- [Voice Notification Configuration](#)
 - [About SIP and VOIP](#)
 - [Hosted SIP Services](#)
 - [Gateway Configuration](#)
 - [Configuring Messages](#)
 - [The Call Lifecycle](#)
 - [Pipeline Configuration](#)
- [Create a Voice Notification Profile](#)
 - [VOIP Voice Notification Profile Settings](#)



Voice Notification Profile

[Watch the Video](#)

How to setup Ignition with AWS Chime

[Link to Knowledge Base Article](#)

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

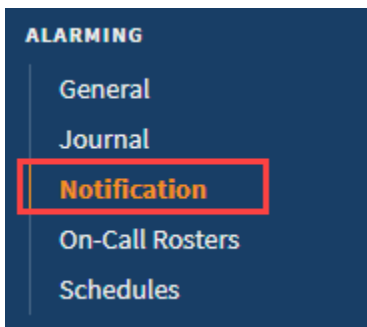
The voice notification system is accessed through the Notification Block in the Designer's pipeline configuration, like all other notification methods. As with other methods, you select a call roster, and can optionally turn on consolidation. There are only two options which can be set on the notification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

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.

Create a Voice Notification Profile

1. Go to the **Config** 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.

Config > Alarming > Alarm Notification Profiles

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

→ Create new Alarm Notification Profile...

→ Test Pipelines and Notification Profiles...

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

Config > Alarming > Alarm Notification Profiles

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.

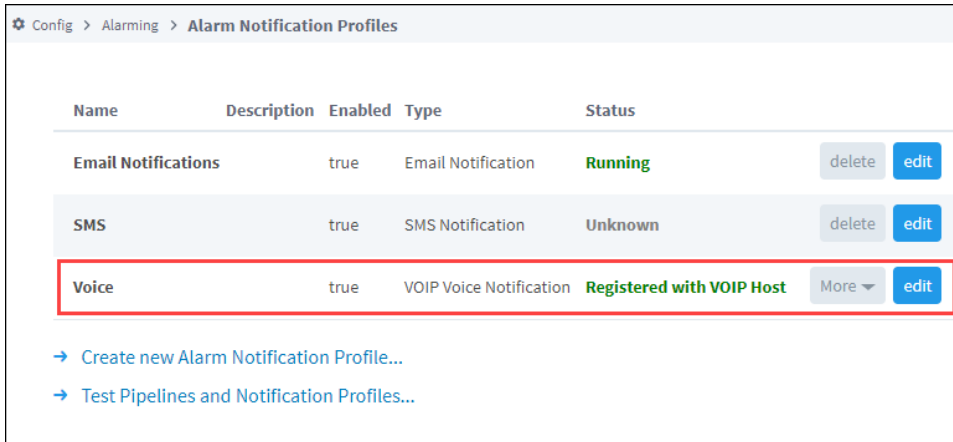
Twilio SMS Notification
Send SMS alarm notifications via a Twilio account.

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 set up depending on how you want to manage your calls, and based on your preference. Refer to the table below for a description of all settings.

6. 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**.



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

VOIP Voice Notification Profile Settings

The following table describes the settings on the Alarm Notification Profiles page for a VOIP Voice Notification profile.

Setting	Description
Main	
Name	Name for this <u>alarm</u> notification profile.
Description	Description of the profile.
Enabled	Whether the profile is enabled or disabled.
VOIP Gateway Settings	
Gateway address	The ip address or domain name of your SIP gateway.
Username/Account	User or account name for this profile.
Password	Enter a password for the account.
Password	Re-enter password for verification.
Outbound Proxy	The proxy to use, if any.
Call Settings	
Max Call Duration (minutes)	The length of time, in minutes, that any particular call is allowed to take. The call will be terminated if it lasts longer than this limit. (Default is 5.)
Answer Timeout (seconds)	How long, in seconds, to wait for an answer before moving on to the next contact. (Default is 60.)
Voice Rate	Adjusts the desired voice rate. (Default is Normal.)
Auditing	
Auditing	If an audit profile is selected, events such as phone calls and acknowledgements will be stored to the audit system. Note that alarm acknowledgements are also stored to the alarm journal
Advanced Settings	
Authorization Id	Some services require a separate authorization name, in addition to the user/account name. If left blank, the username specified above will be used.
SIP Port	SIP Port. (Default is 5,060.)
RTP Port	RTP Port. (Default is 8,000.)

Local Bind Address	Local bind address for the profile.	
Public Bind Address	Public bind address for the profile.	
Media Debug Enabled	If true, call audio will be recorded to the Ignition temp directory. The audio file will be generated after receiving a response from the SIP server. (Default is false.)	

Related Topics ...

- [Notification Contact Info](#)
- [Alarm Notification Pipelines](#)

In This Section ...

Notification Security PIN

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 correct person by requiring a 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 set up 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.

On this page ...

- [User Profile](#)
- [Voice Notification Profile](#)

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 **Config** section of the [Gateway](#) Webpage.
2. Scroll down to **Security > Users, Roles** from the menu on the left.
3. Under **default** user source, select the **manage users** link.

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 a User Source...](#)

4. 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.

Username	Name	Roles	Contact Info	Schedule	
admin		Administrator, Operator	email: admin@mycompany.com, phone: 555-555-5555	Always	Edit Delete
Bob		Driver	email: bob@mycompany.com	Always	Edit Delete
guest		Operator		Always	Edit Delete
Sal		Administrator, Driver, Operator		Always	Edit Delete

[→ Add User](#)

5. The user profile screen will open. Scroll down to the bottom of the screen, and verify that the contact information for Email and Phone is up-to-date.
6. Create a **Security Pin**, and press **Save Changes**.

Now, this PIN is associated with this user.

The screenshot shows a configuration page for 'User Sources' under 'Security'. It features a 'Contact Info' section with a table of user details and an 'Extended Properties' section with a 'Security PIN' field.

Type	Value	
E-Mail	admin@mycompany.com	Up / Down Delete
Phone	555-555-5555	Up / Down Delete

[Add Contact Info](#)

Extended Properties

Security PIN	<input type="text" value="8765432"/>
--------------	--------------------------------------

[< Cancel](#) [Save Changes](#)

Voice Notification Profile

The next step is to configure the Voice Notification Profile to require a PIN.

1. In the **Designer**, go to your **Alarm Notification Pipelines** and open your pipeline. This example uses the **Basic Email** pipeline.
2. Select the **Notification Block** in the workspace.
3. Click on the **Notification** tab, and choose **Voice**.
4. 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.

The screenshot displays a software interface for configuring alarm notification pipelines. The interface is divided into several sections:

- Project Browser:** Shows a tree view of 'Alarm Notification Pipelines' with sub-items like 'Basic Email', 'Basic Email 2', 'Email', 'Escalate', 'Looping Pipeline', 'New Pipeline', 'Notify', and 'Operators'.
- Pipeline Blocks:** A palette containing icons for 'Notification', 'Delay', 'Splitter', 'Switch', and 'Expression'.
- Pipeline Block Editor:** The main workspace for configuring a 'Notification Block'. It has tabs for 'Notification', 'Contacts', and 'Consolidation'. The 'Notification' tab is active, showing a configuration for '1. Voice'. Below this, a table lists various settings for the 'Voice' notification type.

Voice	
Require PIN	False
Allow Acknowledgement	True
Retries Per Contact	False
Delay Between Retries (S...	0
Test Mode	false

The Pipeline Block Editor also shows a 'Delay Between Messages (seconds):' field set to 0. The main workspace displays a flowchart with a 'START' block, a 'Notification' block (containing '1. Voice' and 'Roster: Managers'), and a 'Delay' block (set to '20 seconds').

Related Topics ...

- [Voice Notification Scripts](#)

Voice Notification Scripts

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](#) under Notification, and install them in your Gateway.

On this page ...

- [Voice Languages Overview](#)
- [Modifying a Voice Notification Script](#)

Notification	Version	Checksum
Alarm Notification Module (810KB)	5.0.10	sha-256
SMS Notification Module (519KB)	5.0.10	sha-256
Twilio Notification Module (8MB)	2.0.10	sha-256
Voice Notification Module (44MB)	5.0.10	sha-256

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

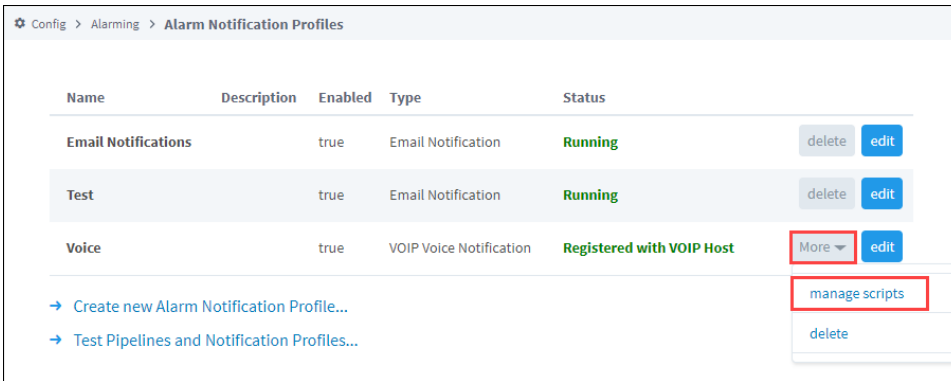
Once you [installed the voice modules](#), you can configure the voice scripts for any language in the voice notification module. For example, you may want to have a different script that gets sent out when you're calling someone in English verses someone that you're calling in Spanish.

Note: When we refer to a "script" on this page, we are talking about a speech or a dialog for Ignition to read to your users over the phone. These scripts have nothing to do with the scripting language (Python) that you write code with.

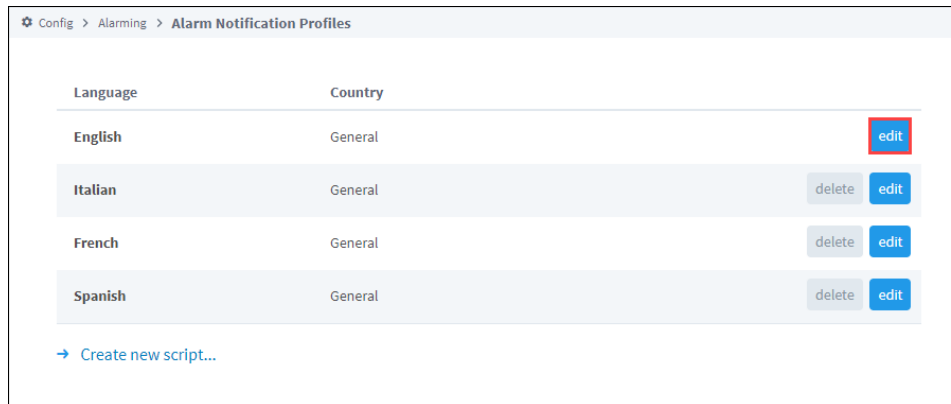
Modifying a Voice Notification Script

From 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 Config** page, and select **Alarming > Notification**. To the right of the Voice Notification Profile, click on the **manage scripts** link.



- 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.



- Default voice scripts are the same in all supported languages. You can edit default scripts to your requirements or stop any of the scripts from being read by removing all text in that field.

Note: The Call Script consists of a number of smaller scripts that the recipient of the call will hear over the phone informing them of one or more the alarm(s) depending on how the alarm notification is configured. When receiving a voice notification, the recipient will hear each script or instruction one at a time. For example, the first script is the Greeting script: "Hello this is an Ignition alarm notification". The Greeting script will then be followed by each of the remaining scripts to relay the alarm name, date and time of the alarm, alarm status, and how to acknowledge the alarm.

- When you're finished editing, click **Save Changes**.

Config > Alarming > Alarm Notification Profiles

Main	
Language	English (General) <small>(default: en)</small>

Call Script	
Greeting	Hello, this is an Ignition alarm notification.
PIN Challenge	Please enter your pin, followed by the pound sign. <small>A message asking the user to enter their PIN number. Must instruct them to press '#' when finished.</small>
Non-PIN Challenge	Please press any key to continue. <small>When a PIN is not required, this message is played to instruct the users to press any key to continue.</small>
Invalid PIN	Invalid pin number.
Summary	There are %d events. <small>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.</small>
Active Message	At {eventTime hh:mm:ss}, alarm named {name} became {eventState} with a value of {eventValue} <small>The message to play for active alarms. This message can reference properties of alarms by using syntax like "{propertyName}".</small>
Clear Message	At {eventTime hh:mm:ss}, alarm named {name} became {eventState} with a value of {eventValue} <small>The message to play for cleared alarms. This message can reference properties of alarms by using syntax like "{propertyName}".</small>
Option 1: Acknowledge	Press 1 to acknowledge <small>Tells the user to press '1' to acknowledge.</small>
Acknowledge Confirmation	Alarm has been acknowledged. <small>Tells the user that the alarm event has been successfully acknowledged.</small>
Option 2: Ignore	Press 2 to ignore <small>Tells the user to press '2' to ignore/bypass the alarm.</small>
Ignore Confirmation	Alarm has been ignored.
Option 3: Repeat	Press 3 to repeat <small>Tells the user to press '3' to repeat the last message.</small>
Closing	Thank you, goodbye.

[Save Changes](#)

5. Here is an example of the some of the default scripts in Spanish. Once again, you can edit the scripts to your requirements. When editing is complete, click **Save Changes**.

Greeting: Hola esta es una notificación de alarma.

PIN Challenge: Por favor, introduzca su número de PIN seguido de la tecla numeral.

Non-PIN Challenge: Presione cualquier número para continuar.

Invalid PIN: Ese número PIN es incorrecto.

Summary: Hay %d alarmas.

Config > Alarming > Alarm Notification Profiles

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.

Note: You can attach a different script for any language that Ignition supports depending on the user, and what that user is set up to use. 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.

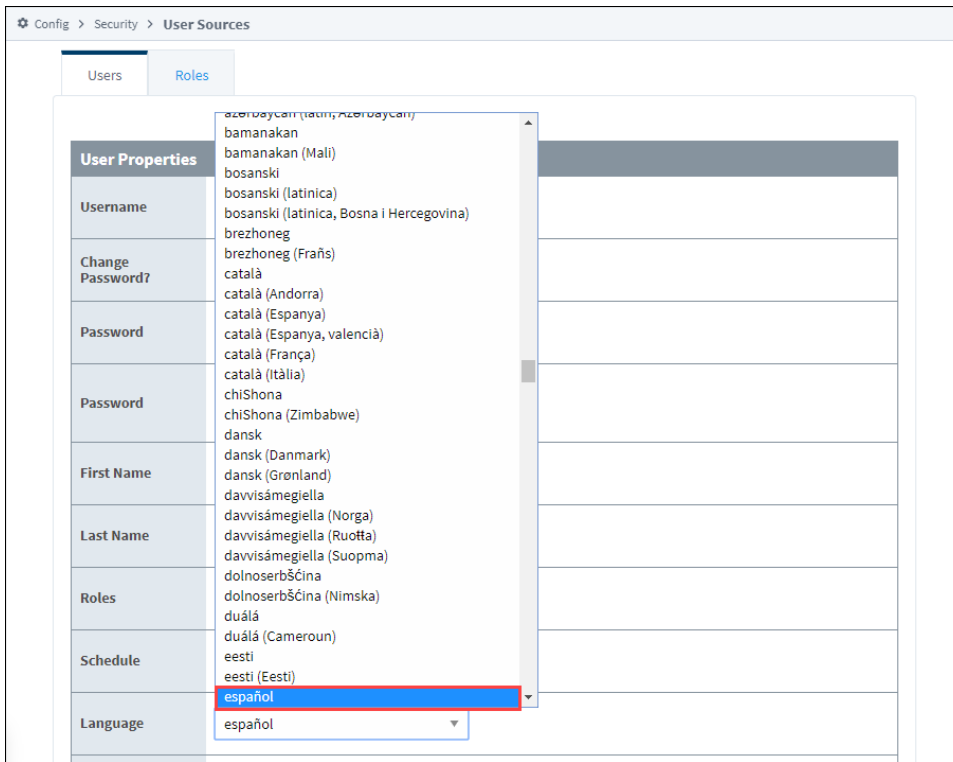
- Once you have the appropriate voice module and language module(s), go to **Security > Users, Roles**. Select the **manage users** link for the default user source.

Config > Security > 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 a User Source...](#)

- 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 Sal and set their **Language** to Spanish. If the voice module is installed for the Spanish language, Ignition will automatically choose the Spanish script.



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.

Registering Voice Alarm Module to CUCM

Cisco Unified Communications Manager (CUCM) is a communication system, that can work in tandem with Ignition's Voice Notification module. This page demonstrates how to register the Voice Notification module with CUCM.

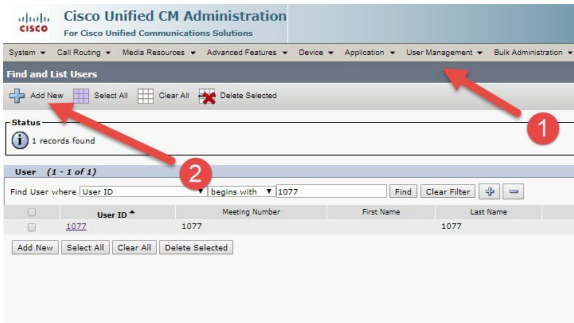
Registering Guide to CUCM

To register Ignition, you will need to create a third-party Session Initiation Protocol (SIP) device. Do not register Ignition via SIP trunk. Also note if you have created an SIP trunk already to your Ignition server, you will have a conflict with your third-party SIP device not registering to Ignition.

On this page ...

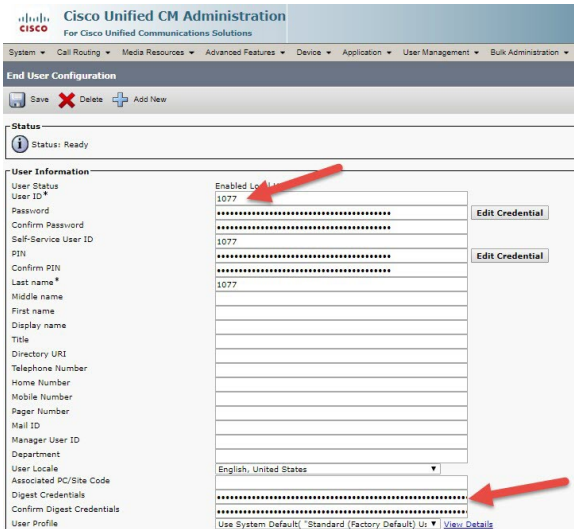
- [Registering Guide to CUCM](#)
- [CUCM and Redundancy Considerations](#)

1. In CUCM, Under **User Management**, select End User then click **Add New**.

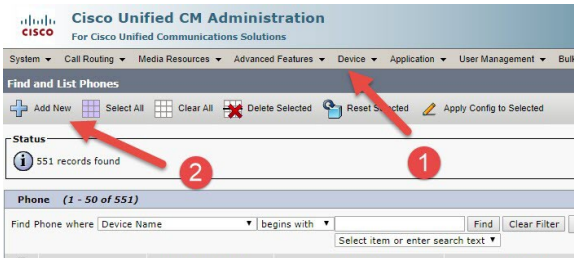


2. Create a user with digest credentials.

Note: Use the extension as the username. Ignition will not register to third-party SIP if you are using regular characters.



3. Under **Device**, select Phone. Click **Add New**.



4. Select **Third-party SIP Device (Basic)** as your phone type.

Cisco Unified CM Administration
For Cisco Unified Communications Solutions

System > Call Routing > Media Resources > Advanced Features > Device > Application > User Management > Bulk

Add a New Phone

Next

Status
Status: Ready

Create a phone using the phone type or a phone template

Phone Type* Third-party SIP Device (Basic)

or

BAT Phone Template* -- Not Selected --

Next

*- indicates required item.
**- Create a phone template using the Bulk Administration Tool to enable template-based phone creation.

5. For phone configuration, configure everything as usual. For the digest user, use the end user you created at the start.

Note: You do not need an actual Mac address, you can use 000000000000. Also you do not need an owner user ID. You would only need a digest user to register Ignition to CUCM.

Cisco Unified CM Administration
For Cisco Unified Communications Solutions

System > Call Routing > Media Resources > Advanced Features > Device > Application > User Management > Bulk

Phone Configuration

Save

Remote Number
Calling Party Transformation CSS < None >
 Use Device Pool Calling Party Transformation CSS (Device Mobility Related Information)

Protocol Specific Information

BLF Presence Group* Standard Presence group

MTP Preferred Originating Codec* 711ulaw

Device Security Profile* -- Not Selected --

Rerouting Calling Search Space < None >

SUBSCRIBE Calling Search Space < None >

SIP Profile* < None > [View Details](#)

Digest User < None > [Find](#)

Media Termination Point Required
 Unattended Port
 Require DTMF Reception

MLPP and Confidential Access Level Information

MLPP Domain < None >

Confidential Access Mode < None >

Confidential Access Level < None >

Save

6. For line configuration, configure how you would usually configure a line.

Note: You do not need a user associated to the line.

CUCM and Redundancy Considerations

CUCM is not able to handle two concurrent connections to the same phone type device using the same username/password combinations. Due to this limitation, redundant Ignition Gateways will fail to send voice notifications if both Master and Backup servers are simultaneously trying to connect to the same CUCM phone device. To get around this problem, create two users and two phone devices CUCM side. On your Ignition Master Gateway, create two Voice Notification Profiles. One Notification Profile will be bound to the Master's IP address and be configured to point to one CUCM phone device using one set of user credentials. The second Notification Profile will be bound to the Backup's IP address and be configured to point to the second CUCM phone device using the second set of user credentials. This will provide a Voice Notification Profile that will only work on your Master Gateway and another that will only work on your Backup Gateway. Your Alarm Pipelines must then include additional logic to switch between Voice Notification Profiles, or Notification Blocks, depending on which redundant node is active.

Related Topics ...

- [Voice Notification Scripts](#)

Remote Gateway Notification

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 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. .

On this page ...

- [Connect to a Remote Gateway and Select a Profile](#)
- [Remote Gateway Notification Settings](#)
 - [Remote Gateway](#)
 - [Advanced](#)
- [Enable the Remote Alarm Notification](#)

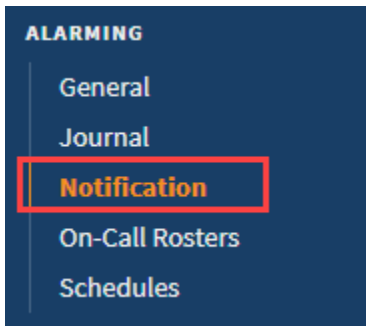


Remote Alarms

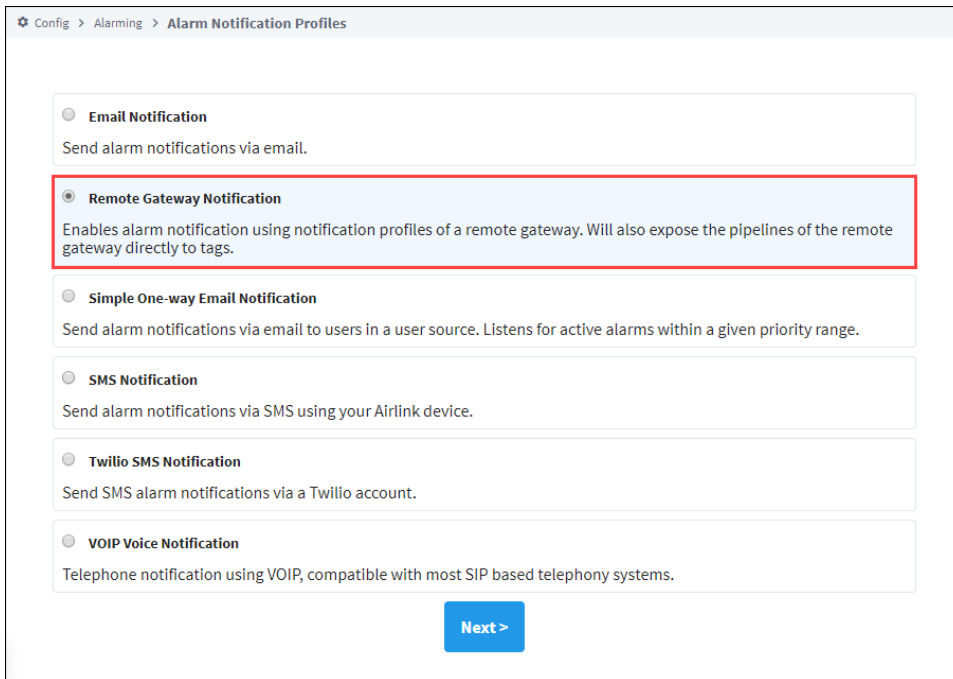
[Watch the Video](#)

Connect to a Remote Gateway and Select a Profile

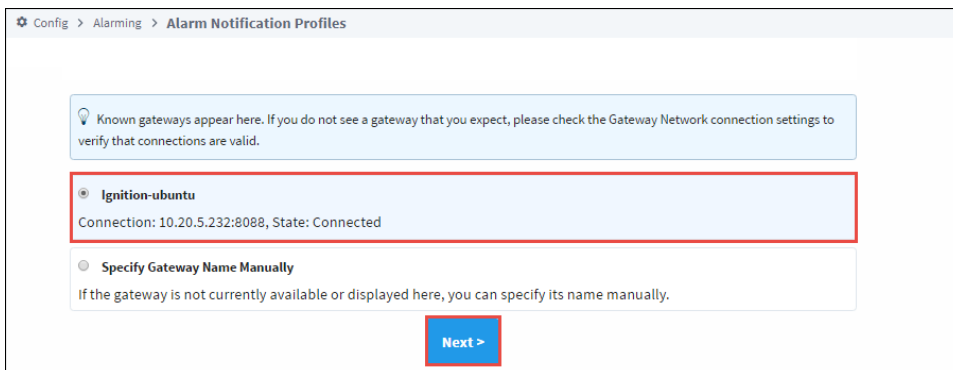
1. Go to the **Config** section of the Gateway Webpage.
2. Scroll down and select **Alarming > Notification**.



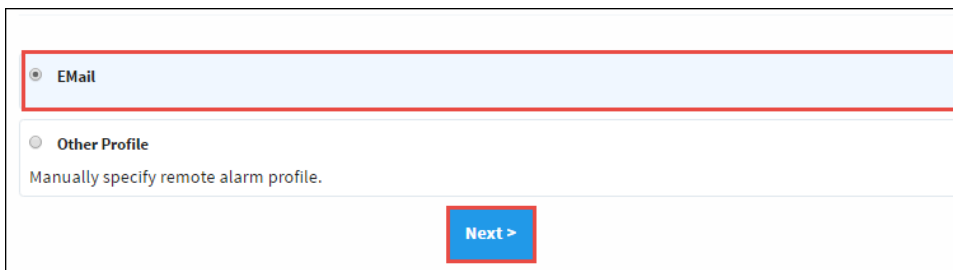
3. Click **Create new Alarm Notification Profile**.
4. Select **Remote Gateway Notification**, and click **Next**.



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



- 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**.



- 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**.

Config > Alarming > Alarm Notification Profiles

Main

Name	<input type="text" value="Ignition-ubuntu_Email"/>
Description	<input type="text"/>
Enabled	<input checked="" type="checkbox"/> (default: true)

Remote Gateway

Gateway	<input type="text" value="Ignition-ubuntu"/>
Notification Profile	<input type="text" value="EMail"/>

Show advanced properties

Create New Alarm Notification Profile

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

Config > Alarming > Alarm Notification Profiles

Name	Description	Enabled	Type	Status	
Email Notifications		true	Email Notification	Running	delete edit
Ignition-ubuntu_Email		true	Remote Gateway Notification	Running	delete edit
Test		true	Email Notification	Running	delete edit
Voice		true	VOIP Voice Notification	Registered with VOIP Host	More edit

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

Remote Gateway Notification Settings

Remote Gateway

Property Name	Description
Gateway	The name of the remote gateway that contains the desired Alarm Notification Profile. The name in this field should match the name of the gateway as is reported in the Gateway Network incoming/outcoming connections.
Notification Profile	The name of the Alarm Notification Profile on the remote Gateway.

Advanced

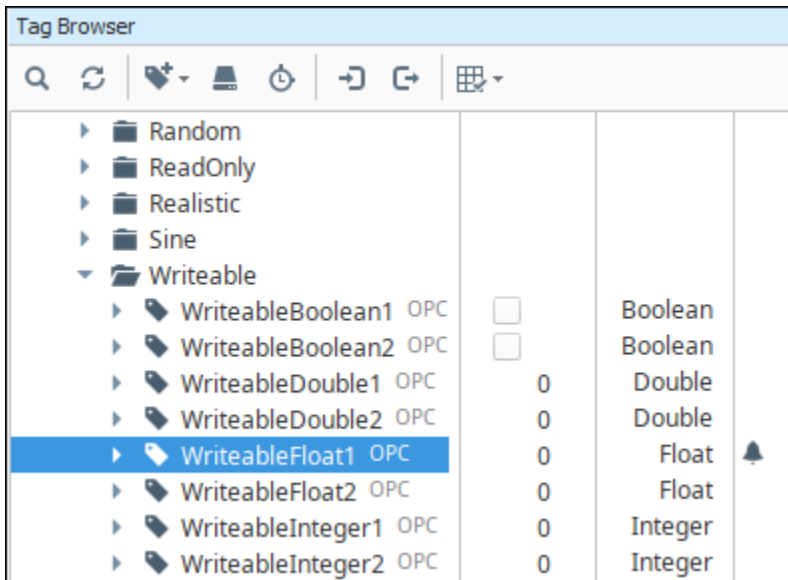
Property Name	Description
Retry Delay	Delay, in milliseconds, between retries of a failed attempt to synchronize with the remote profile. Default value is 10,000.


	<p><u>This feature was changed in Ignition version 8.1.18:</u></p> <p>In cases where multiple Remote Notification Profiles are targeting the same remote Gateway, the highest Retry Delay value amongst the Remote Notification Profiles will be used.</p>
Max Queued Alarm Transitions	<p>Maximum size of the transition evaluation queue before the oldest entries are start getting discarded. This queue is only expected to fill while the connection to the remote is not available. Default value is 10,000.</p> <p><u>This feature was changed in Ignition version 8.1.18:</u></p> <p>In cases where multiple Remote Notification Profiles are targeting the same remote Gateway, the highest Max Queued Alarm Transition value amongst the Remote Notification Profiles will be used.</p>

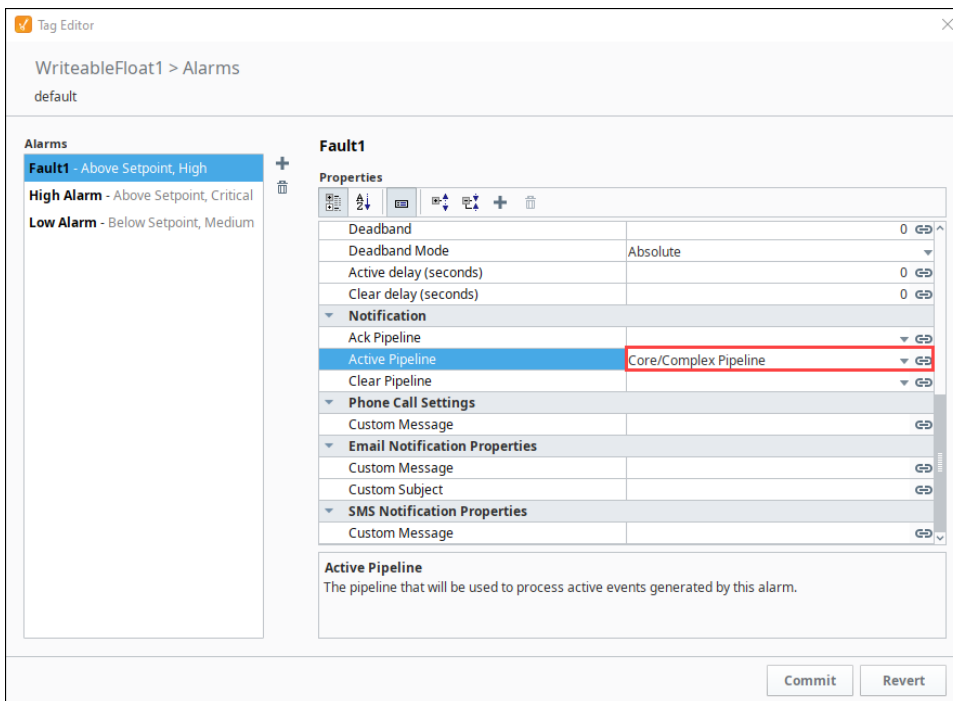
Enable the Remote Alarm Notification

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, scroll down to Alarms and click the **Edit**  icon .
3. 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.](#))
4. Scroll down to the **Notification** area and under **Active Pipeline**, select the remote pipeline (i.e., Core/Complex Pipeline) that you want to use from the dropdown list.
5. Click **Commit**.



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.



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

Related Topics ...

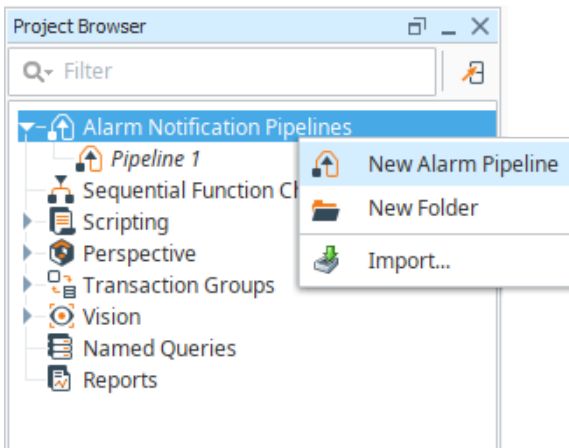
- [Alarm Notification Pipelines](#)

Alarm Notification Pipelines

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.



On this page ...

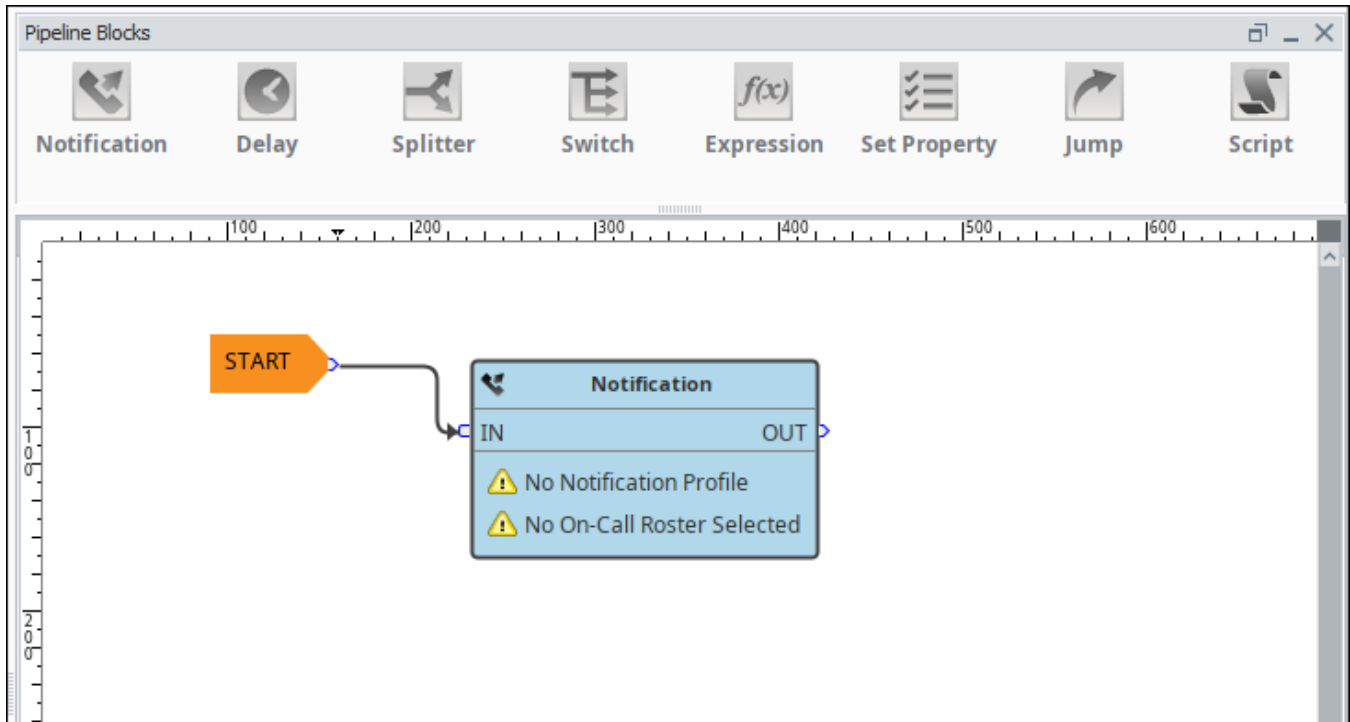
- [Creating Pipelines](#)
- [Understanding Pipelines and Events](#)
 - [Event Instances and Runtime Properties](#)
 - [Pipeline Lifecycle](#)
- [Pipeline Blocks](#)
- [Pipeline Properties](#)
 - [To Display and Modify the Properties](#)
 - [Commonly Used Properties](#)



Alarm Pipeline Overview

[Watch the Video](#)

When you create a new pipeline, or select one to edit, the Designer switches to the Pipeline Workspace. The workspace is 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

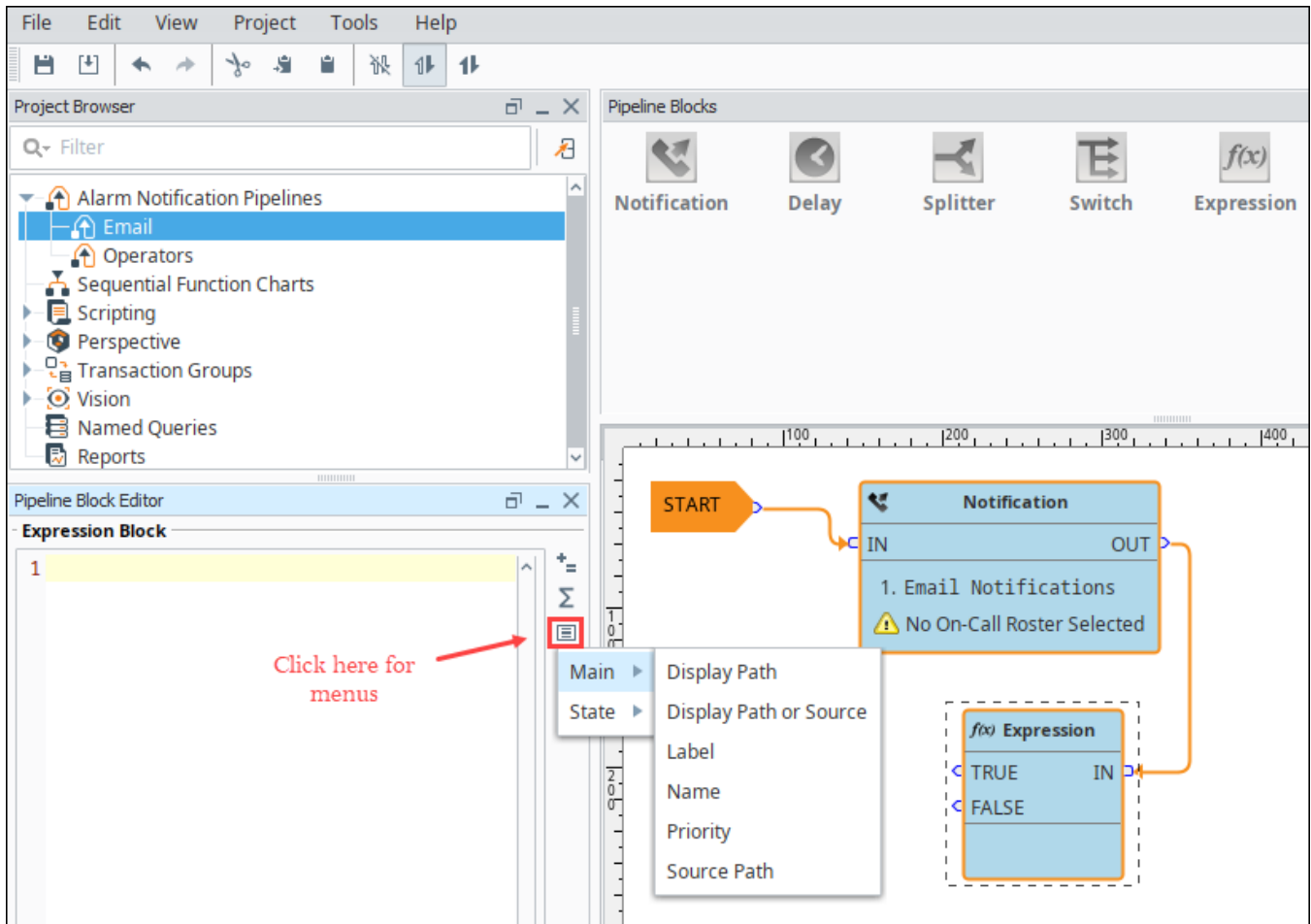
Click on the gray area of the Pipeline workspace anywhere there isn't a block.

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. The dropout condition is checked while transitioning between different blocks. In addition, the dropout condition is checked in-between notifications, so if the condition is met while a notification block is partway through notifying users in a roster, no further users in the roster will receive a notification.
- **Pipeline Enabled**
Indicates whether or not the pipeline is allowed to receive events.

Commonly Used Properties

When an Expression Block is selected, you can access two commonly used sets of properties by clicking the Alarm Properties  icon on the right side of the Pipeline Block Editor. These two properties are called the **Main** and **State** properties. The Alarm Properties  icon is available for all blocks that have expression fields.



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

Alarms 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.

In This Section ...

Alarm Pipeline Designer Interface

The Alarm Pipeline Designer Interface is where you create your own custom pipelines to control the the flow of an alarm to its notification. It provides all the functionality to create tasks that customize how alarms going through the pipeline notify people of the alarm.

The Pipeline Designer Interface is an easy to use graphical mechanism, and uses the drag and drop method for building complex alarm notification scenarios. Like other Ignition module interfaces, the Pipeline Designer Interface is organized into panels, some of which are specific to the Pipeline Designer Interface such as the **Pipeline Blocks** and the **Pipeline Block Editor**, and some you have seen before, such as the Project Browser and Tag Browser.

On the Alarm Notification Pipeline Welcome tab, you'll notice that there are some basic template pipelines to help get you started creating your own alarm notification pipelines. You can choose from the Priority and Escalation pipelines, or have the option to start from the Blank pipeline. These pipelines provide a basic skeleton for you to get started creating a priority or escalation pipeline which are common pipelines for alarm notifications. Once you have your several pipelines created, it will also show you the most recently modified pipelines along with the date it was last modified and who modified it. You can double click on a recently modified pipeline to easily open and update it.

The Alarm Notification Pipeline Welcome tab provides a quick way to create a new custom pipeline and update existing ones.

On this page ...

- [Project Browser](#)
- [Pipeline Blocks](#)
- [Pipeline Block Editor](#)
- [Alarm Pipeline Design Area Example](#)

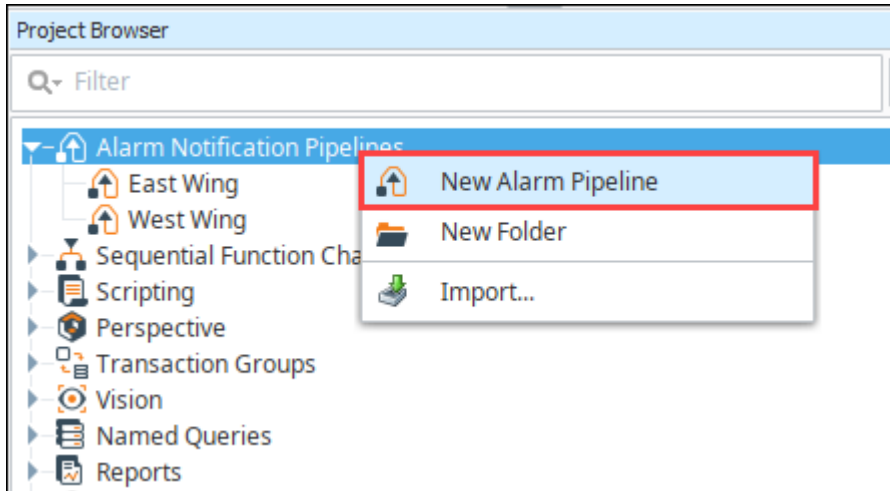
The screenshot displays the Alarm Pipeline Designer Interface with the following components:

- Project Browser:** A tree view on the left showing a hierarchy of folders including 'Alarm Notification Pipelines', 'Sequential Function Charts', 'Scripting', 'Gateway Events', 'Project Library', 'Perspective', 'Transaction Groups', 'Vision', 'Named Queries', 'Reports', and 'Web Dev'.
- Pipeline Blocks:** A toolbar at the top right containing icons for 'Notification', 'Delay', 'Splitter', 'Switch', 'Expression', 'Set Property', 'Jump', and 'Script'.
- Pipeline Block Editor:** A panel at the bottom left with the text 'Select a Block'.
- Alarm Notification Pipelines:** The main workspace area containing:
 - A 'Create a New Pipeline' section with a text input field for 'Name of the pipeline' and three buttons: 'Blank Pipeline', 'Priority Pipeline', and 'Escalation Pipeline'. A 'Create' button is located below these options.
 - A 'Recently Modified Pipelines' section featuring a table with the following data:

Name	Last Modified	Modified By
Blank	Sep 3, 2020, 4:22:17 PM	admin

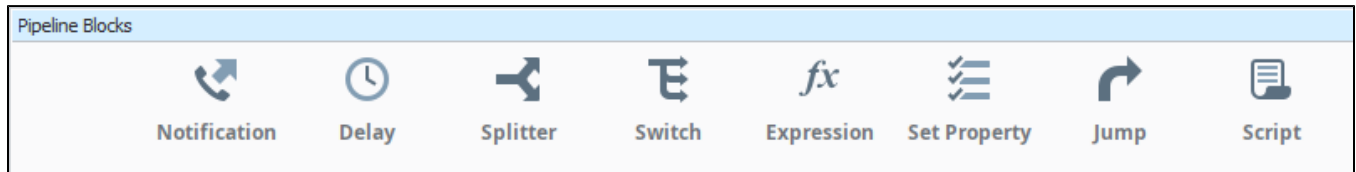
Project Browser

You'll find alarm pipelines in the **Project Browser** panel which appears in the top left corner of the Designer. Here you can create new alarm pipelines, organize your pipelines into folders, and import pipelines from other projects. Like other resources, [pipelines are inheritable](#), so a pipeline can be configured in one project, and inherited by another.



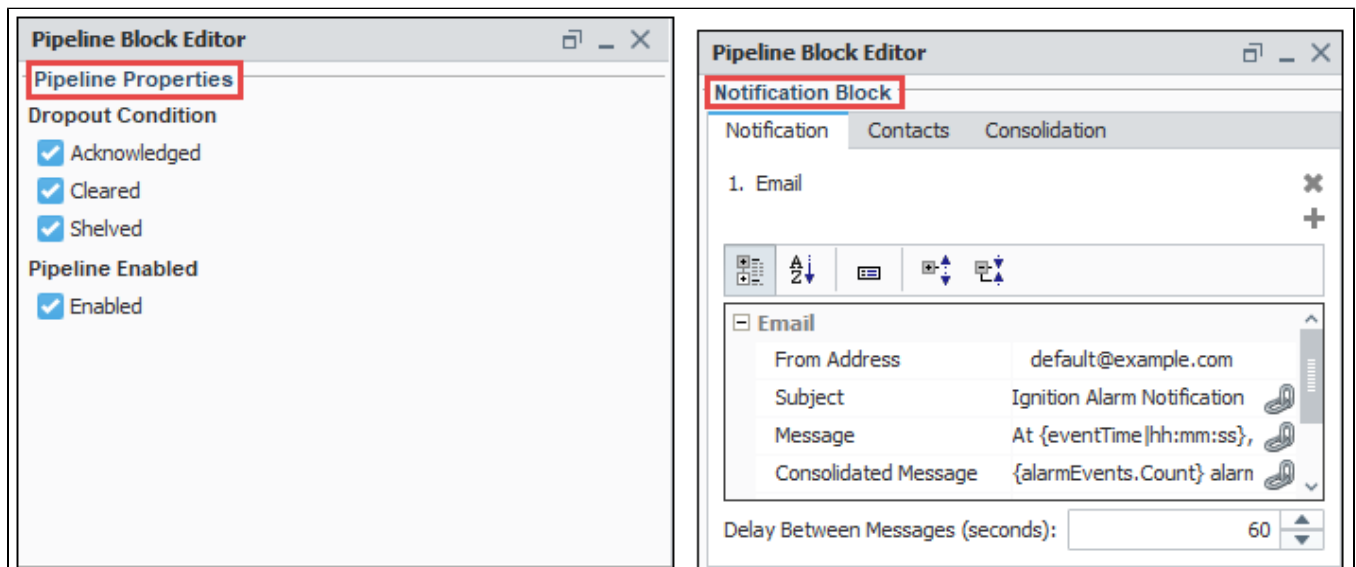
Pipeline Blocks

Pipeline Blocks are the building blocks of the Alarm Pipelines and are located at the top of the Designer. Each pipeline block has unique functionality that performs tasks such as sending notifications, alarm escalation, setting a delay, and evaluating an expression. Simply drag any of the pipeline blocks to your design area and configure each pipeline block's properties.



Pipeline Block Editor

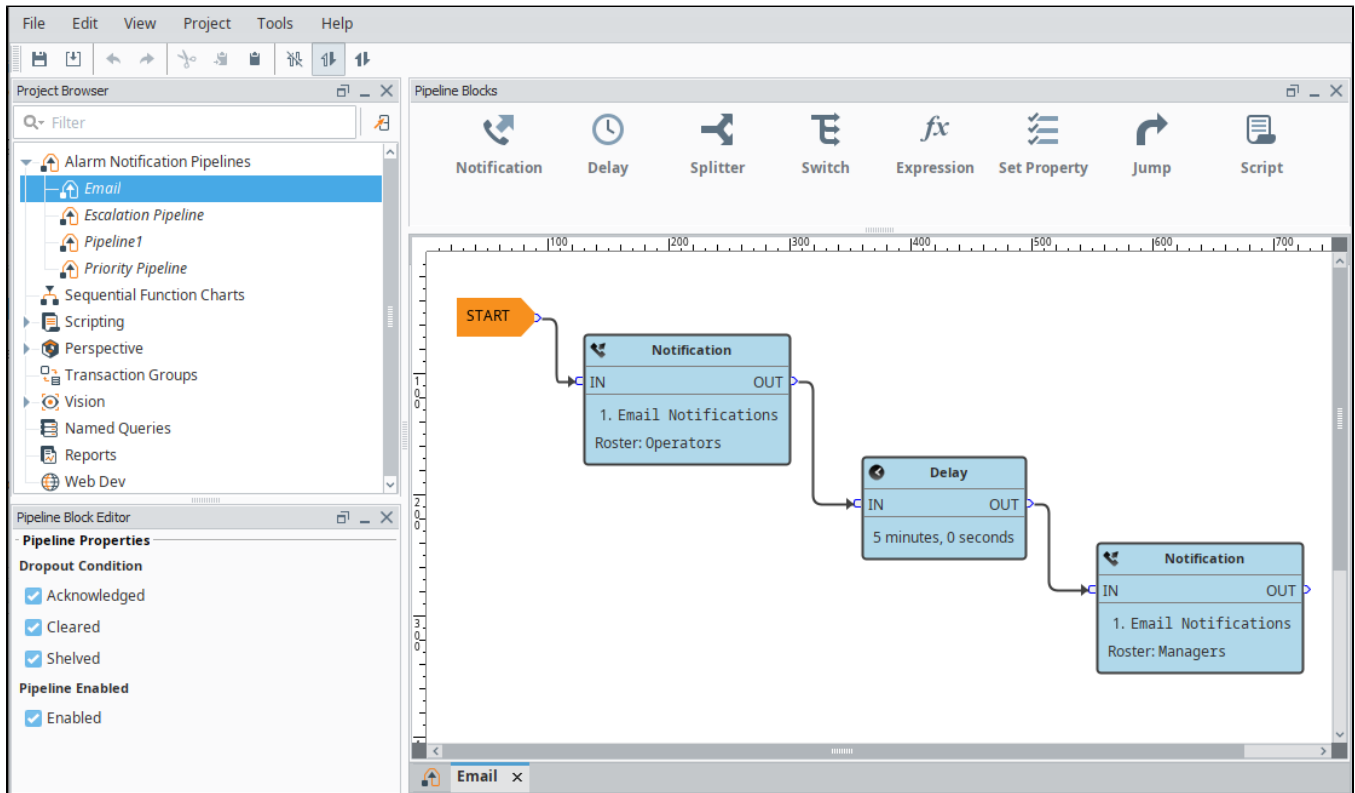
The Pipeline Block Editor displays the pipeline properties and properties of the selected pipeline block. To see pipeline properties, click anywhere in the workspace. Since each pipeline block performs a specific function, it also has its own properties. Select the pipeline block to see its properties. The image below shows both the Pipeline's properties and a Notification block's properties.



Alarm Pipeline Design Area Example

In this example, we will use a Blank pipeline to create an Email pipeline. It will be set up so that when an alarm occurs, a notification will be sent to Operators via email. If an operator doesn't respond to the notification within 5 minutes, a second email will be sent Managers. This assumes an Operator and Managers on-call rosters have been configured.

1. On the Blank pipeline design area (with the START block prepopulated), drag the appropriate pipeline blocks to your canvas to begin building the logic for your notification pipeline. The blocks for this example include two Notification blocks and a Delay block.
2. Set the properties for each pipeline block. The first Notification block needs to send email notifications to the Operators roster. The Delay block time needs to be 5 minutes. The second Notification block needs to send email notifications to the Managers roster.
3. Connect the blocks from the START block, to the first Notification block, to the Delay block, and end at the second Notification block to create your pipeline.



Simple Pipeline

The most basic Alarm Notification Pipeline will simply notify users as soon as the [alarm](#) becomes active.

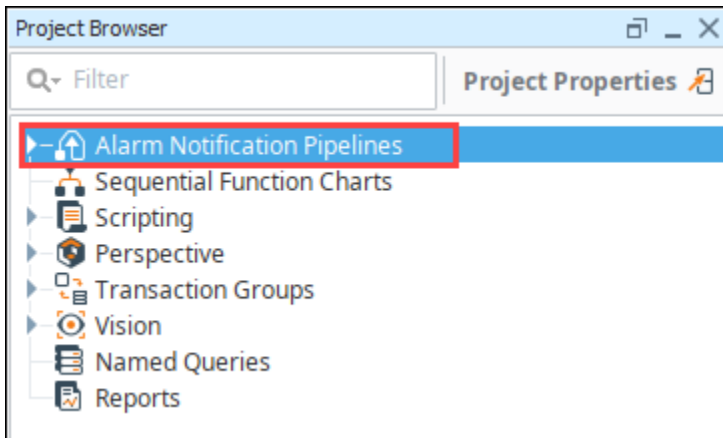
Configure Alarm Notification Profile and On-Call Roster

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.

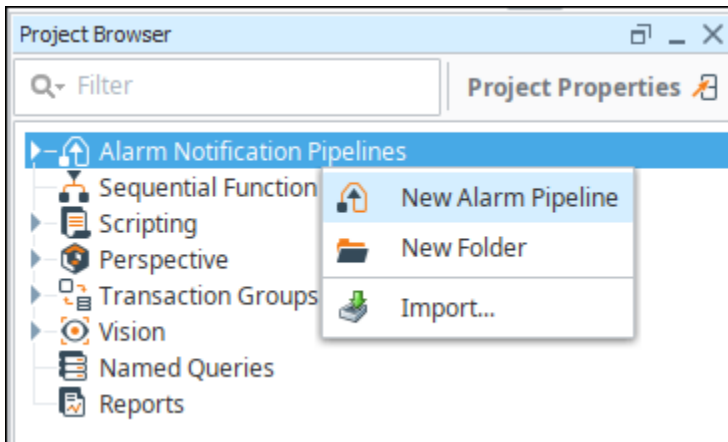
Create an Alarm Notification Pipeline

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

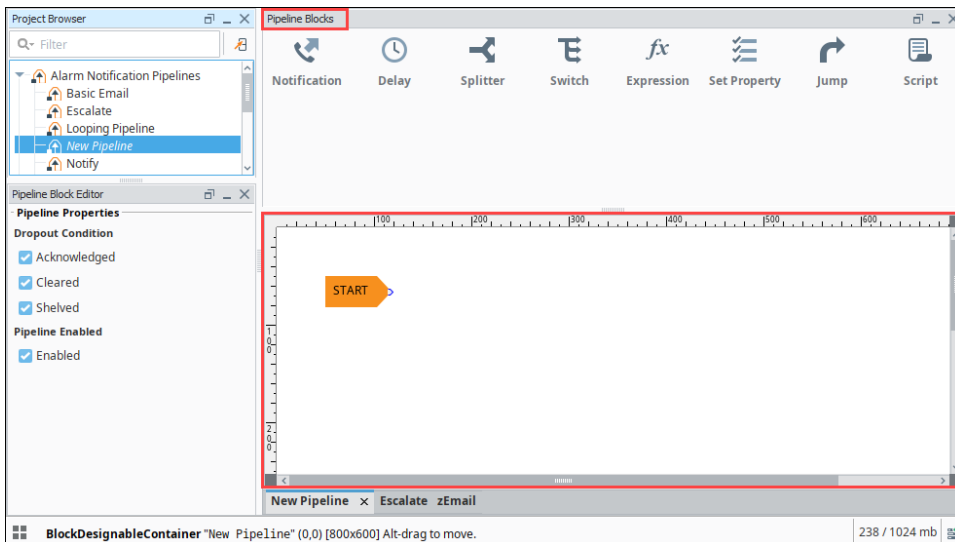
1. In the **Designer**, create a new pipeline in the **Project Browser** by going to the **Global** area.



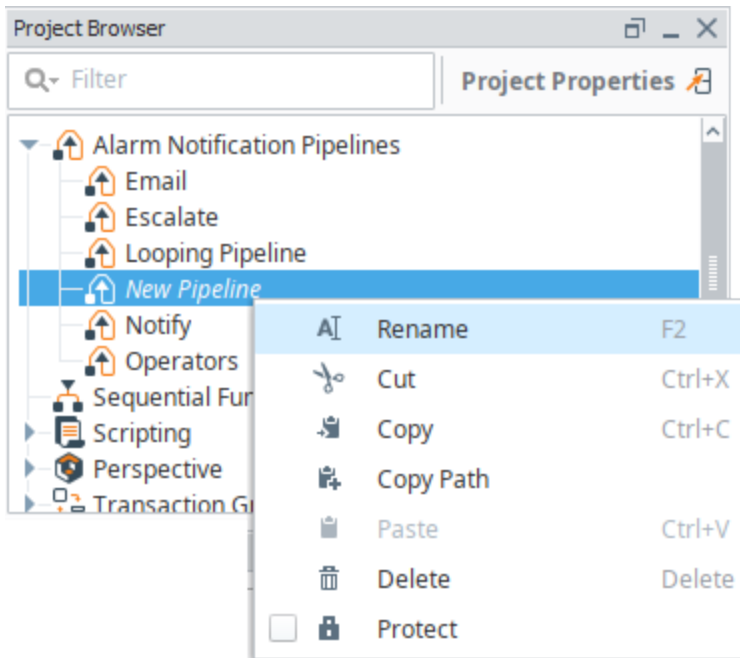
2. Select **Alarm Notification Pipelines**, and right click to create a **New Alarm Pipeline**.



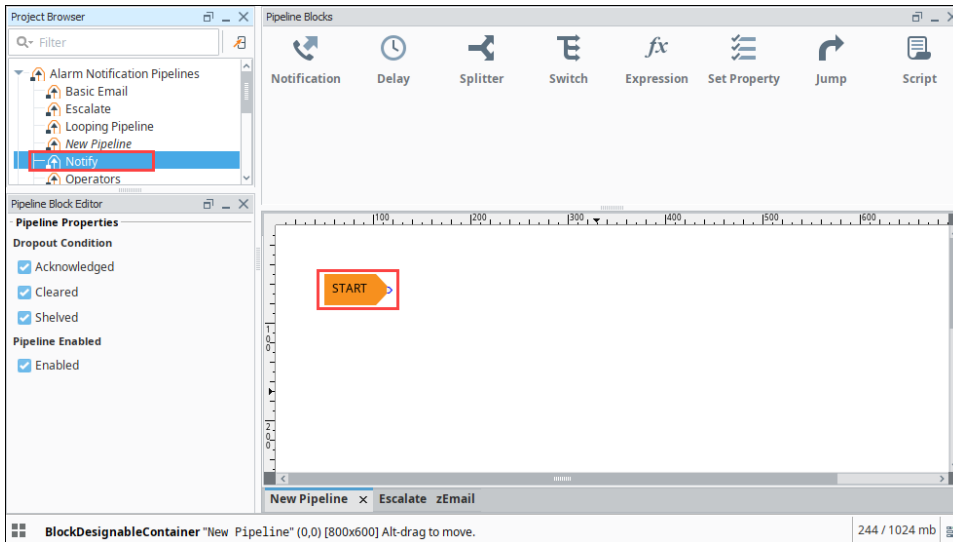
When you create a new pipeline, the Designer switches to the Pipeline workspace. The workspace is essentially a blank canvas and can contain various [Pipeline Blocks](#). By default, the name of the new pipeline is called '**New Pipeline**.'



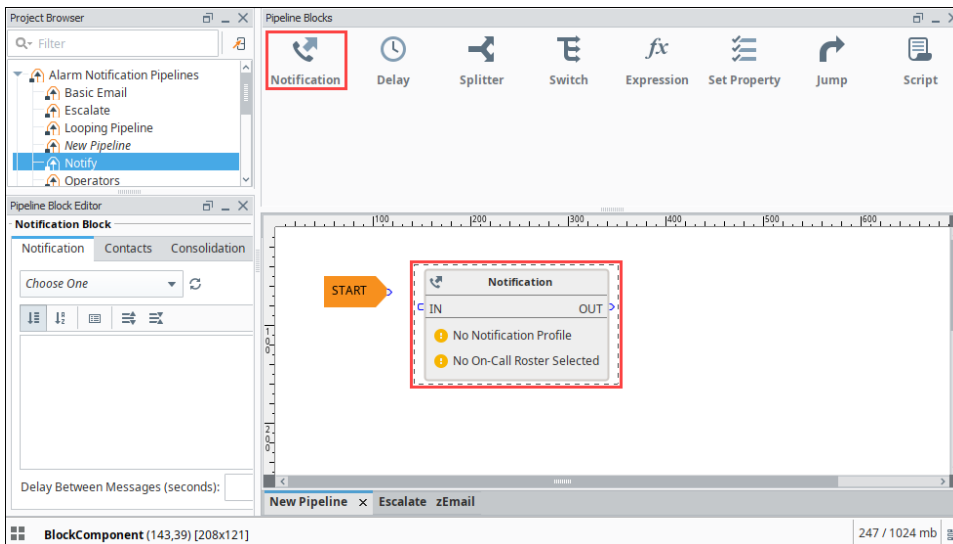
- It is a good practice to give a meaningful name to the pipeline. Commonly, the name acts as a way to describe what the pipeline does. Select the New Pipeline, right click and rename it "Notify".



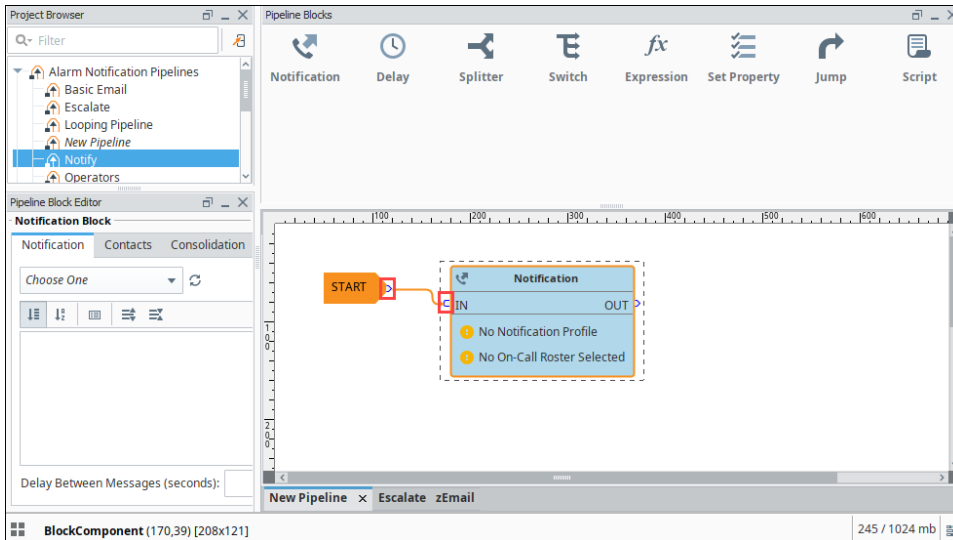
Every pipeline has a Start block which you will see when you create your pipeline. When an alarm is triggered, it automatically goes to the Start block, and from there you can use other pipeline blocks to determine what you want to do with that alarm. All the available pipeline blocks are shown at the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



- 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. Once you drag it in to your pipeline, it is not connected.

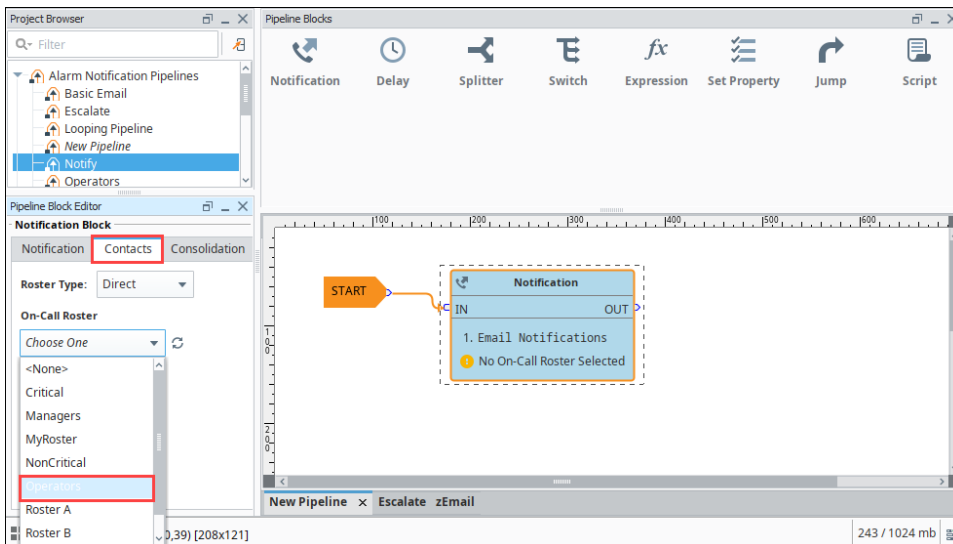


- Drag the output pin from the Start block to the input pin of the Notification block connecting the two blocks. When an alarm occurs, it's going directly into the Start block and then into the Notification block.



6. On the Notification block, you have to configure the [Alarm Notification Profile](#) that you are going use and who you are going to send it too. You should have already configured your Alarm Notification Profile before creating your new pipeline.

7. In your pipeline workspace, click on the Notification block, and the Notification tab will open.
8. 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**.
9. 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.

Pipeline Blocks

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:

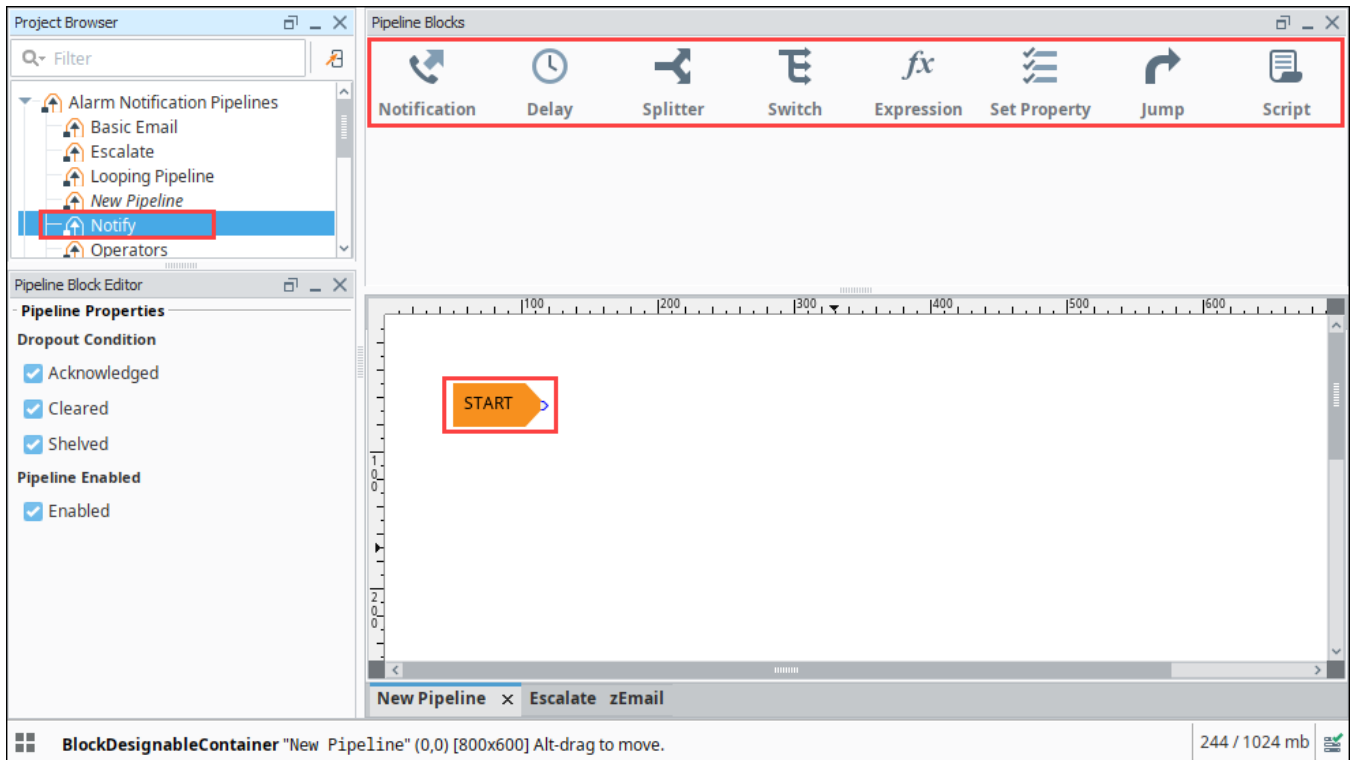
- **Notification**
Delivers a notification through the selected Notification Profile (i.e., Email, SMS and Voice).
- **Delay**
Blocks the alarm event for the specified amount of time.
- **Splitter**
Forwards a single event concurrently to multiple other blocks.
- **Switch**
Evaluates a non-boolean Expression, 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 property 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.

On this page ...

- [Pipeline Blocks](#)
- [Start Block](#)
- [Notification Block](#)
- [Delay Block](#)
- [Splitter Block](#)
- [Switch Block](#)
- [Expression Block](#)
 - [Examples](#)
- [Set Property Block](#)
 - [Examples](#)
- [Jump Block](#)
- [Script Block](#)



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 set up 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.

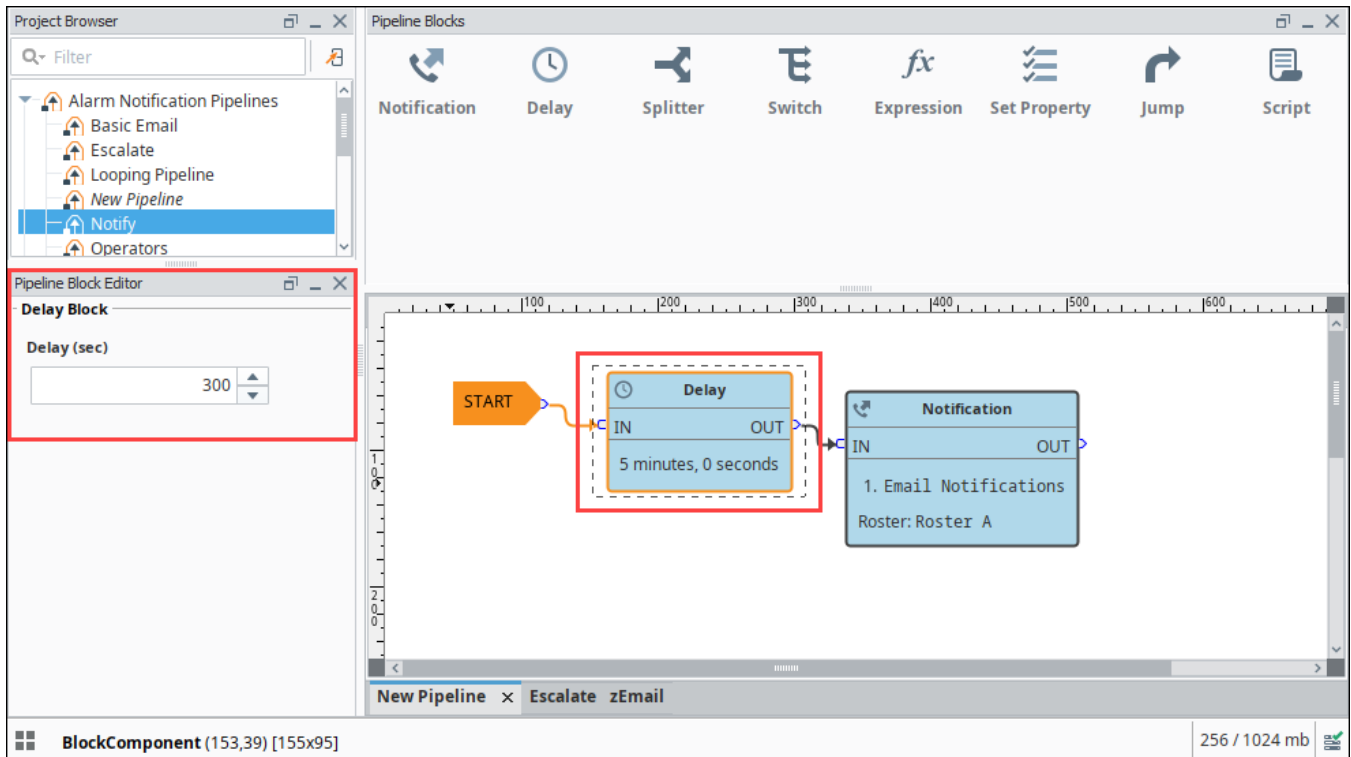
Delay Block

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).

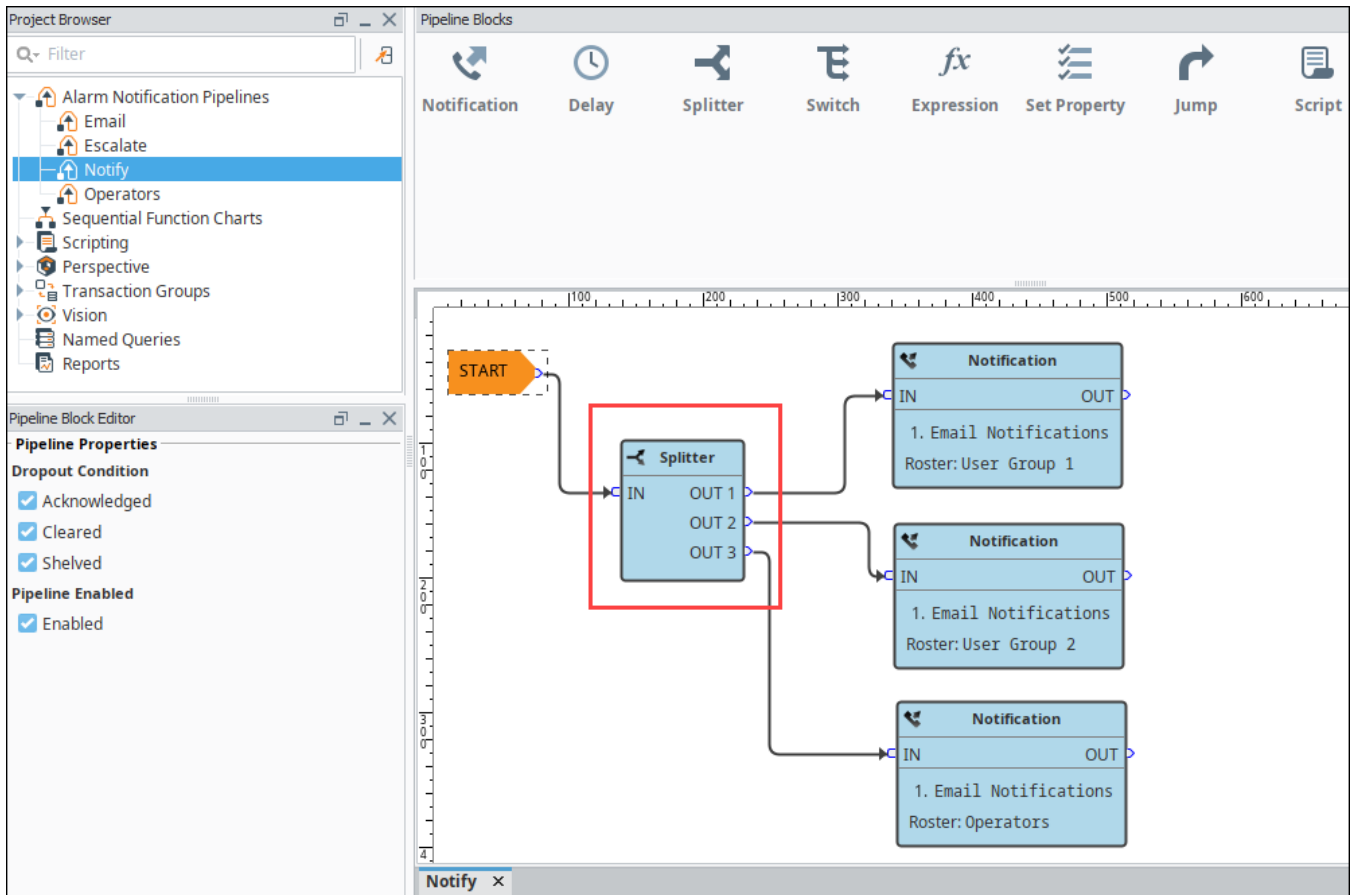
Delays are often also used to control flow in a pipeline. For example, in an "escalation" scenario, a notification block might be used to send emails. Since emails are sent instantly, and acknowledgment occurs "asynchronously" (the user must see the email and click the link or log into the system), a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



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.

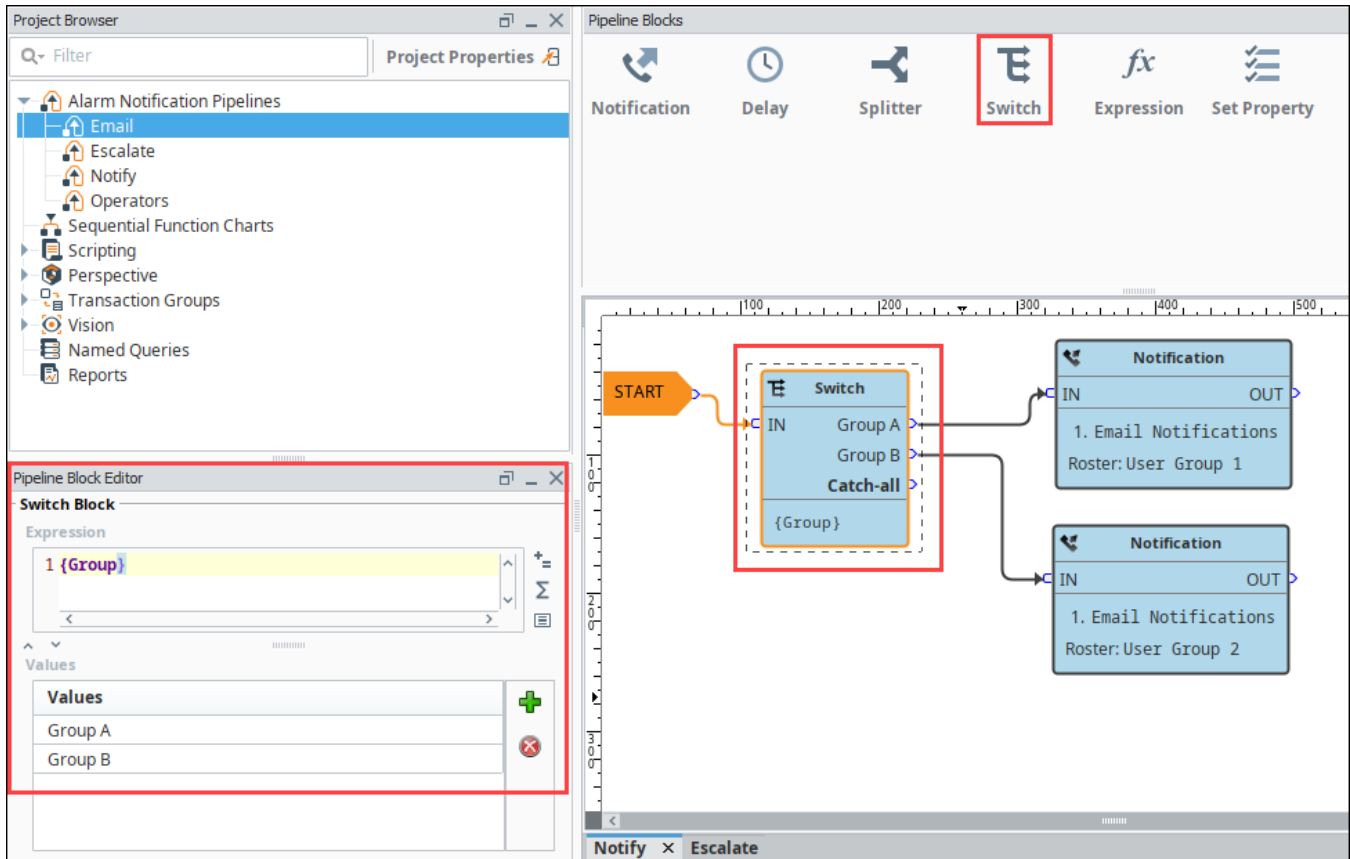


Switch Block

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 two 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.

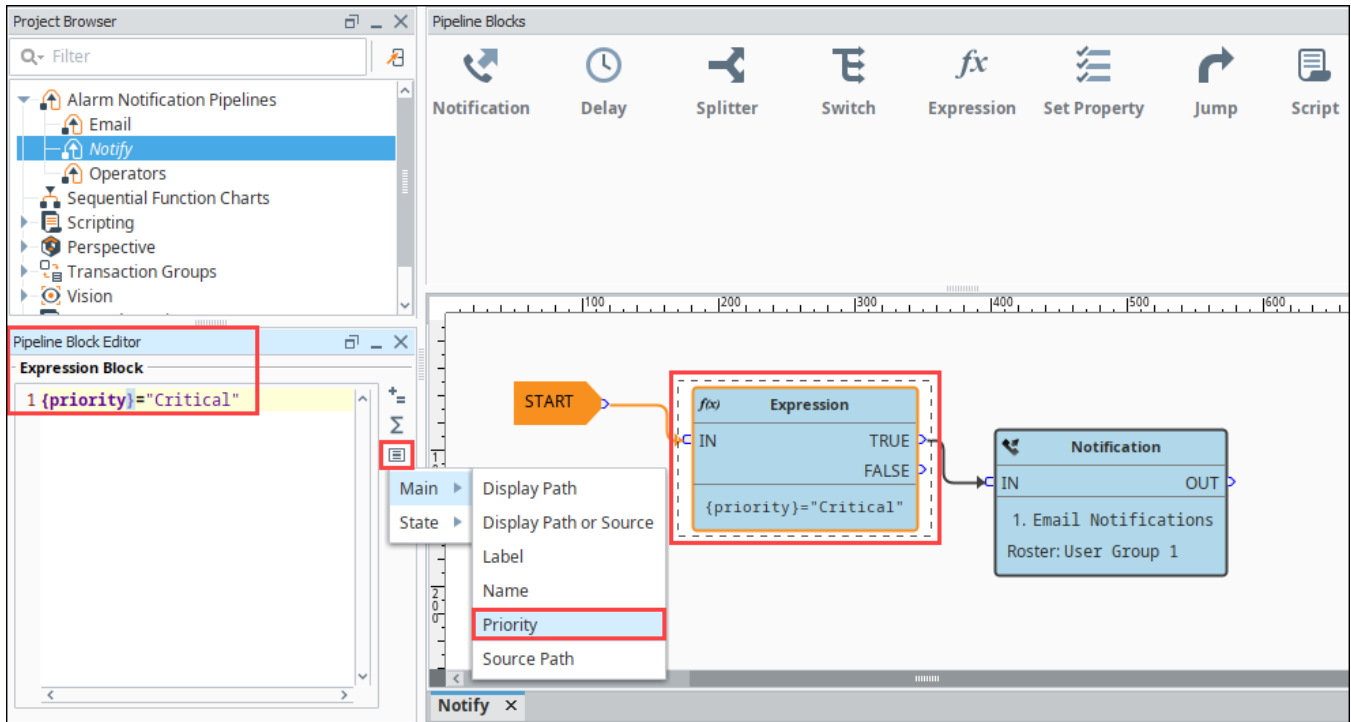
`isPropertyDefined("isEscalated")`

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 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.

Priority Expression Example

```
{priority}="Critical"
```



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`

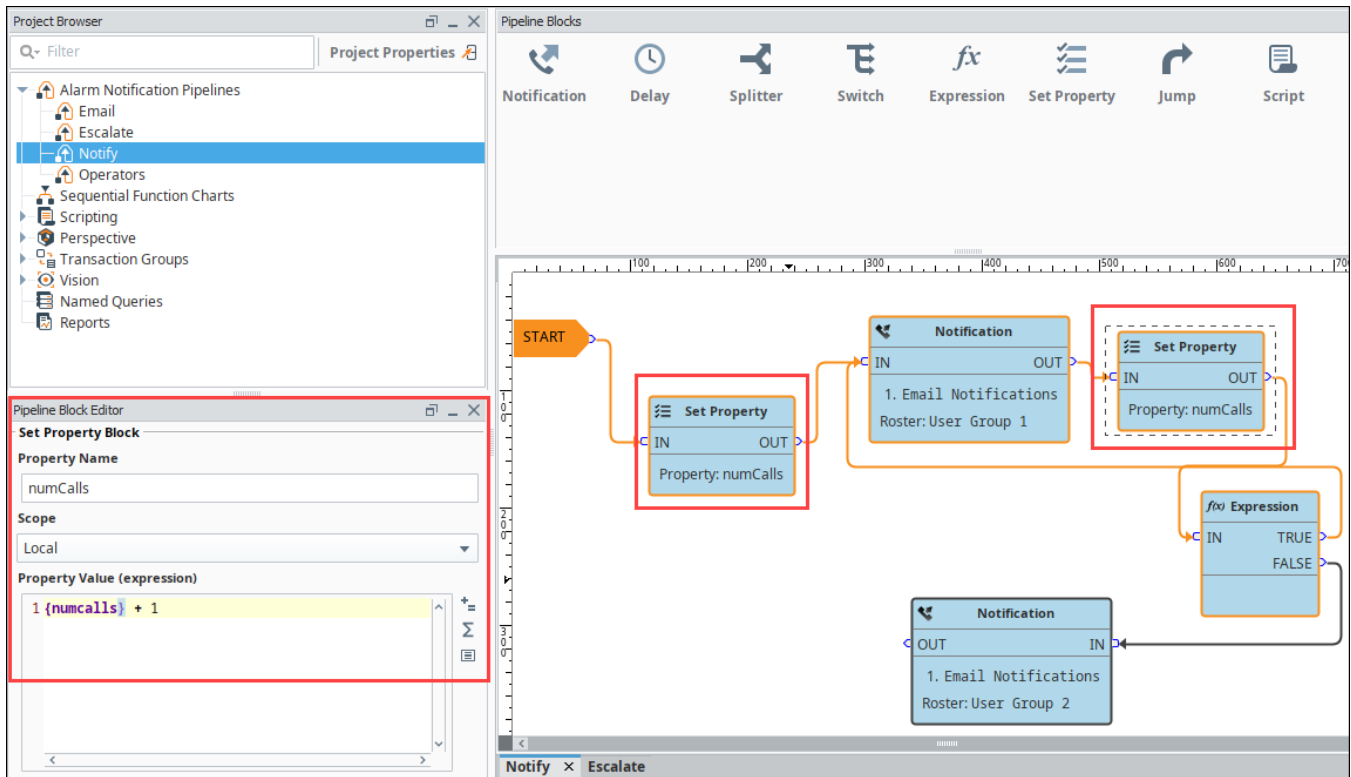
Note that the first time the block is hit, the `counter` property will not exist, and therefore the value will be null. The `coalesce` function returns the first non-null value, allowing us to start with 0. Another way to write this might be:

```
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 two 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.

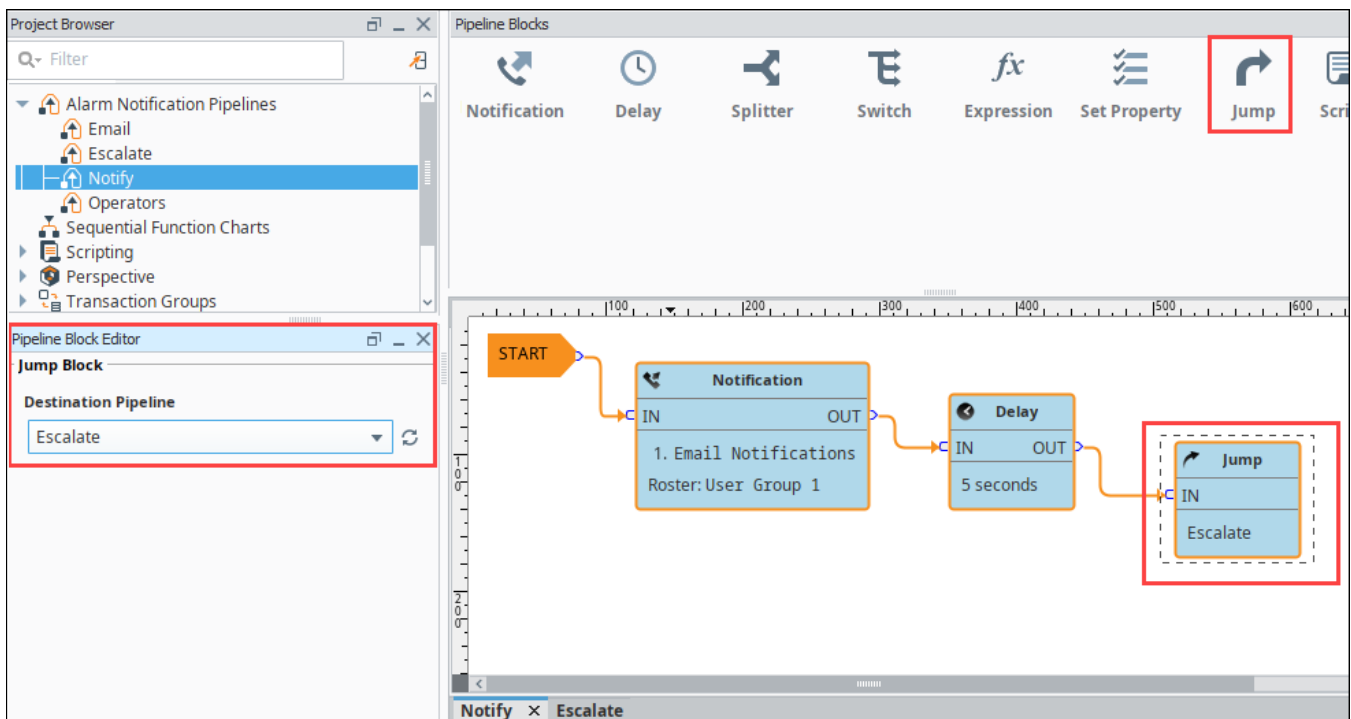


Jump Block

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.

The script block provides access to an event object, which is simply a `ScriptableBlockPyAlarmEvent` object.

The screenshot displays the Pipeline Block Editor interface. On the left, the Project Browser shows a tree view with 'Alarm Notification Pipelines' expanded to 'Email', and 'Notify' selected. The Pipeline Blocks toolbar at the top includes icons for Notification, Delay, Splitter, Switch, Expression, Set Property, Jump, and Script (highlighted with a red box). The main workspace shows a pipeline starting with a 'START' block, followed by a 'Notification' block (outputting '1. Email Notifications' and 'Roster: User Group 1'), a 'Delay' block (set to '5 seconds'), and finally a 'Script' block (highlighted with a dashed box). The Script block editor on the left contains the following code:

```
1 def handleAlarm(event):  
    """  
    Called for each alarm event that passes through the block.  
    event, call "event.cancelNotification()". Note: The event is  
    actually an "event instance", and properties set on it are  
    event while it is in the pipeline. To set properties that p  
    changes in state, use "event.setGlobal(name, value)".  
  
    Arguments:  
        event: The alarm event instance. Properties can be acce  
        created using dictionary syntax.  
    """  
2 system.tag.write({"default"}Alarms/ProdSupAttention", true)
```

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

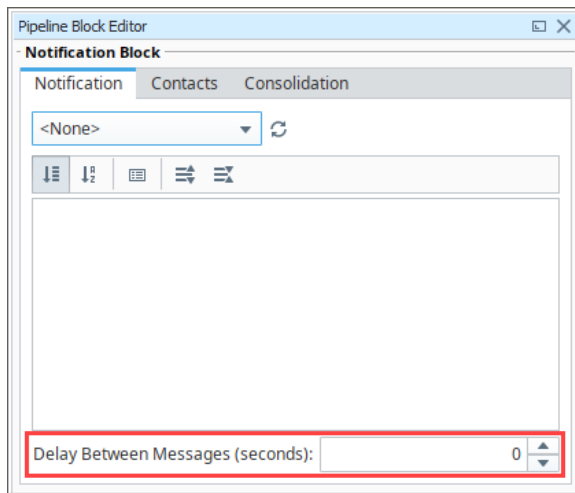
Notification Block

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 from what 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 notifications to each user in a roster. So if your delay time 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.



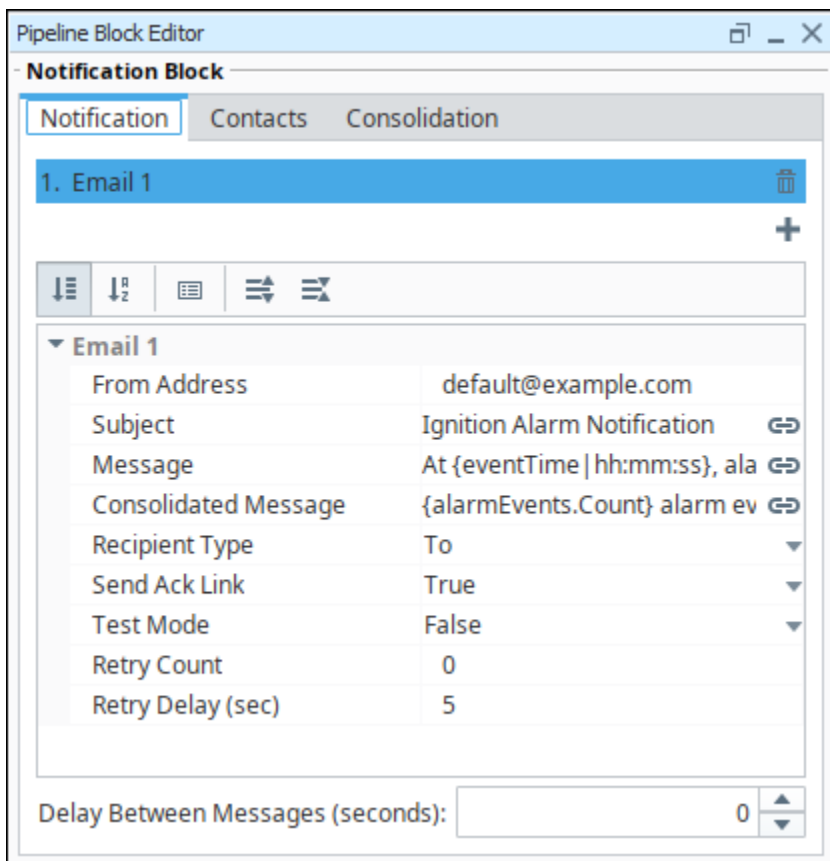
Email Settings

Setting	Description
From Address	Email address used for the From field.
Subject	Subject of the email.
Message	The body of the email. Like the subject, may refer to properties of the alarm. Message supports HTML formatting. Simply add the <html> tag at the beginning of the Message property. An end tag (</html>) is not required. <pre><html>This is my message.
That was a line break.</pre>
Consolidated Message	The message sent when consolidation is turned on, and more than one alarm is being sent. See the Notification Block#Consolidation section for more details.

On this page ...

- [Notification](#)
 - [Email Settings](#)
 - [Voice Settings](#)
 - [SMS Settings](#)
 - [Twilio Settings](#)
 - [Message Properties](#)
- [Contacts](#)
 - [Direct](#)
 - [Expression](#)
 - [Calculated](#)
 - [Ignore User Schedules](#)
- [Consolidation](#)
 - [Consolidated Message Property](#)
 - [HTML Formatting in Email Notifications](#)


Recipient Type	Determines if the email is sent to the user via "To", "CC", or "BCC".
Send Ack Link	Allows the notification block to override the acknowledgment link that's included with two way email notification , potentially hiding it. If Two-way notification is enabled on the email notification profile, and this property is false, then emails sent by the block will not include an acknowledgment link.
Test Mode	If enabled, logs to the console each time an email would be sent, instead of sending the actual email.
Retry Count	The number of retry attempts per email per alarm. Default is 0. This setting applies when there is an exception that occurs when attempting to send a message, such as network connectivity issues.
Retry Delay	Introduces a delay between each email for each alarm. Default is 5. This setting applies when there is an exception that occurs when attempting to send a message, such as network connectivity issues.



Voice Settings

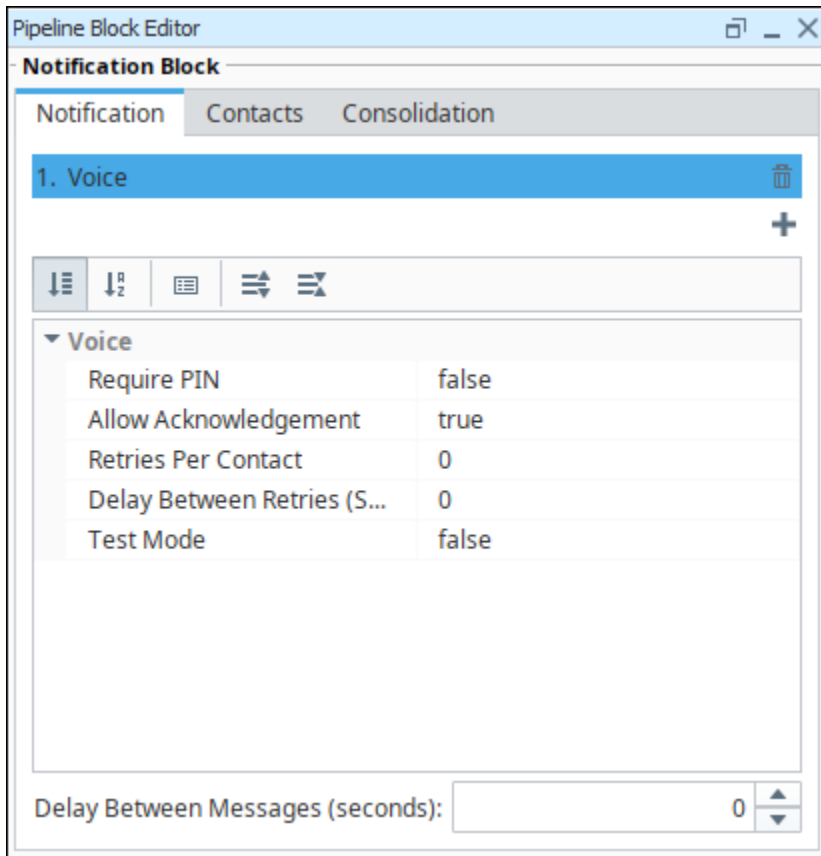
Setting	Description
Require PIN	If true, the user will need to enter their PIN in order to hear the alarm notification messages. The user's PIN is defined in the user management section of the Gateway . If false, then anyone will be allowed to hear the messages after answering the phone.
Allow Acknowledgment	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.
Delay Between Retries	Introduces a delay between each retried contact notification in a Roster for each alarm when the Retries Per Contact setting is greater than 0. This delay setting is independent from the Delay Between Messages setting. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Example Notification Delay Process If there are two Users in a roster and Retries Per Contact is set to 2, the notification process would function as follows:</p> <p>Contact User1 (wait Delay Between Retries time) > Retry User1 (wait Delay Between Retries time) > Retry User1 (wait Delay Between Messages time) > Contact User2</p> </div>

The delay is only enforced after a "successful" call, which means the call was answered. Unanswered or errored calls will move on to the next contact immediately.

 For most phone systems, a delay of at least 1 second is advised since the system must have time between each call to hang up properly before dialing again. Long delays can block other alarms in the call queue.

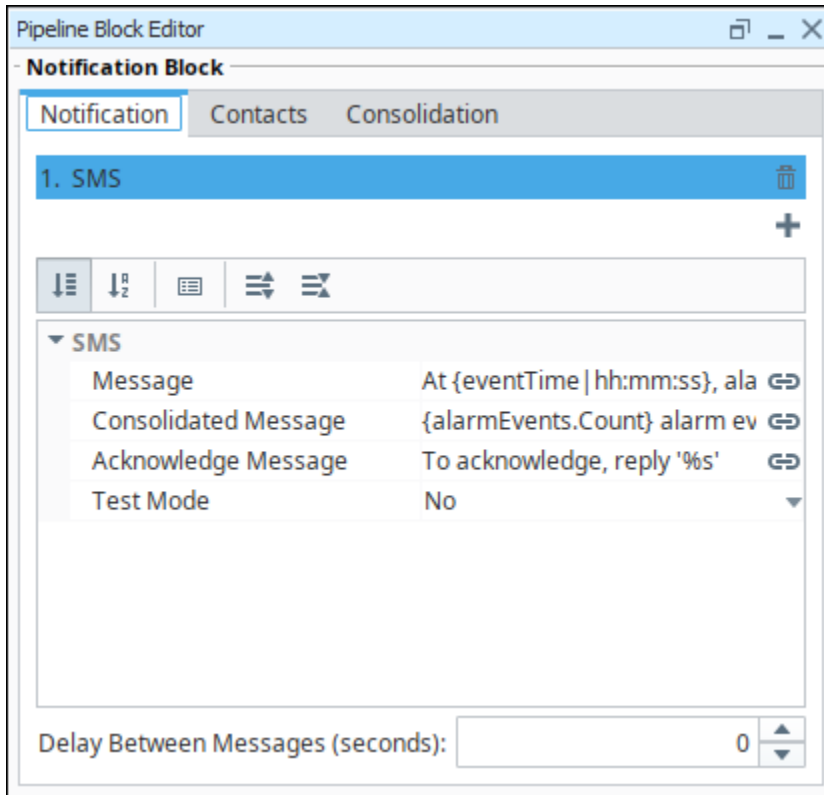
The pipeline dropout conditions are checked regularly between calls and while waiting, so this setting also provides time for the alarm to drop before the next call.

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.



SMS Settings

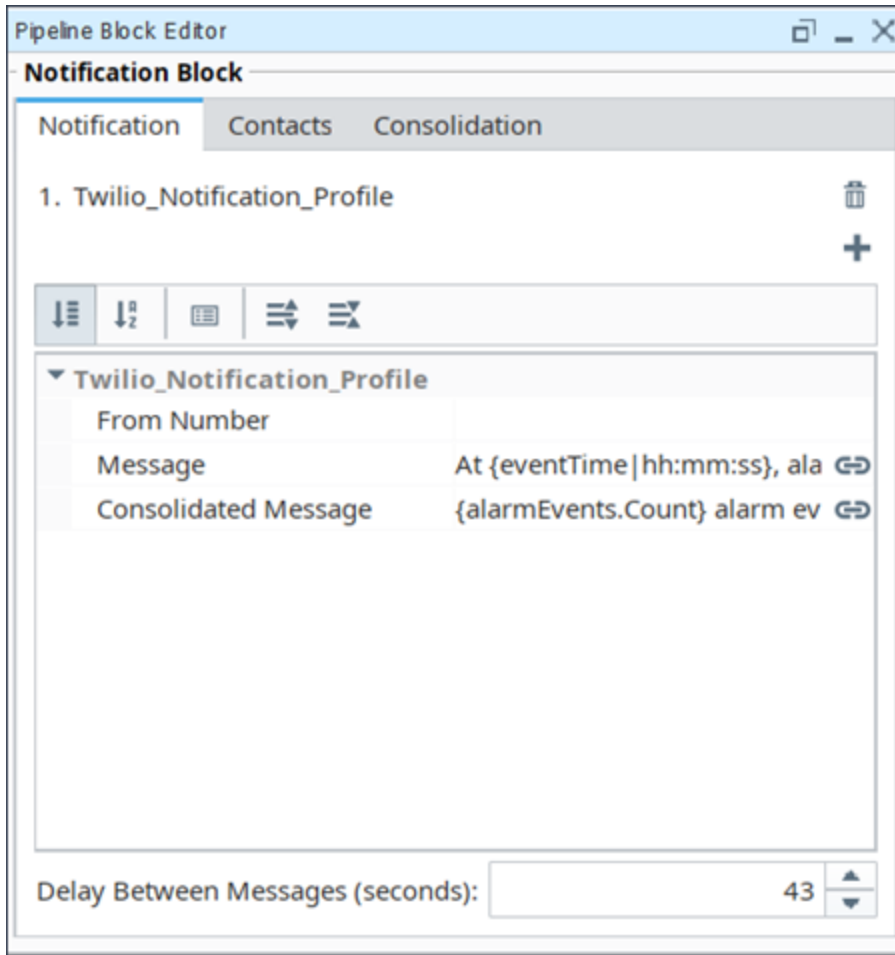
Setting	Description
Message	The message to use for single events.
Consolidated Message	The message sent when consolidation is turned on, and more than one alarm is being sent. See the Notification Block#Consolidation section for more details.
Delay Between Notification	As with voice, a delay between each message sent, to give the recipient time to acknowledge and prevent further notification.
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 acknowledgment code.
Test Mode	If enabled, logs messages that would have been sent to the Gateway console, and does not actually send them.



Twilio Settings

SMS notifications can also be sent with the [Twilio Alarm Notification](#) module. To use Twilio, you must first set up [Twilio SMS Notification](#). 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).

Setting	Description
From Number	Number from which the alarm notification is sent.
Message	Message to send. Default is " At {eventTime hh:mm:ss}, alarm "{name}" at "{displayPath}" transitioned to {eventState}." Click the binding icon to access other alarm properties that can be used to the message.
Consolidated Message	<div style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;"> <p>The following feature is new in Ignition version 8.1.2 Click here to check out the other new features</p> </div> <p>The message sent when consolidation is turned on, and more than one alarm is being sent. See the Notification Block#Consolidation section for more details. In release 8.1.2 this setting was renamed from "Throttled Message" to "Consolidated Message."</p>



Message Properties

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

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

Contacts

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 set up 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

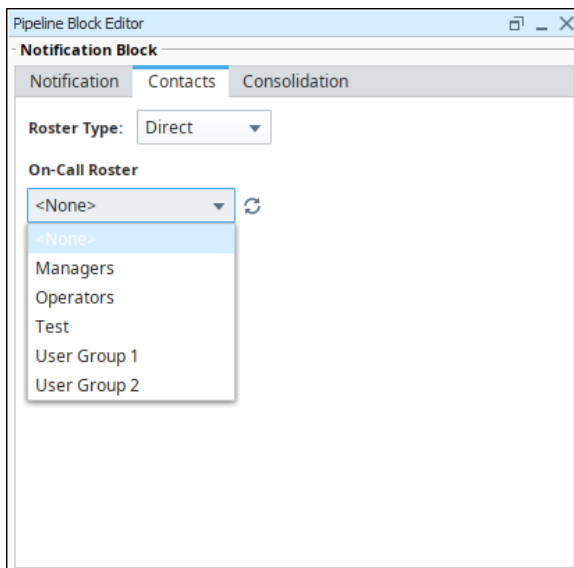
The Direct roster type is the simplest to set up and use, but also the least flexible. Any On Call Roster defined on the Gateway can be used directly.





**INDUCTIVE
UNIVERSITY**

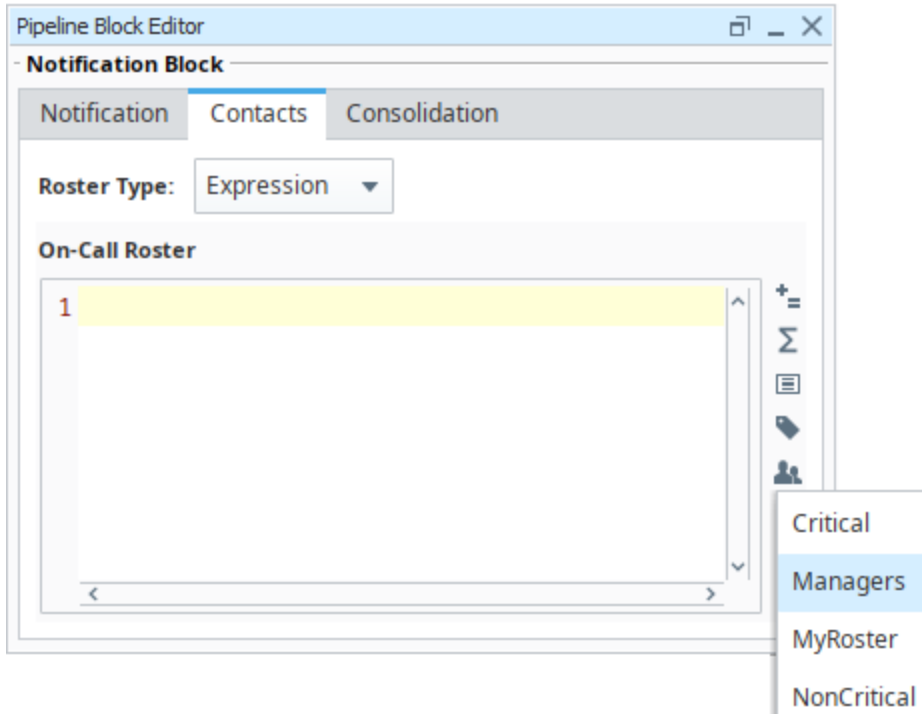
Dynamic Rosters

[Watch the Video](#)



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**  icon 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**  icon.



The above expression 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.

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.

However, quotation marks could be removed in this scenario. 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
```

However, 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 priority was "critical", then the **High Priority Roster** would be notified. For all other priorities, the **Low Priority Roster** would be used.

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

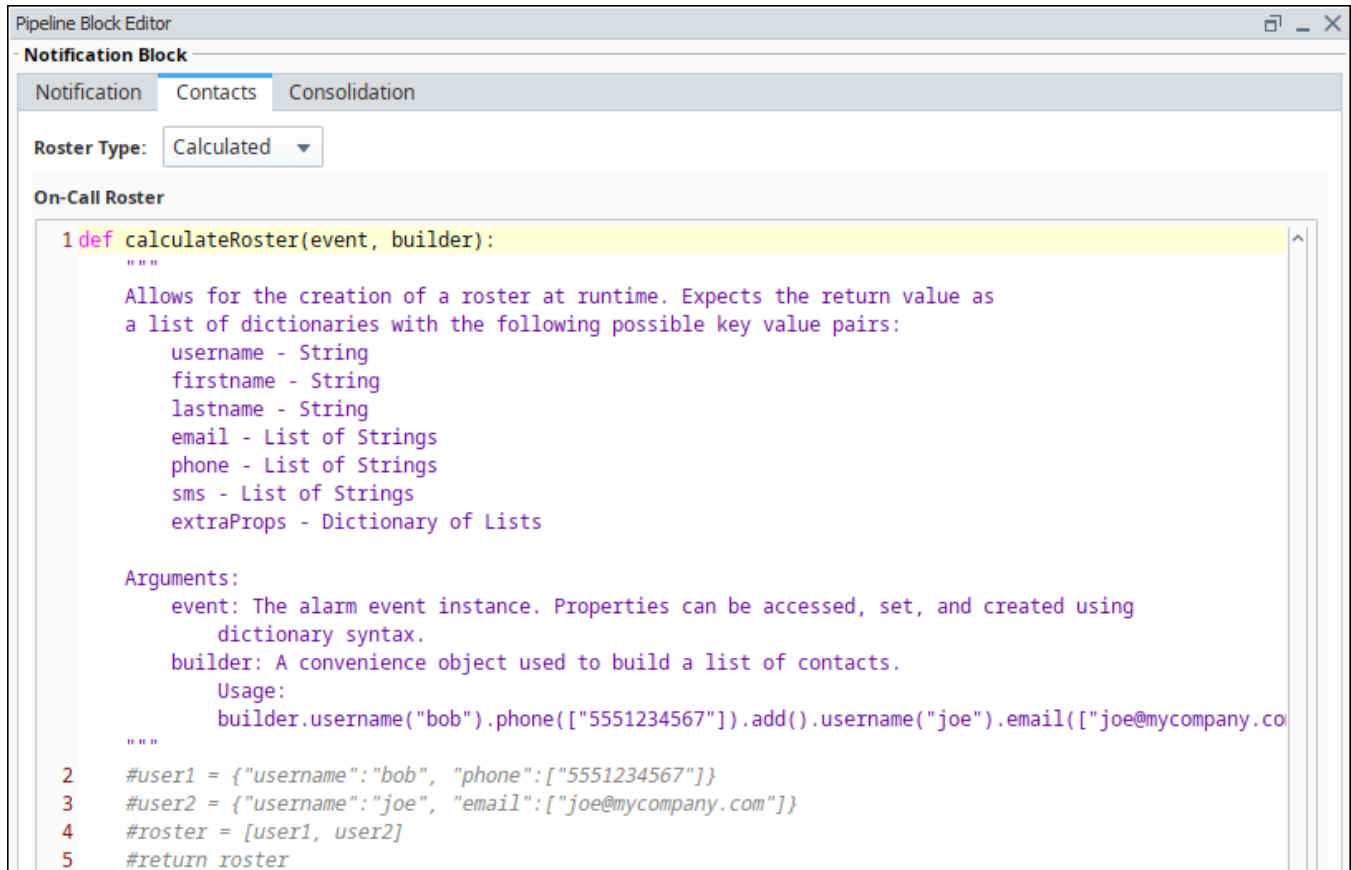
However, if the quotes were removed from the roster names, the expression would fail with a syntax error:

```
//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.



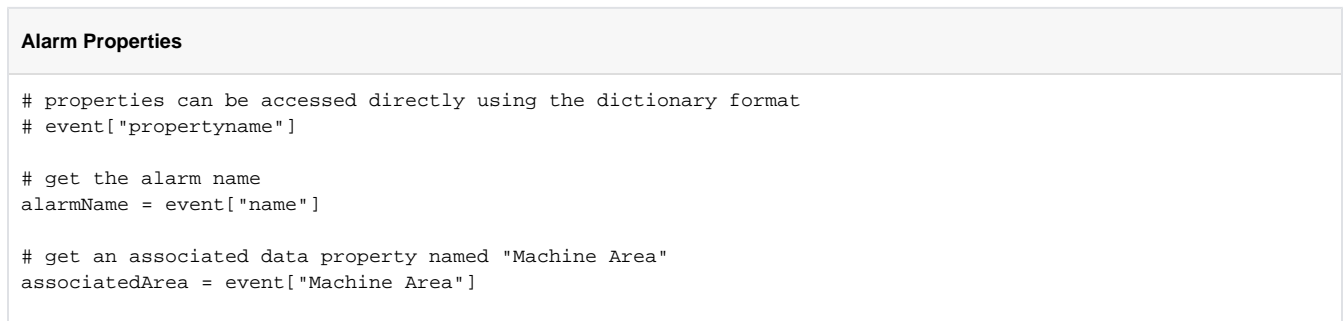
The screenshot shows the Pipeline Block Editor interface. The 'Notification Block' is selected, and the 'Contacts' tab is active. The 'Roster Type' is set to 'Calculated'. Below this, the 'On-Call Roster' section contains a script for a function named 'calculateRoster'. The script is as follows:

```
1 def calculateRoster(event, builder):
  """
  Allows for the creation of a roster at runtime. Expects the return value as
  a list of dictionaries with the following possible key value pairs:
  username - String
  firstname - String
  lastname - String
  email - List of Strings
  phone - List of Strings
  sms - List of Strings
  extraProps - Dictionary of Lists

  Arguments:
  event: The alarm event instance. Properties can be accessed, set, and created using
  dictionary syntax.
  builder: A convenience object used to build a list of contacts.
  Usage:
  builder.username("bob").phone(["5551234567"]).add().username("joe").email(["joe@mycompany.co
  """
2  #user1 = {"username":"bob", "phone":["5551234567"]}
3  #user2 = {"username":"joe", "email":["joe@mycompany.com"]}
4  #roster = [user1, user2]
5  #return roster
```

Accessing Alarm Properties

You can access any of the alarm properties through the event object in a calculated roster. Properties are stored as a dictionary and can be accessed in the same way.



The 'Alarm Properties' section contains the following code examples:

```
# properties can be accessed directly using the dictionary format
# event["propertyname"]

# get the alarm name
alarmName = event["name"]

# get an associated data property named "Machine Area"
associatedArea = event["Machine Area"]
```

Calculated Rosters

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	Data Type
username	<p>A username for this particular user.</p> <p>Note: This can be anything that you want, the user does not have to exist in the Gateway. It will be used for acknowledgement and auditing purposes, so it is important you create meaningful usernames.</p>	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 address 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	<p>Extra properties that can be added to a user. This is useful when dealing with any third party functionality.</p> <p>The following feature is new in Ignition version 8.1.8 Click here to check out the other new features</p> <p>The extraProps key can be used to set arbitrary user properties as of 8.1.8. This is useful when trying to set a user property not mentioned above. For example, if you needed to set the language for a user on a calculated roster to Spanish so that the Voice Notification module uses a Spanish voice, the user would be configured as follows:</p> <pre>user1 = {"username":"bob", "phone":["5550100"], "extraProps":{"language":["es_ES"]}}</pre> <p>The extraProps key can be used to set the following user properties. Technically it can be used to set any user property. See the User object. The value for each key must be a list.</p> <ul style="list-style-type: none"> • badge • language • schedule <p>In addition, extraProps can be used to set user properties that are registered by modules. For example, the SMS module registers a user pin setting, which can be set with the user.pin key:</p> <pre>user1 = {"username":"bob", "phone":["5550100"], "extraProps":{"user.pin":["1234"]}}</pre>	Dictionary of Lists

User Data Created Manually

When putting together a [dictionary](#) manually, you 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 to determine who to contact.

Manual User List
<pre># 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</pre>

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

- `add()` - Adds all previous info as a single dictionary.
- `build()` - Returns 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 of code.
# 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, adding hers first.
# 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('John').lastname
('Smith').email(['john@comapny.com']).phone(['1234567890']).sms(['8905671234']).add()
```

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, I do it in two stages.
# The difference here is that I have to call builder again in the second line when adding John's information.
builder.username("mary").email(['mary@company.com']).add()
builder.username('john').firstname('John').lastname('Smith').email(['john@comapny.com']).phone
(['1234567890']).sms(['8905671234']).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, and SMS 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"]]).add()
```

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 the alarm name property
alarmName = event["name"]

# don't send notification for some alarms
if alarmName in ["Diagnostic Alarm", "Joe's Test Alarm"]:
    # return an empty list so no one will be emailed
    return []

else:
    # 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"]]).add()

# use the builder object's build() function to create the finished list of data
userList = builder.build()
return userList
```

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

Ignore User Schedules

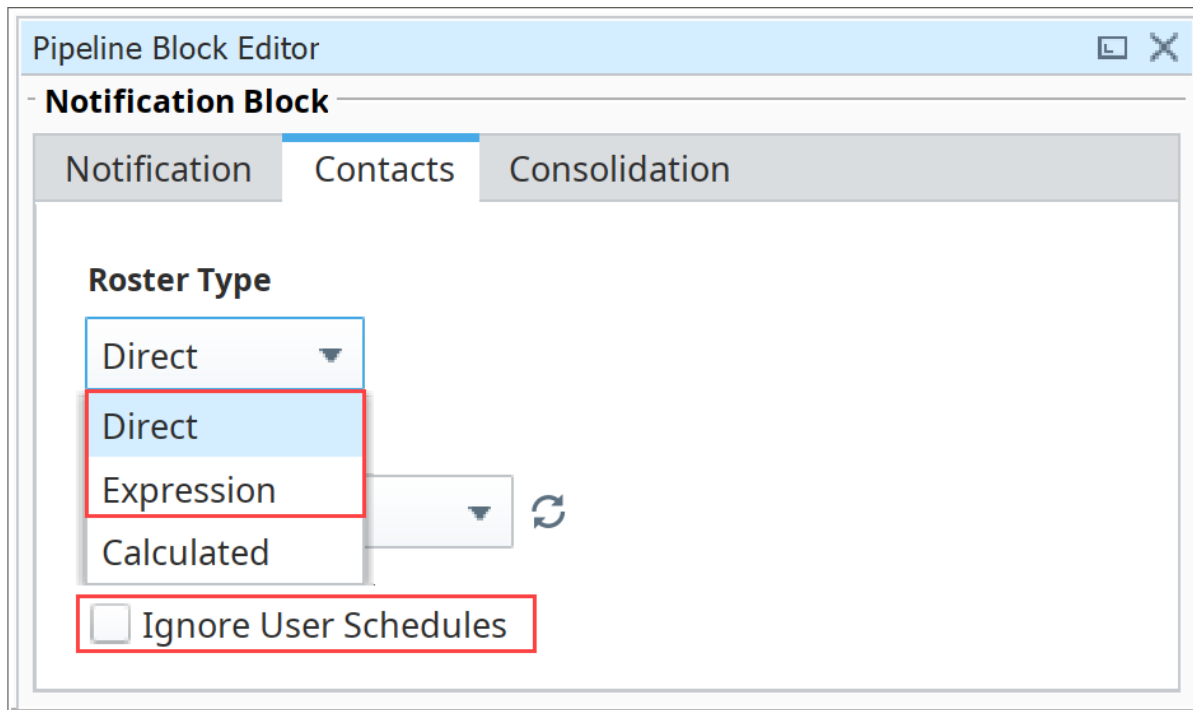
The following feature is new in Ignition version **8.1.13**
[Click here](#) to check out the other new features

This setting allows the notification block to ignore the schedules of any of the users in the roster, notifying users that would normally be "off shift." This setting is available on the Direct and Expression roster types.



This feature was changed in Ignition version 8.1.25:

Be mindful of your User Source settings when using this feature. Users belonging to User Sources where the [Schedule Restricted](#) property is checked means that even though they will receive alarm notifications when they are off shift, they will not be able to log in to address any alarms until their assigned schedule is active.

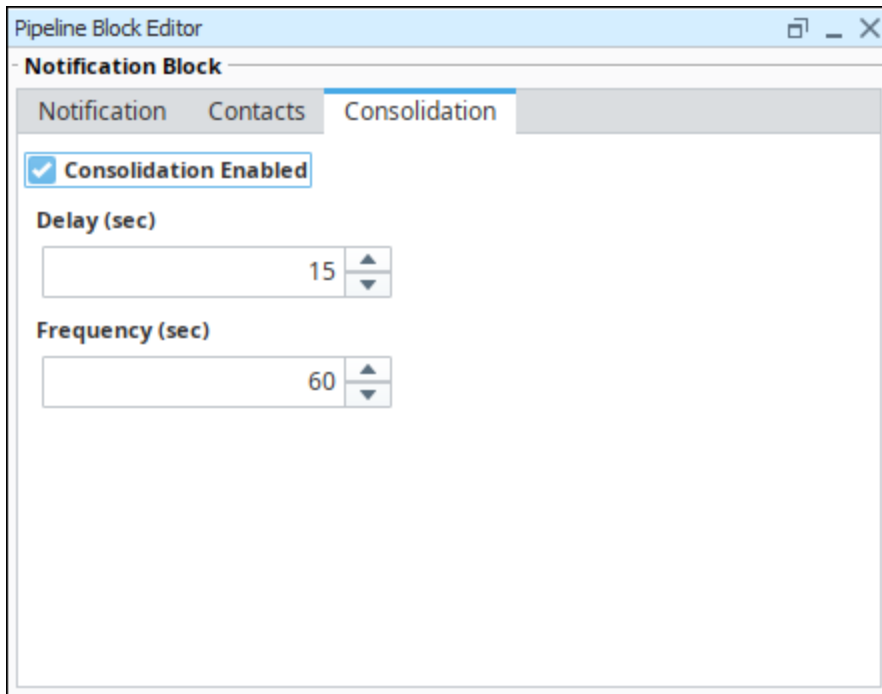


Consolidation

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:

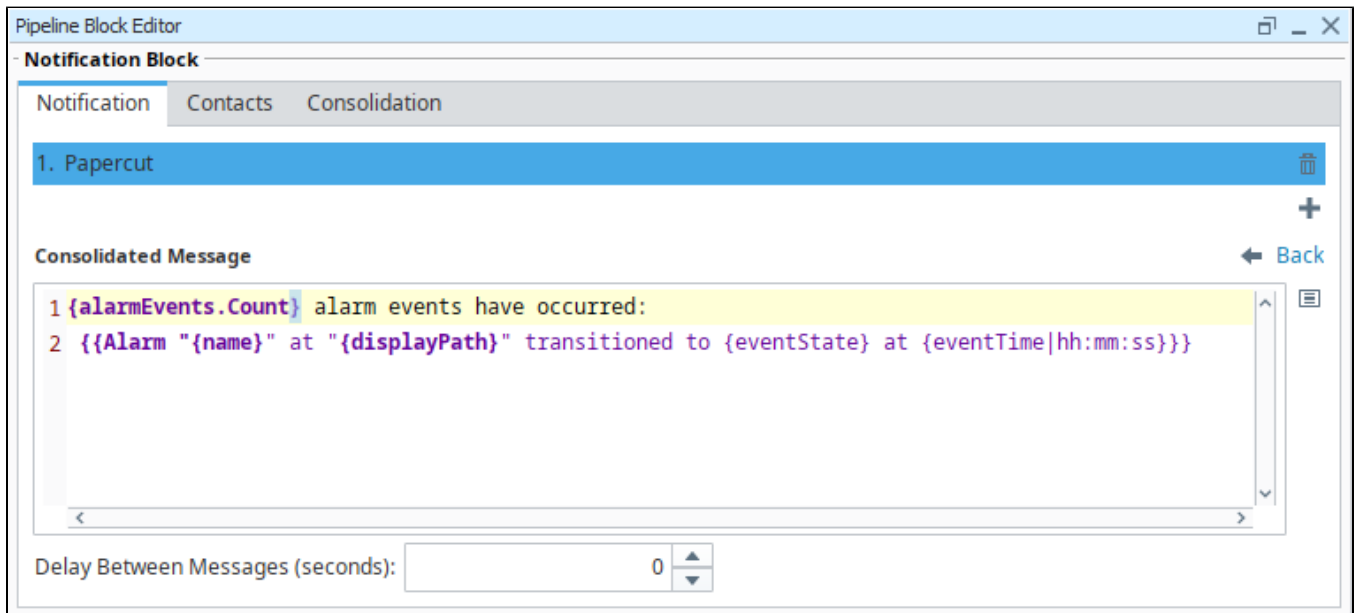
- **Delay** - How long to wait after the first eligible alarm arrives. This setting forces a pause *before* sending a notification. The delay is used to protect against situations where an event might generate many alarms together. In other words, if an alarm comes into the Notification Block, it will 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.



Consolidated Message Property

Some notification settings include a **Consolidated Message**, which dictates the formatting for the message when consolidation is enabled. This property features a text area that allows you to determine the content of the consolidated message.

Note: While the editor looks like an Expression Editor, it's not, so the normal expression functions and rules do not apply here. Although there is a Property Reference button to the right of the editor that can create references to alarm event properties. Since this isn't an expression, you don't need to wrap string literals in quotation marks.



Since consolidated messages will potentially contain multiple alarms, they utilize some unique syntax: the double curly bracket `{{}}` notation. The content within this notation will be repeated for each alarm included in the consolidated notification.

```
{alarmEvents.Count} alarm events have occurred:  
{{Every thing in here will be repeated for each alarm included in the consolidated notification}}
```

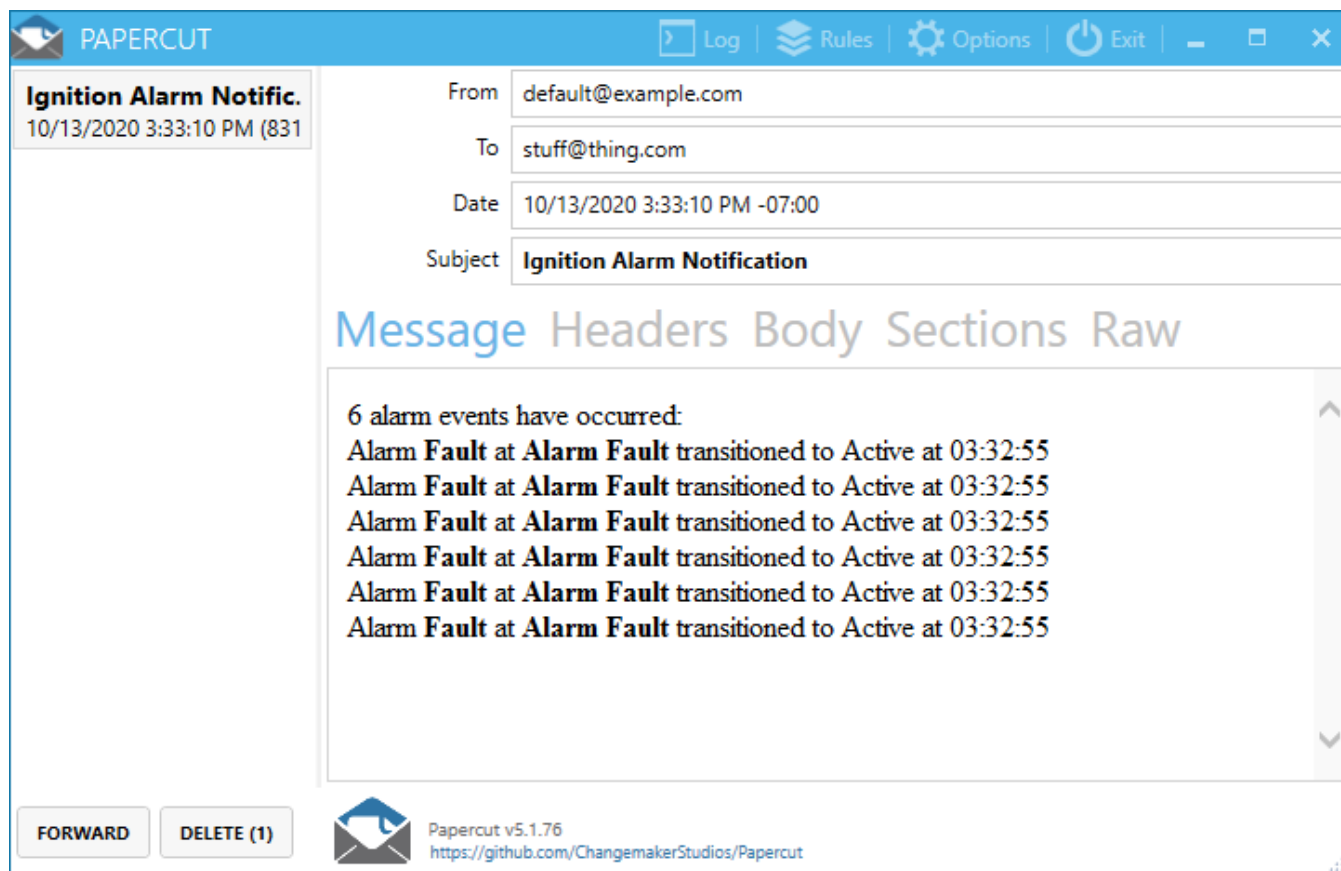
HTML Formatting in Email Notifications

Messages within consolidated messages can use HTML formatting by simply leading with an `<html>` tag at the start of the message. Doing this by itself does cause the resulting formatting to omit the built-in line breaks, so you will need to add line break `
` tags to signify a new line.

For example, we could add some HTML tags to the initial example on a consolidated message to write the alarm name and display path in bold `` using the following:

```
<html>  
{alarmEvents.Count} alarm events have occurred: <br>  
{{Alarm <b>{name}</b> at <b>{displayPath}</b> transitioned to {eventState} at {eventTime|hh:mm:ss} <br>}}
```

Which would result in the following body:



The screenshot shows the Papercut email client interface. The window title is "PAPER CUT" and it includes standard window controls (Log, Rules, Options, Exit). The email header shows:

- From: default@example.com
- To: stuff@thing.com
- Date: 10/13/2020 3:33:10 PM -07:00
- Subject: Ignition Alarm Notification

The main body of the email displays the following text:

6 alarm events have occurred:
Alarm Fault at Alarm Fault transitioned to Active at 03:32:55
Alarm Fault at Alarm Fault transitioned to Active at 03:32:55
Alarm Fault at Alarm Fault transitioned to Active at 03:32:55
Alarm Fault at Alarm Fault transitioned to Active at 03:32:55
Alarm Fault at Alarm Fault transitioned to Active at 03:32:55
Alarm Fault at Alarm Fault transitioned to Active at 03:32:55

At the bottom of the window, there are buttons for "FORWARD" and "DELETE (1)", and a footer with the Papercut logo, version "v5.1.76", and the GitHub repository link "https://github.com/ChangemakerStudios/Papercut".

Pipeline - Filter on Alarm Priority

Note: To learn more, go to [Configuring Alarms](#), [Alarm Notification](#) and [On-call Rosters](#).

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.

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. In this example, we already have two On-Call Rosters set up named "Critical" and "NonCritical."

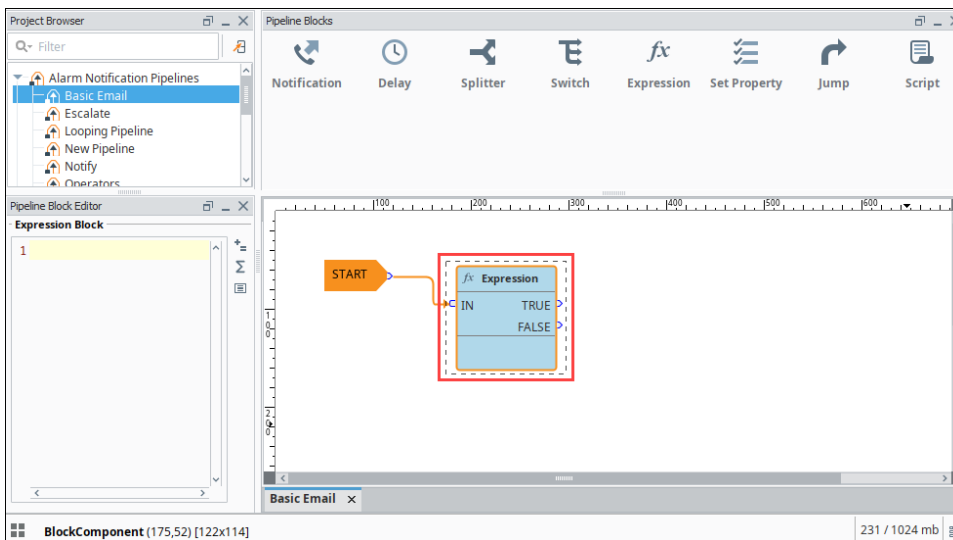


Pipeline - Filter on Alarm Priority

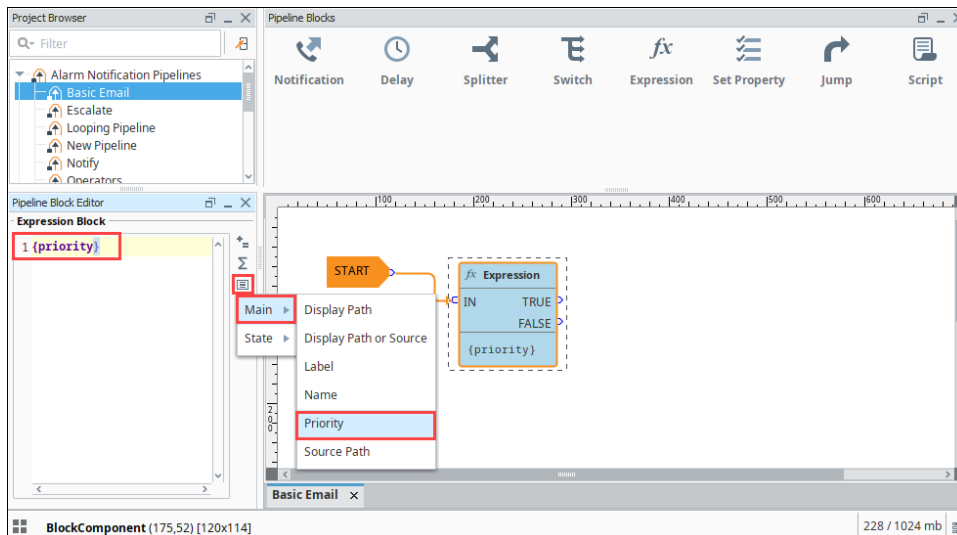
[Watch the Video](#)

To Filter on Alarm Priority, do the following:

1. Drag an **Expression block** into your pipeline.
2. 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.



3. In this expression, you want to check for the alarm's priority. Go to the Pipeline Property Editor, click the **Alarm Properties** icon.
4. **Select Main** and then 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.



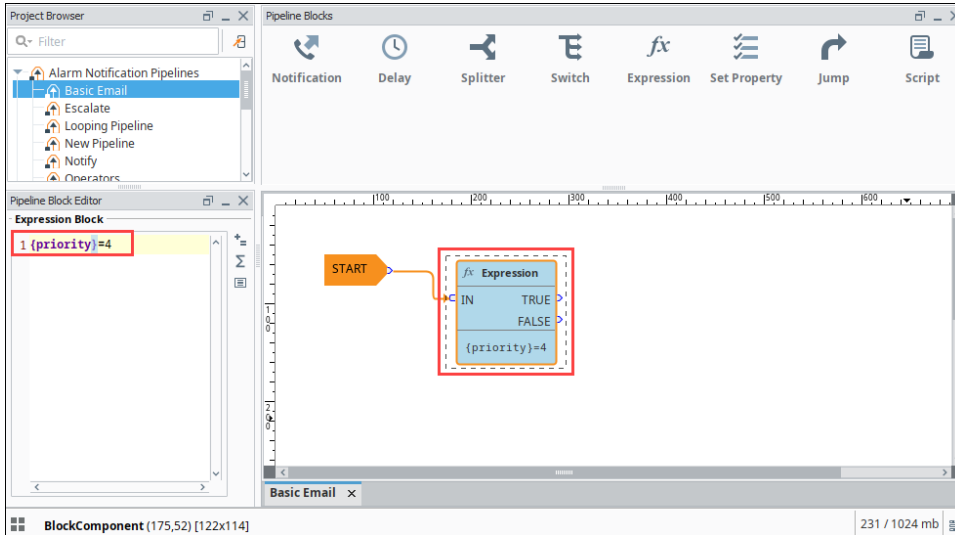
- Alarm priorities have predefined integer values for filtering and sorting. Because this value is an int, you can test against it using =, <, >, >=, or <=. In this example, we are going to filter for all alarms with a priority of 'Critical.' **Critical** has a value of **4**.

i Alarm Properties

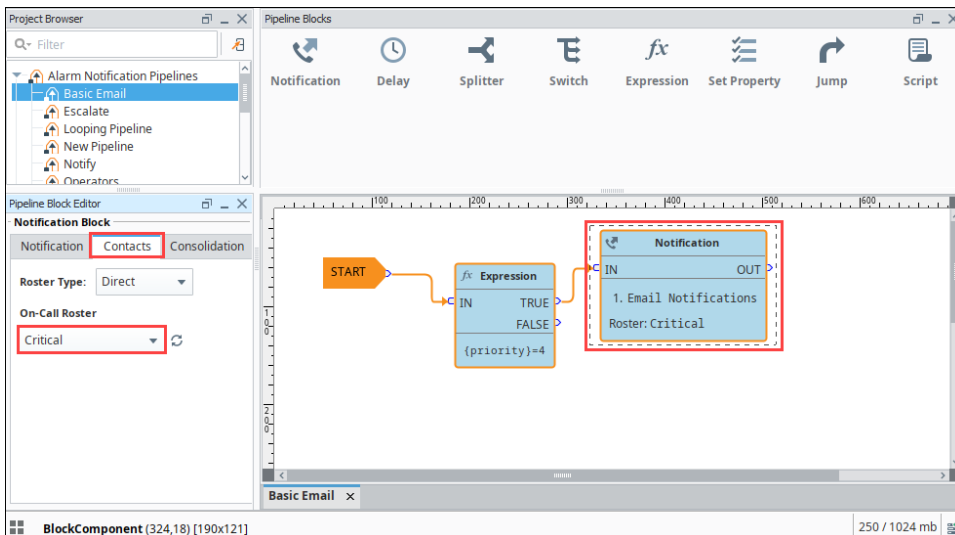
- 0 = Diagnostic
- 1 = Low
- 2 = Medium
- 3 = High
- 4 = Critical

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.

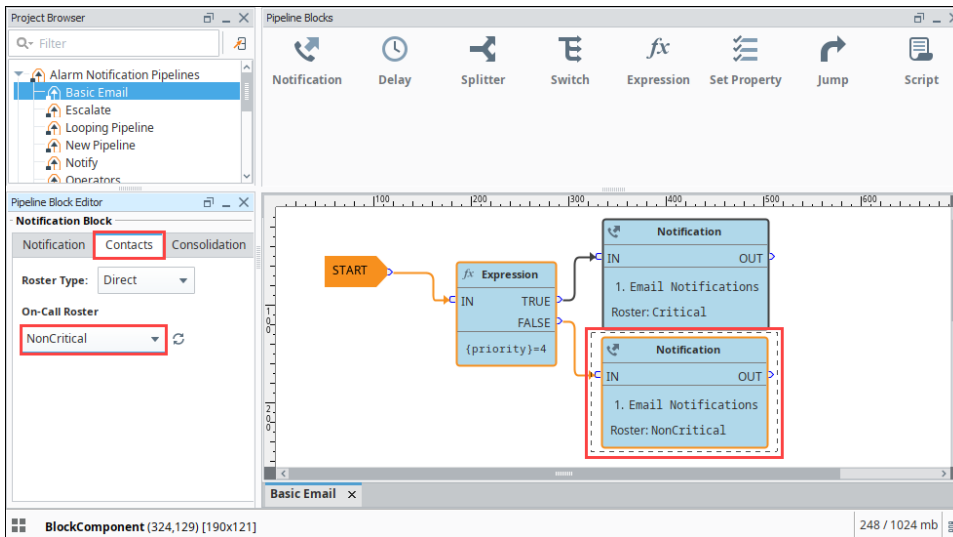


- Drag in a Notification block, and connect the **True** output pin into the IN of the Notification block.
- In the Pipeline Block Editor under the **Notification** tab, choose the notification type.
- On the **Contacts** tab, under the **On-call Roster** dropdown, select the Critical on-call roster list of people you want to email.



- 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 **Noncritical** on-call roster. The Critical list is a different list of people from the Non Critical list.



Now, the alarm notification pipeline notifies different lists of people based on the alarm's priority.

Related Topics ...

- [Pipeline - Filter on Alarm Associated Data](#)

Pipeline - Filter on Alarm Associated Data

Note: The examples in this section assume you have an alarm, alarm associated data, a notification profile, and an existing on-call roster configured.




Pipeline - Filter on Alarm Associated Data

[Watch the Video](#)

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.

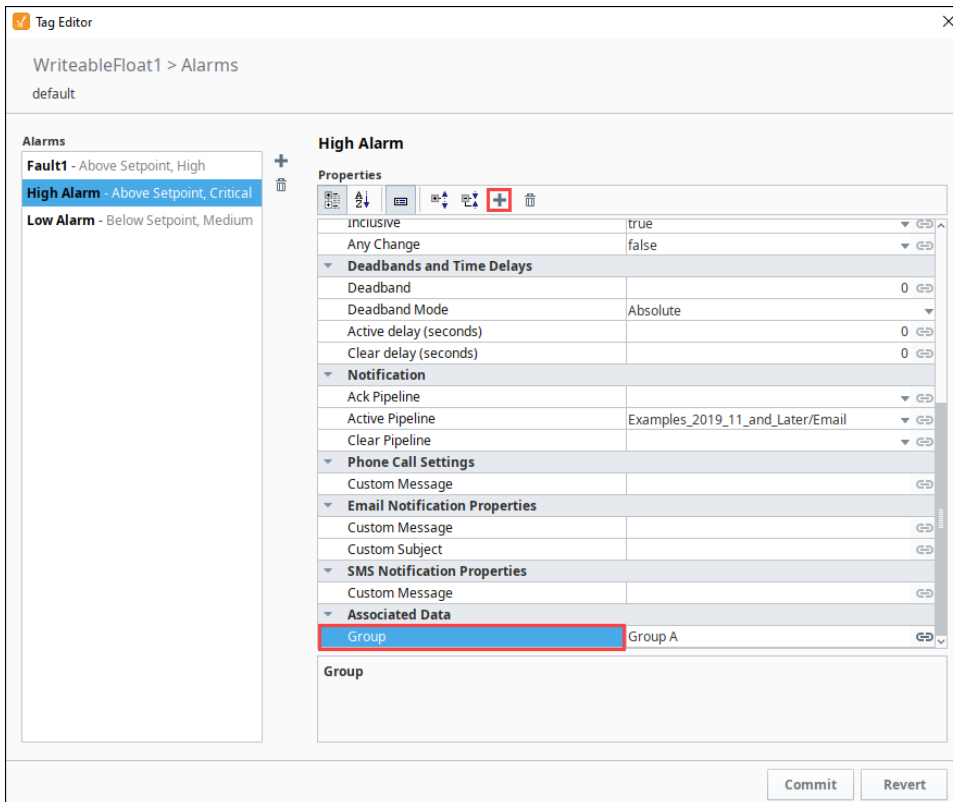
1. In the **Tag Browser**, double-click on **WriteableFloat1** to open the Tag Editor.
2. Scroll down to **Alarms**.
3. Click on the **Edit**  icon to view the alarms. This example has three alarms set up: Fault1, High Alarm, and Low Alarm.

The screenshot shows the 'Tag Editor' window for 'WriteableFloat1 > Alarms'. On the left, a list of alarms includes 'High Alarm - Above Setpoint, Critical'. The main area displays the configuration for the 'High Alarm'. The 'Properties' table is as follows:

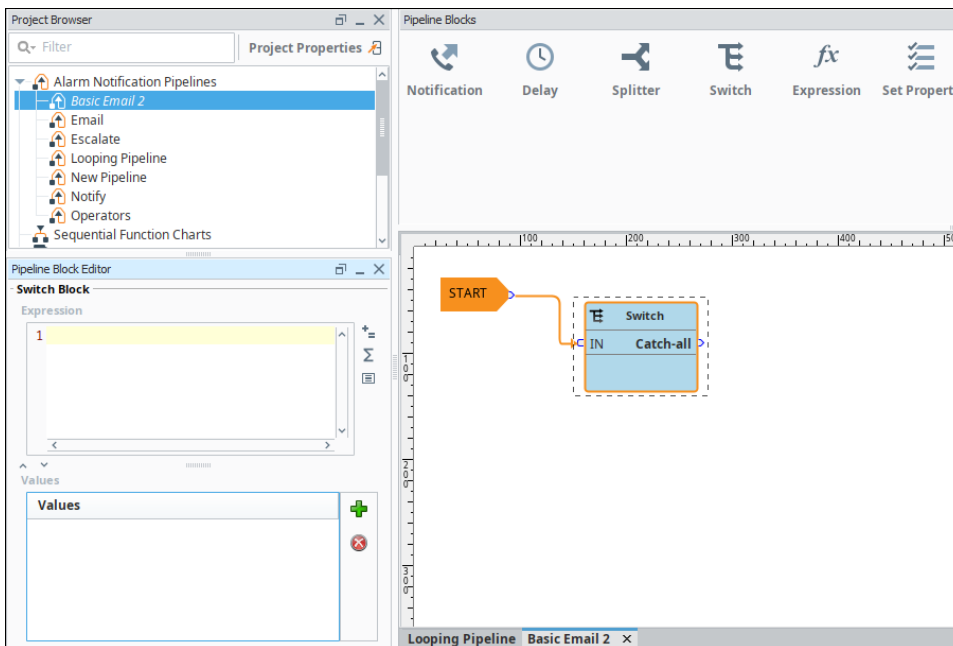
Property	Value
Ack Notes Required	false
Shelving Allowed	true
Alarm Mode Settings	
Mode	Above Setpoint
Setpoint	8
Inclusive	true
Any Change	false
Deadbands and Time Delays	
Deadband	0
Deadband Mode	Absolute
Active delay (seconds)	0
Clear delay (seconds)	0
Notification	
Ack Pipeline	
Active Pipeline	Examples_2019_11_and_Later/Email

Below the table, the 'Active Pipeline' section states: 'The pipeline that will be used to process active events generated by this alarm.'

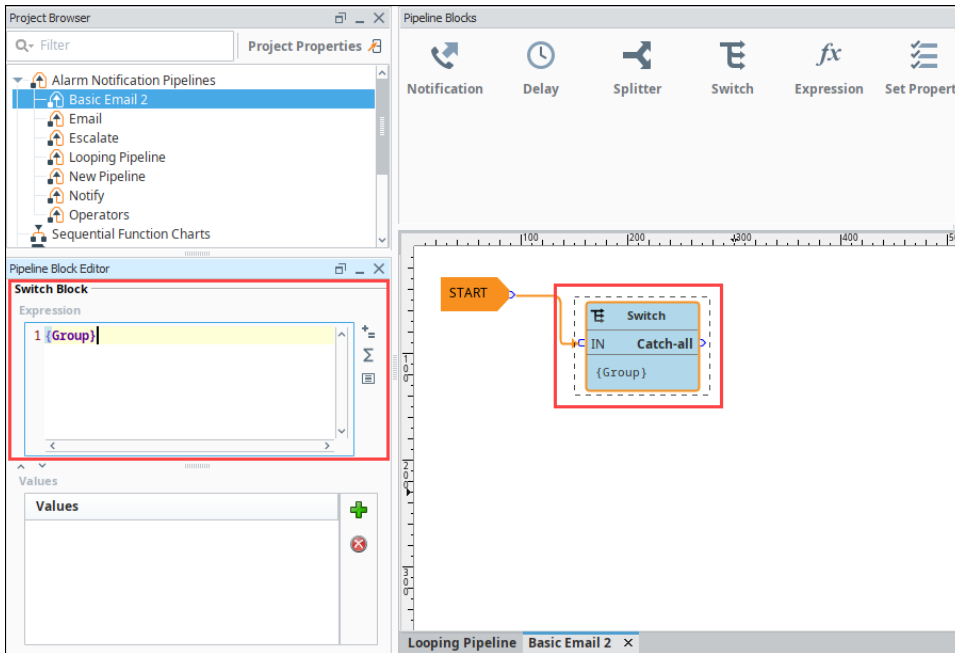
4. Expanding the High Alarm configuration shows the Associated Data called Group, and a value of either "Group A" or "Group B". We are going to use this associated data to filter the alarm notification pipeline.




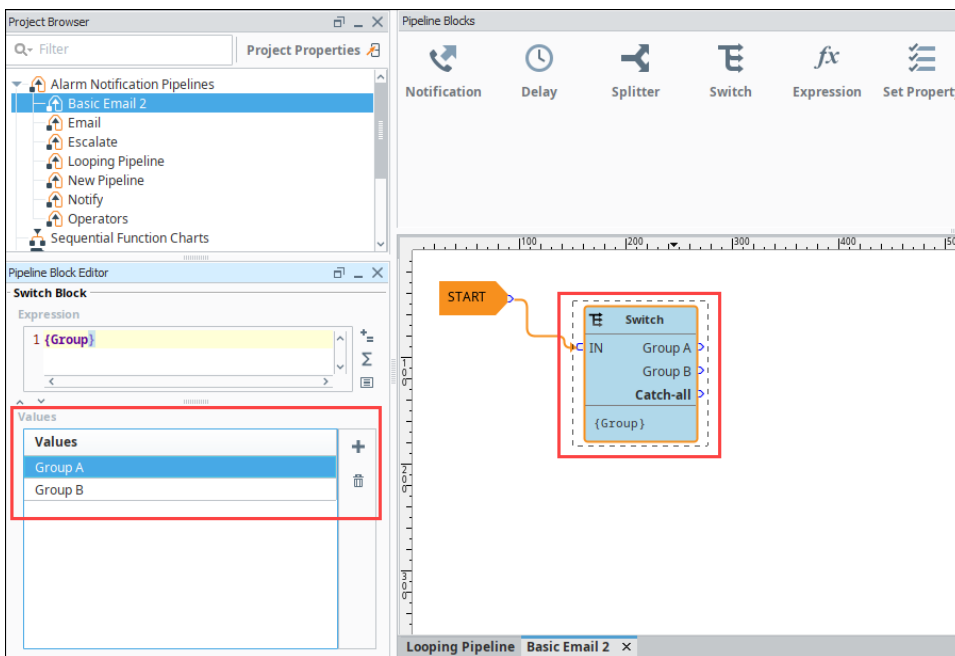
- For this example, drag in the **Switch block** to your pipeline, and connect the **Start block** to the **IN** of the Switch block. There are two blocks that can be used for filtering, either the Switch block or the Expression block. The Switch block allows you to switch on a value and do something different for Group A and Group B. The Expression block does the same thing, but only returns values of True or False.



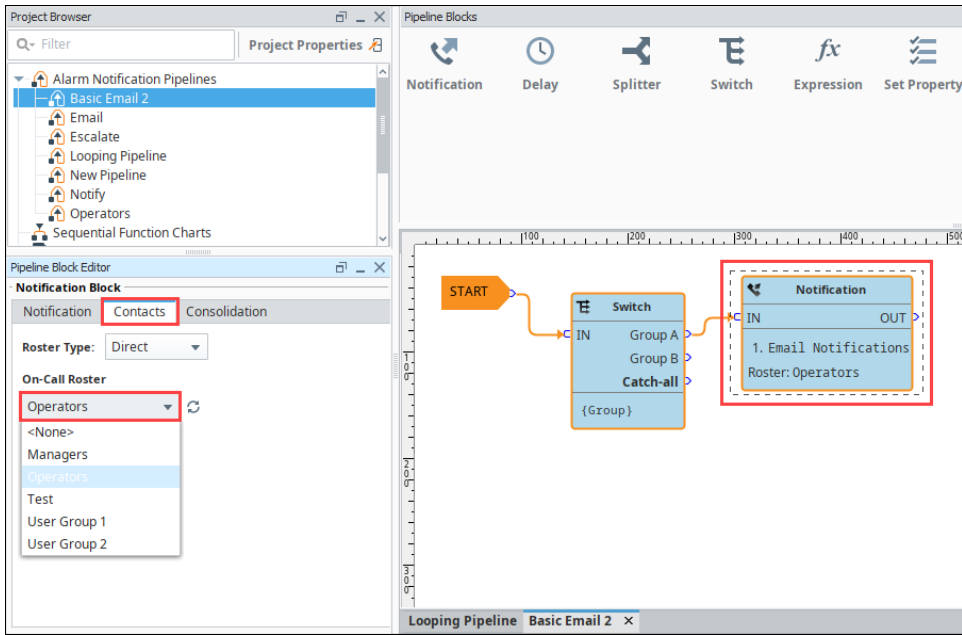
- In the **Pipeline Property Editor**, you have to manually enter the expression because you will not find any of the associated data properties in the dropdown list. Enter the name of the associated data property in curly braces as follows, **{Group}**. The **Group** associated data property acts as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



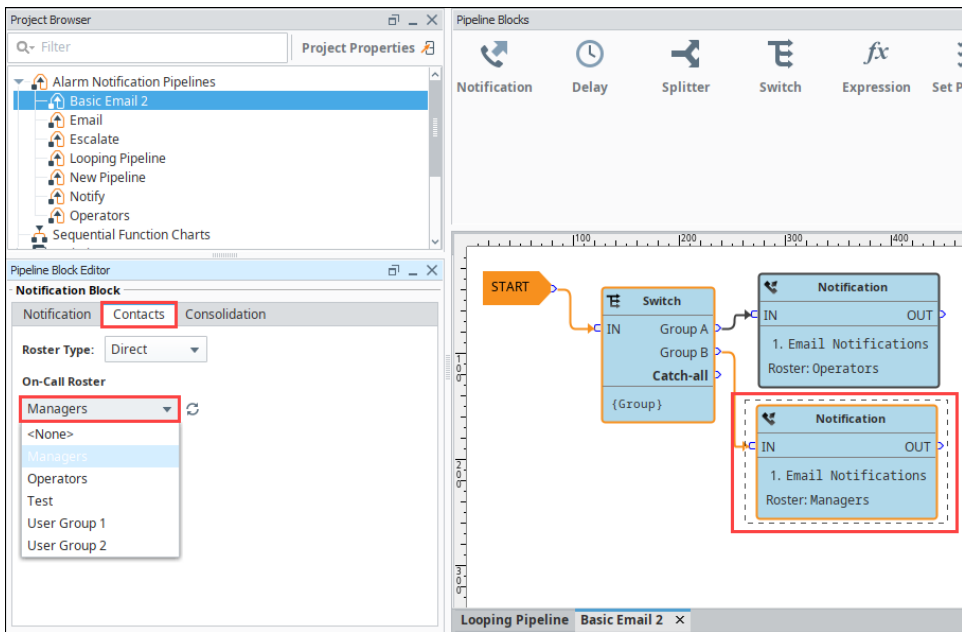
- Next, you need to add some values to the expression. You can add as many values that you have values for. Click on the **Add**  icon and add two values: **Group A** and **Group B**. Once you enter both values, you will have outputs in the Switch block for **Group A**, **Group B**, and by default, **Catch-all**. So now you can notify a different list of people based on each Group.



- 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).



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



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

Related Topics ...

- [Pipeline - Escalation](#)

Pipeline - Simple Loop

This section assumes that alarms are configured

To learn more, go to [Configuring Alarms](#), [Alarm Notification](#) and [On-Call Rosters](#).

On this page ...

- [How to Create a Looping Alarm Notification](#)

i This section assumes that alarms are configured

To learn more, go to [Configuring Alarms](#), [Alarm Notification](#) and [On-Call Rosters](#).

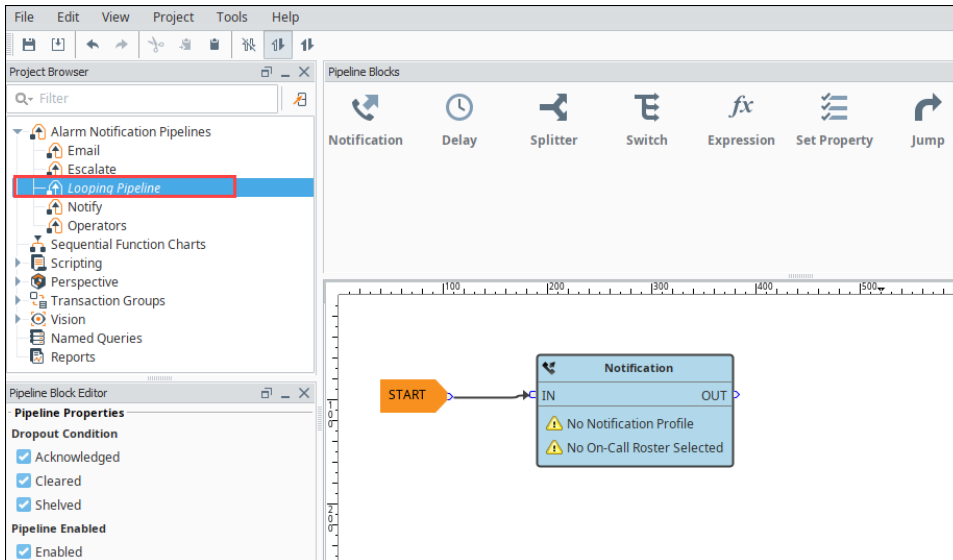
How to Create a Looping Alarm Notification

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.

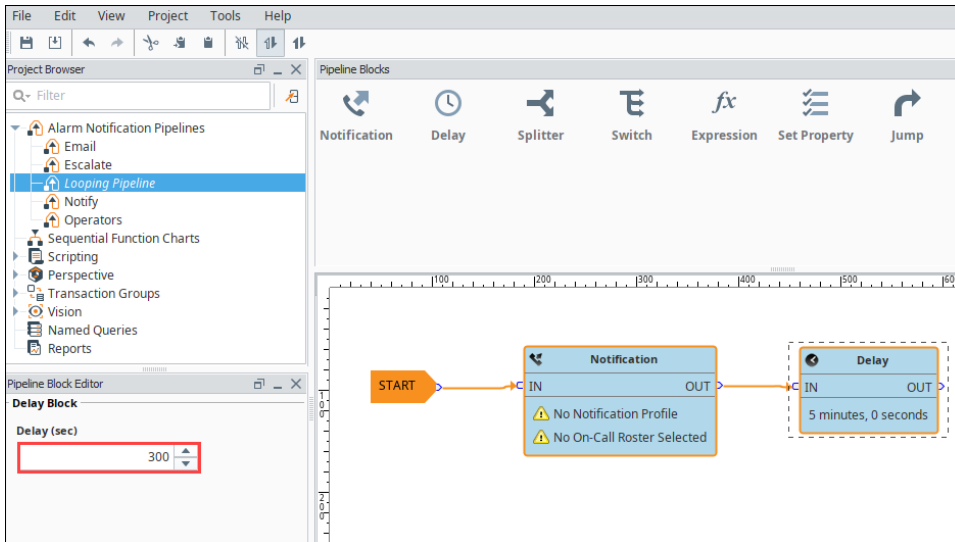
There are two [pipeline blocks](#) that we need to create a loop. The **Notification Block** to send a notification, and the **Delay Block** to create a pause between each notification.

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.

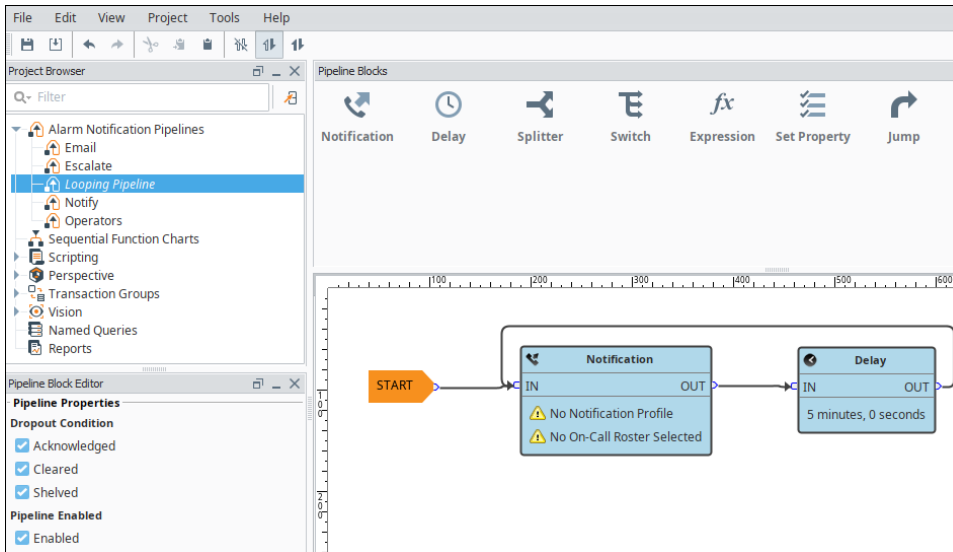


2. Drag a Delay block onto the pipeline and set the **Delay (sec)** property. We will chose 300 seconds (5 minutes).



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.



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.

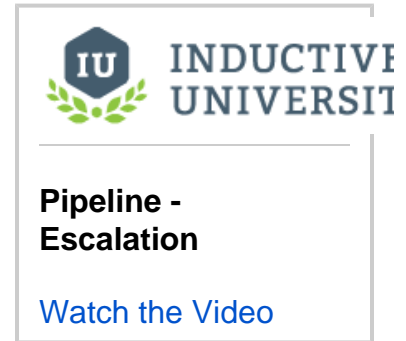
Related Topics ...

- [Pipeline - Escalation](#)

Pipeline - Escalation

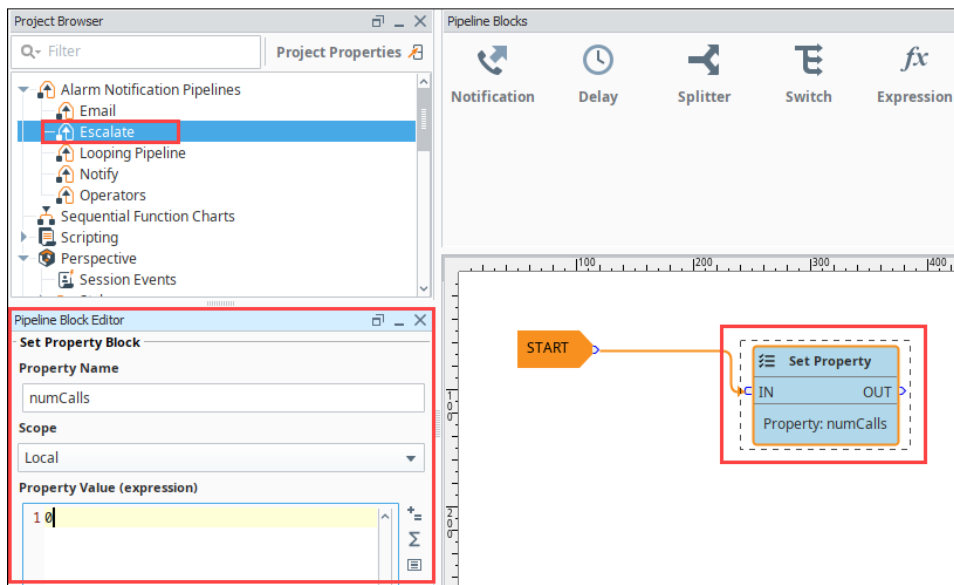
It is possible to set up escalation procedures in [alarm notification pipelines](#) in Ignition. The following example will create a pipeline that notifies operators three times. If a user does not acknowledge the alarm after the third time, the alarm escalates to the Production Supervisor's on-call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

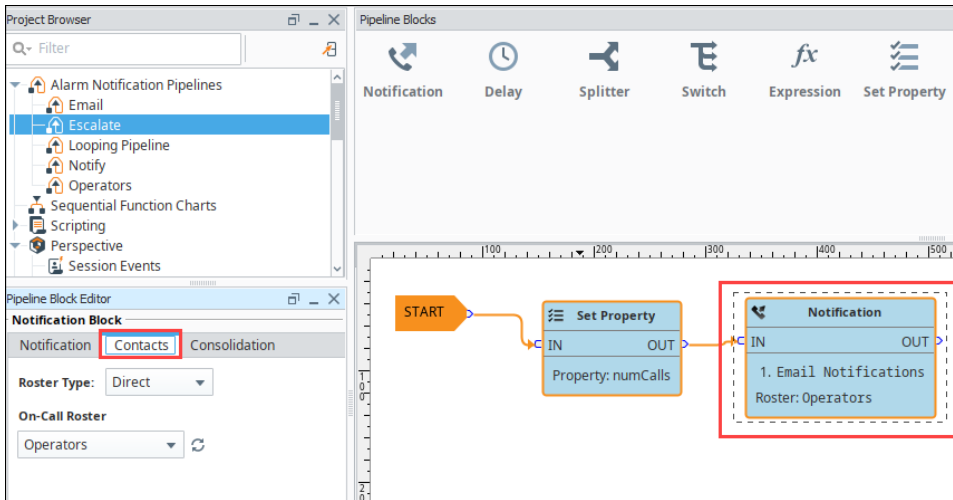


Set Up Escalation Procedures

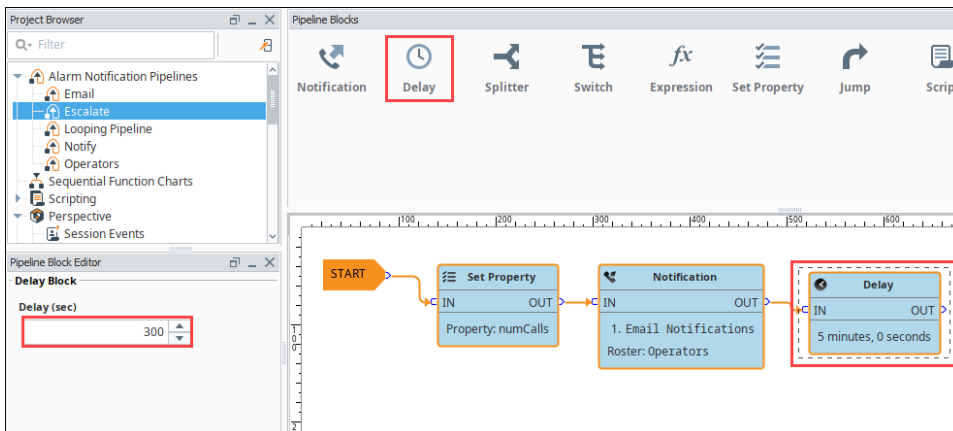
1. In the Project Browser under the Alarm Notification Pipelines, create a New Alarm Pipeline called **Escalate**.
2. In order to keep track of how many times the operators get notified, create a counter using the [Set Property block](#) inside your pipeline. Drag the Set Property block to the Pipeline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



4. Drag the Notification block to the Pipeline Workspace and connect the OUT from the Set Property block to the IN of the Notification block.
5. In the Pipeline Block Editor, set the Notification to use **Email Notifications** and set the Contacts to use the **Operators** on-call roster.



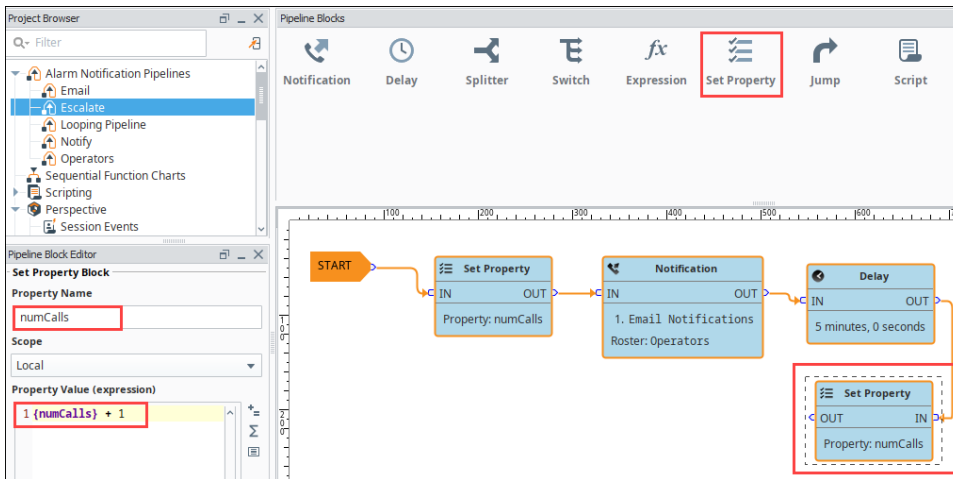
6. A Delay block will be added so the operators have some time to respond to the alarm before another notification is immediately sent out. Drag in the Delay block and connect the **OUT** from the Notification block to the **IN** of the Delay block.
7. In the Pipeline Block Editor, set the Delay property for **300 seconds**. It will give the operators 5 minutes to respond to the alarm before going to the next block.



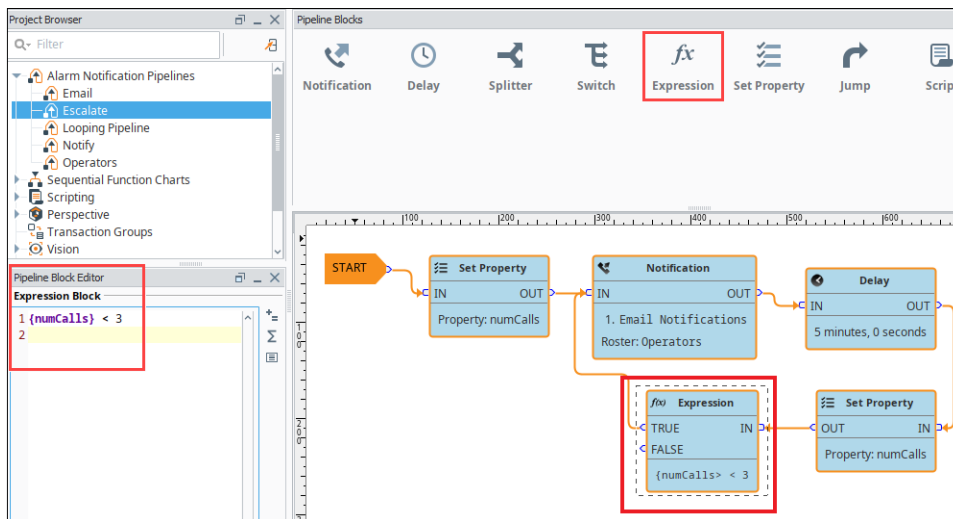
8. If the alarm is still active and not acknowledged, the counter will increment by 1. In order to increment the counter, drag in another Set Property block and connect the **OUT** of the Notification block to the **IN** of the Set Property block. You can right click on the Set Property block to **reverse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

Note: 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.

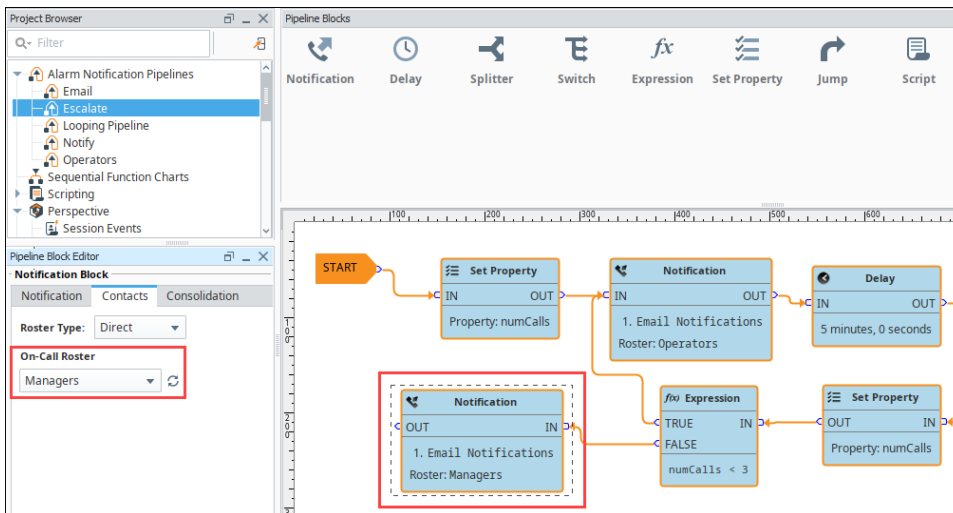
9. In the Pipeline Block Editor, set the Property Name to **numCalls**. In the Property Value Expression enter **"{numCalls} + 1"**. In this expression, use the curly braces to refer to that variable so the counter gets incremented by 1.



10. 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.
11. 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.



12. If the **numCalls** is **> 3**, the alarm event will follow the False path and it will escalate to the Managers.
13. Drag in another Notification block and reverse the block orientation. In the Pipeline Block Editor, set the Notification to use **Email Notifications**, and set the Contacts to use the **Managers** on-call roster.
14. 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 third 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.

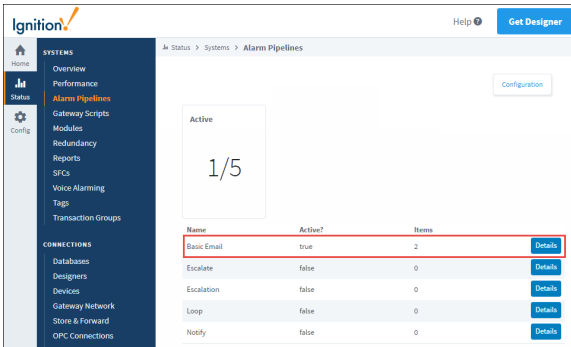
Note: You can also use the [Jump Block](#) as your escalation instead of a second Notification block. This will allow you to jump into another complete pipeline when nobody answers.

Alarm Pipelines Status

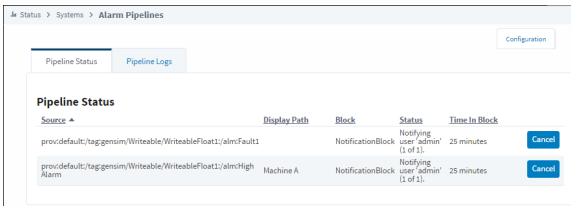
Monitoring the Pipeline Status

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 Gateway Webpage.

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.



2. 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.



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.).

Adding Pipelines to Tags

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.

Different Pipelines for Different 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.

Custom Message Scripts

When adding a pipeline to an alarm, you can also set up custom messages that will be used in each type of notification. These custom messages will only be used for this particular alarm, and will overwrite the settings configured for the message in the pipeline.

Email

Email Notification is unique in that not only can you specify a custom message, but also a custom subject for the email. These will only be used for the alarm they are configured on.

Custom Message Script

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

SMS

The SMS Notification has a property that allows you to create a custom text that will get sent out when this alarm is triggered.

Voice

The Voice Notification system allows you to create a custom message on alarms as well. The custom message will only replace the Active Message portion of the voice scripts, and will play the defined script for all other prompts.

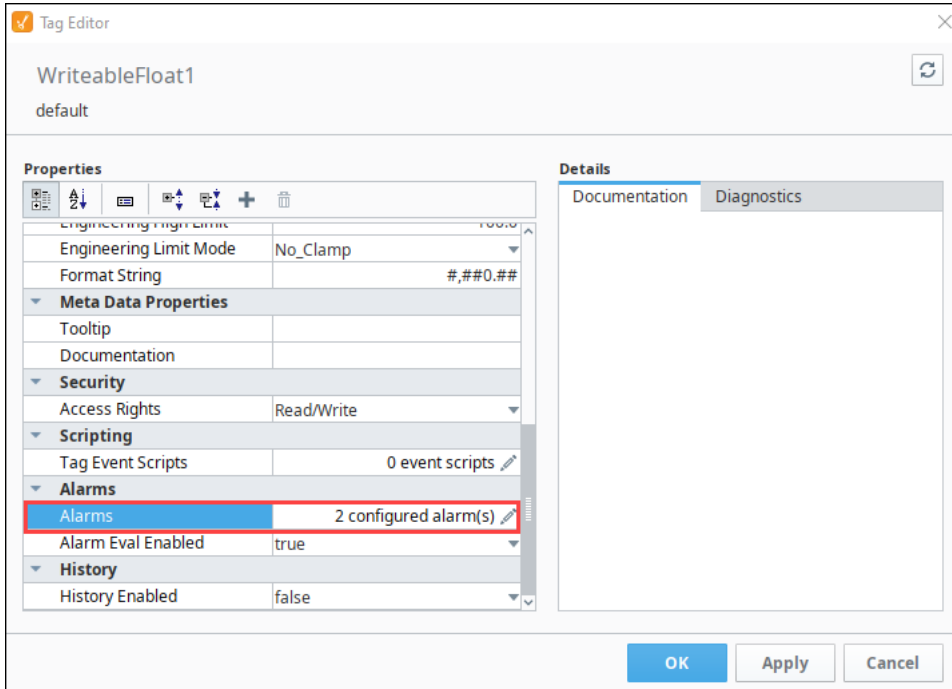
Associate a Pipeline with the Alarm


1. Double-click on a Tag that already has an alarm configured (i.e., WriteableFloat1). The **Tag Editor** is displayed and you can access its properties.

On this page ...

- [Different Pipelines for Different Events](#)
 - [Active Pipeline](#)
 - [Clear Pipeline](#)
 - [Ack Pipeline](#)
- [Custom Message Scripts](#)
 - [Email](#)
 - [SMS](#)
 - [Voice](#)
- [Associate a Pipeline with the Alarm](#)

2. Scroll down to the Alarms section and select an alarm. (If you don't have an alarm configured, create one. To learn more about creating alarms, go to [Configuring Alarms.](#))



3. Click on the Edit  icon to edit the alarms.
4. Select the alarm to edit then scroll down to the **Notification** section.
5. Select the desired alarm pipeline from the dropdown for any of the alarm status states. Typically, operators want to know when an alarm goes active. This example uses the Basic Pipeline when the alarm status becomes Active.
6. Click **Commit**.

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