

Ignition 8.3 User Manual — Alarm Notification Module

Generated 2026-10-12 from
<https://www.docs.inductiveautomation.com/docs/8.3/>

Alarm Notification

The Alarm Notification module allows you to send messages to groups of users when alarms become active, acknowledged, or cleared. This functionality requires the Alarm Notification module to be installed, along with alarm pipelines and at least one configured notification profile. Additional modules like the [SMS notification](#) and [Voice notification](#) modules extend this functionality to include additional communication methods.

Notification Contact Info

To send alarm notifications, users need [contact information](#), such as email addresses or phone numbers. This information is managed through the **Platform > Security > User Sources** section in the Gateway. Users typically have contact information added when their accounts are created, but it can also be updated as needed.

On-Call Rosters

An [On-Call Roster](#) is a collection of users who should receive notifications when an alarm occurs. Rosters evaluate user schedules to ensure only those with active schedules receive notifications, preventing unnecessary alerts. Configure rosters in the **Services > Alarming > Rosters** section of the Gateway.

Notification Profiles

[Notification Profiles](#) define how alarm notifications are sent, such as via email, SMS, or voice calls. Profiles connect Ignition to external systems for message delivery, supporting one-way and two-way communication. Configure profiles in the **Services > Alarming > Notification** section of the Gateway.

Alarm Pipelines

[Alarm Pipelines](#) define the logic for handling alarm notifications, including routing, escalation, and filtering. Pipelines are created and managed in the Ignition Designer. The **Services > Alarming > Pipelines** section of the Gateway provides an overview of active pipelines, including details like the pipeline name, path, active status, and current items. Clicking the ellipses next to a pipeline entry reveals additional details, including the source, display path, block, time in block, status, and a log of recent activity.

Assigning Pipelines to Tags

After creating pipelines, you must assign them to individual alarms. This is done in the Ignition Designer. To **assign a pipeline to a tag**:

1. Open the **Tag Browser** and double-click the tag you want to edit.
2. In the **Tag Editor**, select the **Alarming** section.
3. Choose an alarm, then select the desired notification pipeline.
4. Save your changes.

Once a pipeline is assigned, the alarm will use it whenever it becomes active, acknowledged, or cleared, depending on the pipeline configuration.

Testing Alarm Notifications

Ignition 8.3 includes a **Test** button for verifying alarm notifications directly from the Gateway. To test your alarm notification configuration:

1. Go to the **Services > Alarming > Notification** section of the Gateway.
2. Click the **Test** button next to the **Create Profile** button.
3. In the sidebar that appears, configure the following:
 - **Pipeline**: Select the pipeline you want to test.
 - **Display Path**: (Optional) Enter a display path to help identify the test event.
 - **Priority**: Choose a priority level for the test (Diagnostic, Low, Medium, High, Critical).
4. Click **Submit** to initiate the test.
5. Use the progress table to monitor the test, which includes columns for:
 - **Source**: The originating source of the test.
 - **Display Path**: The path entered during the test setup.
 - **Block**: The current block in the pipeline.
 - **Time in Block**: The time the test has spent in the current block.
 - **Status**: The current status of the test.
 - **Config**: Provides a cancel option for active tests.

Testing through the Gateway ensures that your pipelines, notification profiles, and rosters are correctly configured before going live.

Notification Contact Info



INDUCTIVE
UNIVERSITY

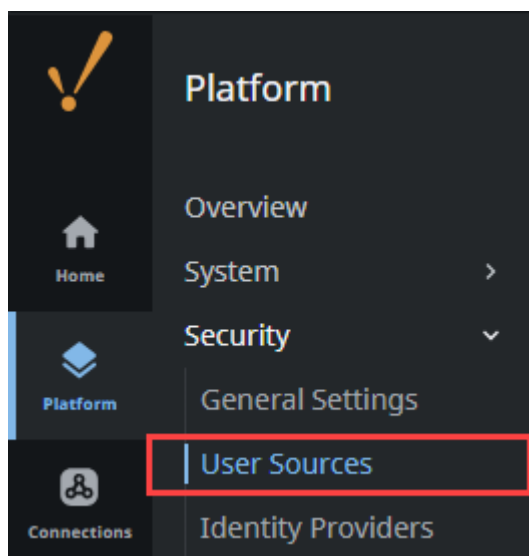
Notification Users

[Watch the video](#)

Ignition provides the capability to send out alarm notifications to a list of people in Email, SMS, and Phone (voice dial-out). There are several steps to get alarm notifications sent out to a list people. The first thing to do is create an [Alarm Notification Profile](#) for Email, SMS, or Voice based on the 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 **Platform** section of the Gateway Webpage.
2. Select **Security > User Sources**.



3. Expand the three dots menu for the appropriate user source and select **Manage Users** from the appropriate user source.

User Sources

Filter user sources... View 20 Verify User Source Create User Source +

NAME	DESCRIPTION	TYPE	
default	This is the default and always present internal authentication profile.	internal	⋮
opcua-module	OPC UA clients will authenticate against this profile by default.	internal	

- Edit
- Rename
- Duplicate
- Manage Users
- Disable
- Delete

4. Click **Create User +** to add a user.

default Manage Users

< Users Roles

Create User +

USERNAME	ROLES	SCHEDULE	
admin	Administrator	Always	⋮

5. Complete the fields that correspond to the user's information in the Create User form.

Create User

USER PROPERTIES *Required

Username *

Password * ⓘ

Must meet requirements. See tooltip.

Confirm Password *

Re-type password for verification

First Name

Cancel Create User

6. In the **Contact** section, click **Add +**.

7. Choose the delivery method from the Type dropdown list and then enter the appropriate Value, such as an email address if email was selected. Make sure to **Save** when done.

Create User

CONTACT

User Contact Information

Add +

TYPE	VALUE
Email Phone SMS	

Save

Cancel

PROPERTIES

Cancel

Create User

8. Click **Create 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

Setting	Description
Type	Dropdown list of contact information types, such as Email.
Value	Contact information (i.e., email address).

Extended Properties

Setting	Description
Security PIN	Security PIN for the user (Optional).

On-Call Rosters



INDUCTIVE
UNIVERSITY

On-Call Rosters

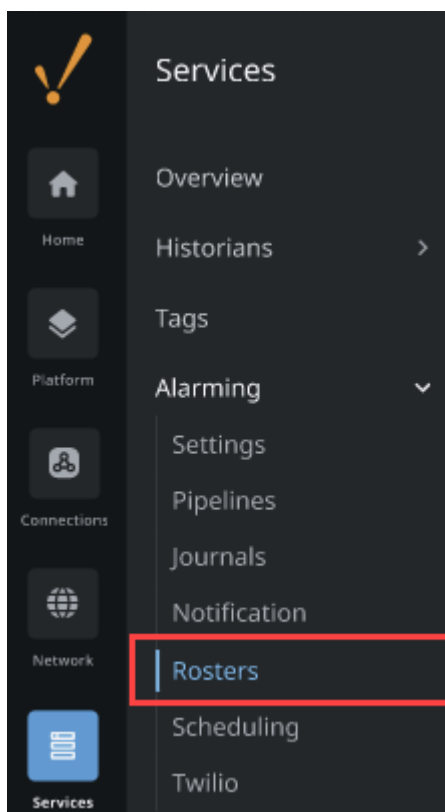
[Watch the video](#)

The On-Call Roster defines user groups to be notified when an alarm occurs. Each group includes a list of users in a specific order. Alarm pipeline notification blocks must choose an on-call roster that 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, except for any 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 **Services** section of the Gateway Webpage.
2. Click **Alarming > Rosters**.



3. Click on **Create Alarm Roster**.

Alarm Rosters

On-Call Rosters

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

View 20 ▾

Create Alarm Roster +

☐	NAME	DESCRIPTION	# OF USERS	
☐	Test		56	
☐	Shift A	desc.	2	

Showing 2 of 2 items

- On the **Create Alarm Roster** form, enter a **Name**, and **Description** if desired, and click **Create Alarm Roster**.

Create Alarm Roster



GENERAL

*Required

Name *

Description

Enabled

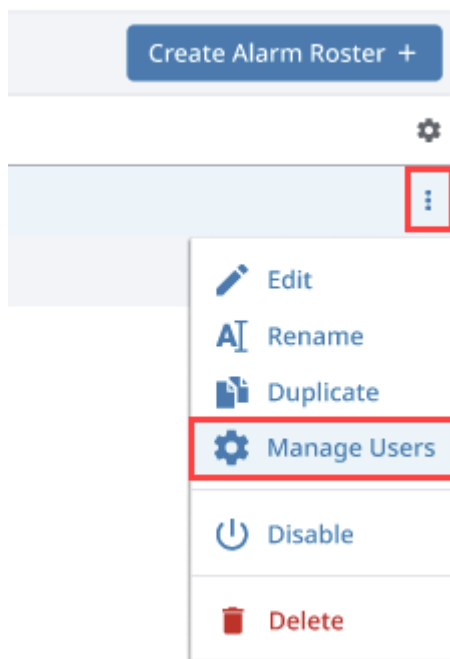


Set whether this roster is enabled or disabled within the Gateway.

Cancel

Create Alarm Roster

- Now on the Alarm Rosters page, expand the three dots menu for the new roster and select **Manage Users**.



6. The **Manage Users** panel 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 roster.

× {rosterName} Manage Users

ⓘ Click the right arrow on each user in the user source column (left) to move them to the on-call list (right). Re-order the users on the on-call list using the up and down arrows to control who gets called first. ×

Select User Source

Select... ▼

Users

You must select a user source to view available users.

On-Call Roster

Remove All

ID / USERNAME	CONTACT
No users assigned to roster.	

Cancel

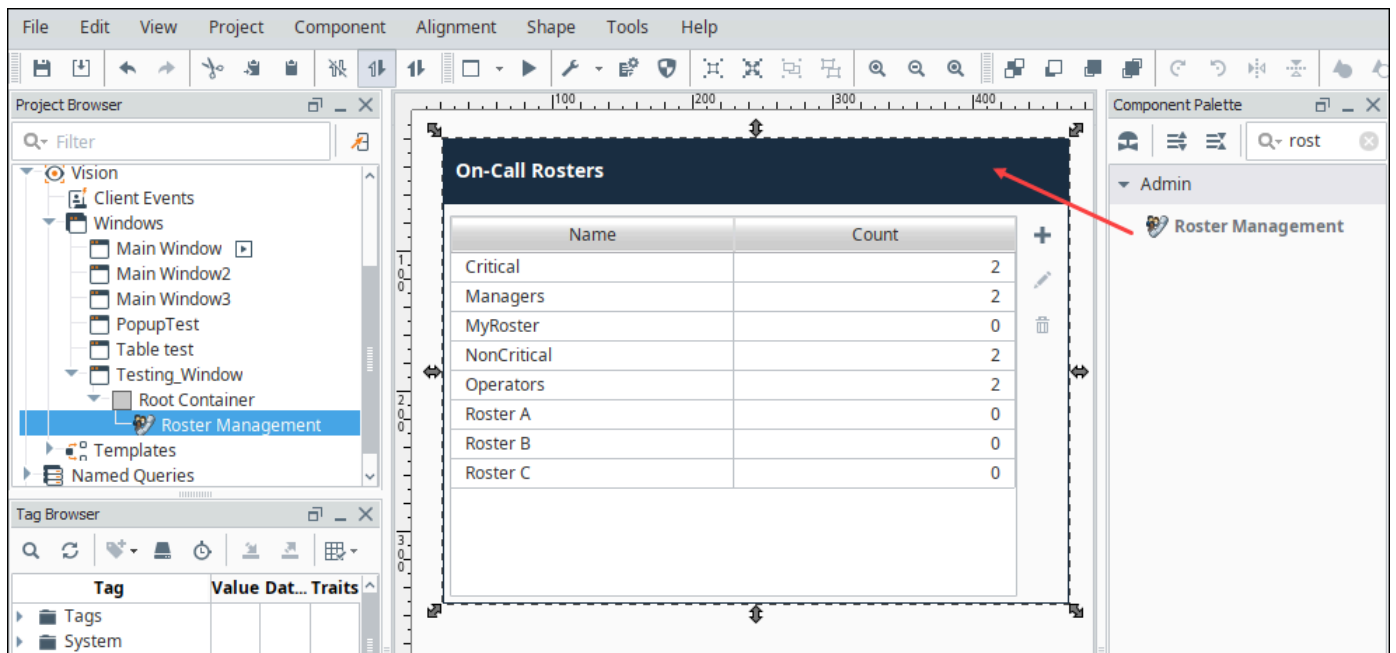
Save Changes

7. Click **Save Changes**.

Manage the Roster from the Client

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

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



3. If you have more than one user source, go to the Property Editor and enter the one you want to be used by the roster in the **User Source** property.
4. **Save** 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.

On-Call Rosters		
Name	Count	
Critical	2	  
Managers	2	
MyRoster	0	
NonCritical	2	
Operators	2	
Roster A	0	
Roster B	0	
Roster C	0	

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.

On-Call Rosters > Edit Roster

< back

Save

Add users from default

Available Users

admin
Bob
guest
Sal

On-Call Roster

Sal

admin

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 Notification Profile Types that allow email, SMS or text message, and Voice through a phone call in Ignition. All 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. To ensure acknowledgements are processed correctly, the contact information used to receive the alarm, such as an email address or phone number, must match the contact information assigned to a user in the On-Call Roster. Details on how to create and configure each of these profiles are found on the following sub-pages:

Email:

- **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.
- **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 Messaging:

- **SMS Notification Profile** - Sends a text message to users notifying them of an alarm, and a link to acknowledge the alarm.
- **Twilio SMS Notification** - Send SMS alarm notifications via a Twilio account.
- **Twilio WhatsApp Notification** - Sends WhatsApp notifications via a Twilio account.

Voice:

- **Voice Notification Profile** - Delivers alarm notifications to users via telephone. Acknowledgement is provided as a voice instruction as part of the call.
- **Twilio Voice Notification** - Send a Twilio Voice Notification.

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**.

The **Twilio Notification module** is required for any of the Twilio Notification profiles and can also be found on 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. Alternatively, a Twilio account may be used.

Email Notification Profile



INDUCTIVE
UNIVERSITY

Email Notification Profile

[Watch the video](#)

Set up an Email Notification Profile so alarms can trigger email alerts, and optionally allow recipients to acknowledge alarms directly from their inbox.

To send notifications, Ignition must be able to connect to an SMTP server. Once the profile is created, you can use it in your Alarm Pipelines to notify users when alarms become active or cleared.

Create an Email Notification Profile

This section walks you through creating a basic Email Notification Profile using the minimum required fields.

1. In the Gateway, go to **Services > Alarming > Notification**.
2. Click **Create Profile +**.
3. Select **Email Notification** and click **Next**.

Create Alarm Notification Profile

1. Profile Selection

2. Profile Setup

☒ **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.

☐ **VOIP Voice Notification**

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

Cancel

Next

**TIP**

If **Email Notification** isn't listed, make sure the **Alarm Notification** module is installed and running.

Go to **Platform > System > Modules** to verify.

4. Enter values for your device into the required fields. In this example, we used the following:

- **Name:** Email 1
- **Hostname:** mail@inductiveautomation.com
- **Enable SSL/TLS:** True
- **Username:** user@inductiveautomation.com

GENERAL

*Required

Name *

Email 1

Enter a name for the alarm notification profile.

Description

Enabled



Set whether this alarm notification profile is enabled or disabled within the Gateway.

EMAIL SETTINGS

*Required

Use SMTP Profile



If selected, this notification profile will use one of the Gateway defined email profiles. Otherwise, it will use the settings defined here.

SMTP Profile

Select



If Use Email 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.

Enable SSL/TLS

[Restore Default](#)



Connect using SSL/TLS. Only used when Use SMTP Profile is false.

Username

user@inductiveautomation.com

Only used when Use SMTP Profile is false.

5. Click **Create Alarm Notification Profile**.

6. The profile appears in the list. You can now assign it to an alarm pipeline.

Alarm Notification Profiles

View 20 ▾

Test ▶

Create Profile +

☐	NAME	DESCRIPTION	ENABLED	TYPE	STATUS	⚙
☐	Email 1		true	Email Notification	✓ RUNNING	⋮

Showing 1 of 1 items

Email Notification Profile Settings

The following tables describe all available configuration options for Email Notification Profiles.

Email Settings

Property Name	Description
Use SMTP Profile?	Use a Gateway-defined SMTP profile. If enabled, the Hostname, Port, SSL/TLS, Username, and Password fields are ignored.
SMTP Profile	Name of the Gateway SMTP profile to use.
Hostname	Hostname of the SMTP server.
Port	Port number for the SMTP service.
Enable SSL/TLS	Enables a secure connection to the SMTP server.
Username	SMTP server login username.
Password	SMTP server login password. Accepts None, Embedded or Referenced passwords.

Two-Way Settings

Property Name	Description
Two-way Enabled	Enables acknowledgment links in the email body.
Gateway	External address and port of the Gateway.
Send HTTPS Link	Use https:// in the email link.

POP3 Two-Way Settings

Property Name	Description
Pop3 Two-way Enabled	Enables acknowledgment from a POP3 mailbox.
Enable SSL/TLS	Enables a secure connection to the POP3 server.
Delete On Acknowledge	Deletes the email after it is used to acknowledge an alarm.
Hostname	Hostname of the POP3 server.
Port	Port for the POP3 server.
Username	POP3 login username.
Password	POP3 login password.
Custom Message	Optional message to include in the email body.
Polling Interval	Frequency (in ms) for checking the POP3 server.

Auditing

Property Name	Description
Audit Profile	Audit profile to log email sends and acknowledgments.

Advanced

Property Name	Description
SMTP Timeout	Timeout (in milliseconds) for SMTP communication.
Debug Mode Enabled	Logs email session output to wrapper.log.
STARTTLS Enabled	Enables STARTTLS if supported by the server. Ignored if SMTP Profile is used.

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.

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.



 Alarm(s) marked as "required" must include notes to acknowledge.

Ignition Alarm Acknowledgement

ALARM NAME	NOTES REQUIRED	ACTIVE TIME
myTagAlarm @ [default]myTag	 REQUIRED	12:39 PM

Notes:

Recalibrated the equipment.

Acknowledge

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.

 Alarm(s) marked as "required" must include notes to acknowledge.

Ignition Alarm Acknowledgement

 ALARM NAME	NOTES REQUIRED	ACTIVE TIME
☒ myTagAlarm @ [default]myTag	 REQUIRED	12:43 PM
☒ otherAlarm @ [default]myOtherTag	 REQUIRED	12:43 PM

Notes:

Fixed the valves.

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 **Diagnostics > 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](#).

Simple One-Way Email Notification Profile

The Simple One-Way Email Notification Profile is used to notify users when alarms occur, based on priority level. Unlike the standard Email Notification Profile, this one does not use alarm pipelines and does not include acknowledgment links in the emails.

Instead, it listens for active alarms within a specified priority range and sends notifications to a configured [On-Call Roster](#). When the criteria is met, the profile sends alerts to all users in the configured roster.

NOTE

The Simple One-Way Email Notification Profile requires the [Alarm Notification module](#) to be installed.

Create a Simple One-Way Email Notification Profile

Follow these steps to create a Simple One-Way Email Notification Profile.

1. In the Gateway, go to **Services > Alarming > Notification**.
2. Click **Create Profile +**.
3. Select **Simple One-Way Email Notification**, then click **Next**.

**TIP**

If the profile type doesn't appear, verify that the **Alarm Notification module** is installed and running.

Go to **Platform > System > Modules** to confirm.

4. Enter values for your device into the required fields. In this example, we used the following:

- **Name:** One-Way Email
- **SMTP Profile:** SMTP
- **On-Call Roster:** Managers
- **Minimum Priority:** High
- **Maximum Priority:** Critical

5. Click **Create Alarm Notification Profile**.

6. The profile will appear in the list and start running automatically. You can edit it at any time to update the On-Call Roster or alarm priority range.

Simple One-Way Email Notification Profile Settings

Main

Property Name	Description
Name	Name of the profile.
Description	Optional description of the profile.
Enabled	Whether the profile is active.

Email Settings

Property Name	Description
SMTP Profile	The SMTP Profile used to send notifications.
On-Call Roster	The roster of users who will receive the notification.

Alarm Criteria

Property Name	Description
Minimum Priority	The lowest priority that will trigger this notification. Options are Diagnostic, Low, Medium, High, or Critical.
Maximum Priority	The highest priority that will trigger this notification. Options are Diagnostic, Low, Medium, High, or Critical.

Consolidation

New in 8.3.7

Property Name	Description
Consolidation Enabled	When checked, enables consolidation to combine multiple alarms into single notifications.
Delay	The time, in seconds, to wait after an alarm is received to wait for another alarm before sending a message.
Frequency	The time, in seconds, to wait after a notification is sent before sending another notification.

Advanced

Property Name	Description
From Address	Optional email address to appear in the "From" field. If left blank, the SMTP username is used.

SMS Notification Profile



INDUCTIVE
UNIVERSITY

SMS Notification Profile

[Watch the video](#)

The SMS Notification Profile allows you to send alarm notifications over a cellular network using a supported AirLink modem. It supports both one-way and two-way SMS communication, including alarm acknowledgments via SMS reply if enabled.

This profile is commonly used in remote environments where network access is limited, but cellular coverage is available.

SMS Notification Module

Short Message Service (SMS) is the protocol used by most text messaging systems. The SMS Notification module enables SMS alarm delivery using a cellular modem configured with a valid SIM card. If two-way messaging is enabled, recipients can respond to alarm messages with an acknowledgment code.

NOTE

The SMS Notification module depends on the Alarm Notification module. Both must be installed and running.

Device Configuration

Inductive Automation officially supports the following Sierra Wireless AirLink devices:

- RV50 / RV50X
- RV55
- LX40

Other modems that support the AirLink IP protocol may also work, but may require additional configuration. Template settings for supported devices are available in the following Knowledge Base Articles:



Knowledge
Base

AirLink LS300 Setup Guide

AirLink LS300 Setup Guide

[Link to Knowledge Base Article](#)



Knowledge
Base

AirLink RV50 Setup Guide

How to set up the Sierra Wireless
AirLink RV50 device for use with the
Ignition SMS Notification module.

[Link to Knowledge Base Article](#)

Multiple Gateways with One Modem

A single AirLink modem can send messages for multiple Ignition systems in one-way (outgoing only) mode. However, only one Gateway should receive incoming SMS responses if two-way messaging is enabled.

Create an SMS Notification Profile

Follow these steps to create an SMS Notification Profile.

1. In the Gateway, go to **Services > Alarming > Notification**.
2. Click **Create Profile +**.
3. Select **SMS Notification**, then click **Next**.

**TIP**

If the profile type is missing, go to **Platform > System > Modules** to verify that the Alarm Notification module is installed and running.

4. Enter values for your device into the required fields. In this example, we used the following:

- **Name:** SMS
- **Airlink Host Address:** The IP address of the Airlink modem.
- **Send Port:** 17341
- **Receive Port:** 17342
- **Airlink Backup Host Address:** The IP address of the backup Airlink modem.
- **Backup Send Port:** 17341
- **Backup Receive Port:** 17342
- **Two-way Enabled:** True

- 5. Click **Create Alarm Notification Profile**.
- 6. The profile appears in the list and will begin running immediately. You can edit it at any time to modify the modem address, backup ports, or acknowledgment behavior.

Once configured, you can assign phone numbers to users using the **SMS** contact type.

SMS Notification Profile Settings

Main

Property Name	Description
Name	Name of the notification profile.
Description	Optional description for the profile.
Enabled	Whether the profile is currently active.

SMS Settings

Property Name	Description
Airlink Host Address	IP address of the primary AirLink modem.
Send Port	UDP port used to send messages to the modem. Default: 17341.
Receive Port	UDP port used to receive alarm acknowledgments from the modem. Default: 17342.
Airlink Backup Host Address	IP address of the backup modem used by a redundant Gateway if redundancy is configured.
Backup Send Port	Send port to use when in backup mode.
Backup Receive Port	Receive port to use when in backup mode.
Two-way Enabled	Allows recipients to acknowledge alarms by replying to the SMS.
Numeric Only Ack Code	Forces acknowledgment codes to be numeric only (instead of alphanumeric).

Auditing

Property Name	Description
Audit Profile	Stores SMS message events and acknowledgments in the selected audit profile. Alarm acknowledgments are also written to the Alarm Journal.

Twilio Notification Profiles

With the Twilio Notification module installed, you can configure notification profiles for your Twilio account to send SMS or voice notifications. SMS scripts for alarm notifications can be sent through Twilio's SMS messaging services or Twilio's WhatsApp integration. Additionally, profiles can be configured to allow alarm acknowledgement among other customizable options.

- **Twilio SMS Notification Profile:** Send SMS alarm notifications through scripts via a Twilio account.
- **Twilio Voice Notification Profile:** Send a Twilio voice notification with an automated phone call when an alarm is triggered.
- **Twilio WhatsApp Notification Profile:** Send SMS alarm notifications through WhatsApp messaging via a Twilio account.

Receiving Alarm Acknowledgement via Twilio

Your Ignition Gateway needs to be accessible by Twilio's servers to acknowledge an alarm when using 2-way alarming. This is because Twilio's servers send the acknowledgement as a POST request to your Gateway's web server port (443 or 80, depending on if you are using SSL). If you're running Ignition in the cloud, this is likely already the case.

! INFO

If Ignition is on-premise, you may need to explore port forwarding or putting 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, the SMS or Voice Notification module is typically preferred over the Twilio module for users running Ignition on-premise.

Define a Twilio Account

Before a profile can be created, a Twilio account with the necessary capabilities must be created (www.twilio.com) and then defined in Ignition.

1. Navigate to the **Services** section of the Gateway Webpage.
2. Choose **Alarming > Twilio**.
3. On the Twilio page, click **Create Account +**.
4. Provide a **Name** for the account as well as the **Twilio Account Sid**, and **Twilio Auth Token**. Both the Account SID and Auth Token can be retrieved from the Account Settings menu in Twilio.
5. If desired, 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 **Create Twilio Account**.

i NOTE

New in 8.3.7 Ignition will automatically retry connection attempts to Twilio accounts if the connection fails immediately after being configured.

Twilio Accounts Settings

The following tables describe the settings for Twilio accounts.

General

Setting	Description
Name	Name for the Twilio account.
Description	An optional description for the Twilio account.
Enabled	Whether or not the Twilio account is enabled or disabled within the Gateway.

Account

Setting	Description
Twilio Account Sid	The Twilio account Sid, retrieved from the Account Settings menu in Twilio. (Required.)
Twilio Auth Token	The Twilio auth token, retrieved from the Account Settings menu in Twilio. (Required.)
Local Address	The IP address of the local network interface to be used for Twilio communications. Leave blank to use the default interface.

Inbound

Setting	Description
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.
Public SSL Port	The HTTPS port on the Public Hostname to make the connection to. Default is 443.
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.
Backup SSL Port	The HTTPS port on the Public Hostname to make the connection to that is used when the backup of a redundant pair becomes active. Default is 443.
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.

Twilio SMS Notification

SMS notifications can be sent with the **Twilio 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). The Twilio account must then also be **defined in Ignition**.

CAUTION

This module requires the Alarm Notification module to be installed and enabled. If the Alarm Notification module is not installed or is disabled, this module will fault with a warning in the logs and display the state as "Faulted Missing Dependency" on the Platform > System > Modules page.

See the **system.twilio** section of the appendix for details on the scripting functions available for the Twilio SMS Notification Profile.

Create a Twilio SMS Notification Profile

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

3. Click on **Create Profile +**.

4. Select the **Twilio SMS Notification** profile type, and click **Next**.

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

Once your Notification profile is created, you can use it in an Alarm Pipeline.

Twilio SMS Notification Profile Settings

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

General

Setting	Description
Name	Name for this alarm notification profile.
Description	Description of the profile.
Enabled	Whether the profile is enabled or disabled.

Twilio

Setting	Description
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

Setting	Description
Auditing Profile	Select an audit profile from the dropdown list.

Twilio Voice Notification

Voice notifications can be sent with the [Twilio Notification module](#). To function, the Gateway must have Internet access and a Twilio account (www.twilio.com) with voice capabilities [defined in Ignition](#). This profile allows you to configure a custom message associated with the alarm using the Alarm Notification Pipeline via an automated phone call when an alarm is triggered. The alarm can optionally be acknowledged through the phone call.

Prerequisites



INDUCTIVE
UNIVERSITY

Twilio Voice Settings Review

[Watch the video](#)

When defining your Twilio account in Ignition, you'll already have noted your Account SID and Auth Token, but more settings are required for voice notification to function successfully. Alongside the SID and token, you'll also want to confirm the number listed in the My Twilio phone number field, as this is the from number where Twilio calls will originate once our notifications are set up in Ignition. Finally, make sure your Twilio Markup Language or [TwimML app](#) includes accurate Voice Configuration properties, like the [Request URL and Status Callback URL](#) webhooks so the Twilio service knows how to handle incoming calls and messages.

These URLs need to end with `/data/twilio/voice/message`. Navigate to your Twilio account **Phone Numbers > Manage > TwiML apps** page to update or add these properties.

Once all the required configuration is complete on the Twilio side, you can get started with creating a Twilio Voice Notification Profile.

CAUTION

This module requires the Alarm Notification module to be installed and enabled. If the Alarm Notification module is not installed or is disabled, this module will fault with a warning in the logs and display the state as "Faulted Missing Dependency" on the **Platform > System > Modules** page.

Messaging Configurations

Notification profile settings include enabling acknowledgement, pin validation, broadcasting notification calls to all users, and recording calls. Additionally, the language and text-to-speech voice to use for generating calls are selected when creating or editing the notification profile.

The Twilio module adds a User PIN field to user profiles, which is where the pin validation value used to authenticate users is pulled from. When enabled, validation is required before the user can read alarm information or acknowledge the alarm. When acknowledgement is enabled, the call will ask users if they want to acknowledge or ignore an alarm, and if the user selects acknowledge, the alarm state is changed to acknowledged.

If you choose to enable broadcasting, the voice message is sent to all users on the assigned roster at once. When broadcast notifications are used, the above PIN authentication and alarm acknowledgement settings are ignored. This means that broadcasted calls are unauthenticated by default, even as they might contain sensitive alarm information.

It's also important to be aware that when recording calls is enabled, the recording is stored and accessible in the Twilio dashboard. Recorded calls can be accessed in the Twilio dashboard after they are recorded. Select an auditing profile from the setting dropdown on the Voice Notification Profile page to store phone calls and acknowledgements to the audit system.

Pipeline Configuration

Like all notification methods, the Twilio Voice profile is selected in the Notification Block Editor in the Designer's pipeline configuration. The Twilio Voice profile includes two properties, From Number and Max Pin Attempts.

- From Number: The phone number configured in the Twilio dashboard, from which to send the phone call.
- Max Pin Attempts: The maximum number of attempts allowed for a user to correctly enter a PIN when pin validation is enabled.

The Call Script and Delay Between Messages (seconds) property are also displayed here.

Expanding the Call Script list allows you to customize the alarm message the user hears through a call script text. There are several points where the call may stop to collect user input, so the call script is the list of messages used by the text-to-speech engine to generate the call.

Call Script Options

The following table is a full list of configurations for the call script, with their respective defaults:

Configuration	Description	Default
Greeting	Introductory message.	Hello, this is an Ignition alarm notification.
PIN Challenge	A message asking the user to enter their PIN number. Must instruct them to press '#' when finished.	Please enter your pin, followed by the pound sign.
Non-PIN Challenge	When a PIN is not required, this message is played to instruct the users to press any key to continue.	Please press any key to continue.
Invalid PIN	A message telling the user their PIN was invalid, and to try again.	Invalid pin number. Please re-enter your pin, followed by the pound sign.
Summary	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.	There are %d events.
Active Message	The message to play for active alarms. This message can reference properties of alarms by using	At {eventTime hh:mm:ss}, alarm named {name} became {eventState} with a value of {eventValue}

Configuration	Description	Default
	syntax like <code>{propertyName}</code> .	
Clear Message	The message to play for cleared alarms. This message can reference properties of alarms by using syntax like <code>{propertyName}</code> .	At <code>{eventTime hh:mm:ss}</code>, alarm named <code>{name}</code> became <code>{eventState}</code> with a value of <code>{eventValue}</code>
Option 1: Acknowledge	Tells the user to press '1' to acknowledge.	Press 1 to acknowledge
Acknowledge Confirmation	Tells the user that the alarm event has been successfully acknowledged.	Alarm has been acknowledged.
Option 2: Ignore	Tells the user to press '2' to ignore/bypass the alarm.	Press 2 to ignore
Ignore Confirmation	Tells the user the alarm event has been ignored.	Alarm has been ignored.
Option 3: Repeat	Tells the user to press '3' to repeat the last message.	Press 3 to repeat
Closing	Closing message	Thank you, goodbye.

Voice Functions

See the [system.twilio](#) section of the appendix for more details on the `sendPhoneCall` function available for the Twilio Voice Notification Profile.

Create a Twilio Voice Notification Profile



INDUCTIVE
UNIVERSITY

Twilio Voice Notification
Profile

Watch the video

1. Go to the **Services** section of the Gateway Webpage.
2. Select **Alarming > Notification** from the menu on the left side.
3. Click on **Create Profile +**.
4. Select the **Twilio Voice Notification** profile type, and click **Next**.
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 Alarm Notification Profile**.

Once your Notification profile is created, you can use it in an Alarm Pipeline.

Twilio Voice Notification Profile Settings

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

General

Setting	Description
Name	Name for this alarm notification profile.
Description	Description of the profile.
Enabled	Whether the profile is enabled or disabled.

Twilio Settings

Setting	Description
Twilio Account	Select a Twilio account from the dropdown list.
Allow Acknowledgement	Enables users to respond to notifications directly from the call; also known as Two-Way notification . If enabled, the Public Hostname property in the Twilio Account must be configured. Default is true.
Require PIN Authentication	Whether to authenticate users before alarm summary. Default is true.
Broadcast Notifications Enabled	When enabled, all users on the connected on-call roster will be notified. Default is false. Note: It is recommended to enable this property if there is no valid inbound public hostname configured on the Twilio account.
Record Calls	Activates Twilio's phone call recording feature. Default is false.
Voice	Selects the text-to-speech voice to use.
Language	Selects the language to use.
Answer Timeout	Sets the amount of time, in seconds, a call is allowed to go unanswered before it's considered failed. Default is 60 seconds.
Max Call Duration	Sets the maximum time, in minutes, for a call to last. Default is 5 minutes.

Other

Setting	Description
Auditing Profile	Select an audit profile from the dropdown list. If an audit profile is selected, phone calls and acknowledgements will be stored to the audit system. Note that alarm acknowledgements are also stored to the alarm journal.

Twilio WhatsApp Notification

WhatsApp SMS notifications can be sent with the [Twilio Notification module](#). This profile 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). The Twilio account must then also be [defined in Ignition](#).

CAUTION

The Alarm Notification module must be installed and enabled. If the Alarm Notification module is not installed or is disabled, a warning will appear in the logs and display the state as "Faulted Missing Dependency" on the Platform > System > Modules page.

See the [system.twilio](#) section of the appendix for details on the two scripting functions available for the Twilio WhatsApp Notification Profile.

Message Types

WhatsApp allows two types of messages, template and free-form. Upon initial communication, Twilio can only send these pre-approved template messages.

- **Template Messages:** Template messages are configurable messages on Twilio that can be sent to users via Twilio WhatsApp. Templates are generic messages, but parameters allow templates to be a bit more versatile. Template Parameters are entered in the Notification Block settings in the Designer. Refer to the [WhatsApp Message Template Overview](#) for more information on template messages.
- **Free-Form Messages:** Any message that is not a pre-approved template message is considered free-form. You can configure your free-form messages using the Message field on the alarm or on the alarm Notification block.

The only way to send free-form SMS messages in WhatsApp is for the user to respond to a template in order to open a 24-hour messaging session. Once a user replies to a template message, then the alarm notifications will send out the configured free-form messages. The 24-hour Session will remain open for 24 hours from the last message received from the user.

Create a Twilio WhatsApp Notification Profile

1. Go to the **Services** section of the Gateway Webpage.
2. Select **Alarming > Notification** from the menu on the left side.
3. Click on **Create Profile +**.
4. Select the **WhatsApp SMS Notification** profile type, and click **Next**.
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 Alarm Notification Profile**.

Once your Notification profile is created, you can use it in an Alarm Pipeline. Note that your **user contact information** needs to be set to **WhatsApp SMS** to receive messages.

Twilio WhatsApp Notification Profile Settings

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

General

Setting	Description
Name	Name for this alarm notification profile.
Description	Description of the profile.
Enabled	Whether the profile is enabled or disabled.

Twilio

Setting	Description
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

Setting	Description
Auditing Profile	Select an audit profile from the dropdown list.

WhatsApp

Setting	Description
WhatsApp Service SID	Twilio's messaging service ID, found on Twilio's Console when navigating under Develop > Messaging > Services.

Voice Notification Profile



INDUCTIVE
UNIVERSITY

Voice Notification Profile

[Watch the video](#)

The Voice Notification Profile allows you to deliver alarm notifications to users by telephone using a SIP-compatible phone system. Messages are generated dynamically using text-to-speech (TTS), with support for multiple languages and user acknowledgments.

This profile is ideal for environments where voice communication is preferred or required as part of an alarm response workflow.

NOTE

To use this feature, you must install a compatible [Voice Notification module](#) and at least one TTS voice. See the [Voice Notification Downloads](#) Knowledge Base article for supported voices.

CAUTION

This profile requires the Alarm Notification module. If that module is not installed or is faulted, the Voice Notification Profile will also fault and appear as “Missing Dependency” on the **Modules** page.

Core Features

- Delivers alarm notifications over telephone via SIP
- Uses high-quality text-to-speech for messages
- Supports multiple languages and user-specific preferences
- Allows alarm acknowledgment by phone
- Logs call and acknowledgment events to the audit log
- Supports message consolidation and optional PIN protection

How It Works

The Voice Notification module uses Session Initiation Protocol (SIP) to place calls over VOIP networks. Once connected to a SIP Gateway, Ignition can call physical phones, softphones, or external systems without dedicated telephony hardware.

Asterisk-based PBX systems, hosted SIP services, or direct SIP devices can all be configured with this profile.

Create a Voice Notification Profile

Follow these steps to create a Voice Notification Profile.

1. In the Gateway, go to **Services > Alarming > Notification**.
2. Click **Create Profile +**.
3. Select **VOIP Voice Notification**, then click **Next**.

The screenshot shows a 'Create Alarm Notification Profile' dialog box. On the left, there is a sidebar with 'Create Alarm Notification Profile' and two steps: '1. Profile Selection' and '2. Profile Setup'. The main area contains five radio button options, each with a description. The 'VOIP Voice Notification' option is selected and highlighted with a red border. At the bottom, there are 'Cancel' and 'Next' buttons.

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

4. Enter the following required values as an example:

- **Name:** Voice

- **Gateway Address:** sip.inductiveautomation.com
- **Username/Account:** [user@inductiveautomation.com](#)

GENERAL *Required

Name *

Voice

Enter a name for the alarm notification profile.

Description

Enabled
☒ Set whether this alarm notification profile is enabled or disabled within the Gateway.

VOIP GATEWAY SETTINGS

Gateway Address

sip.inductiveautomation.com

The ip address or domain name of your SIP Gateway.

Username/Account

user@inductiveautomation.com

5. Click **Create Alarm Notification Profile**.

6. The profile appears in the list and will display **Registered with VOIP Host** once configured correctly.

Alarm Notification Profiles						
Filter alarm notification profiles...			View	20	Test ▶ Create Profile +	
☐	NAME	DESCRIPTION	ENABLED	TYPE	STATUS	⚙
☐	Voice		true	VOIP Voice Notification	✓ REGISTERED WITH VOIP HOST	⋮

Showing 1 of 1 items

Once your profile is active, you can assign it to a Notification block within an [Alarm Pipeline](#).

Voice Notification Profile Settings

Main

Property Name	Description
Name	Name of the notification profile.
Description	Optional description of the profile.
Enabled	Whether the profile is currently active.

VOIP Gateway Settings

Property Name	Description
Gateway Address	IP address or domain name of the SIP Gateway.
Username/Account	SIP username or account name, if required.
Password	Password for the SIP account. Options include None, Embedded, or Referenced.
Outbound Proxy	Optional SIP proxy address.

Call Settings

Property Name	Description
Answer Timeout (seconds)	Time to wait for the call to be answered. Default: 60.
Max Queue	Max number of notification jobs in the queue.
Max Wait	Maximum time (in minutes) a call can wait in queue.
Max Call Duration (minutes)	Time limit for a single call. Default: 5.
Use Fair Scheduling	When enabled, places only one call per job at a time.
Voice Rate	Voice playback speed. Options include Normal, Slow, or Fast.

Advanced

Property Name	Description
Authorization ID	Optional alternate authorization name for SIP services.
SIP Port	Port used to send SIP messages. Default: 5060.
RTP Port	Port used for voice media (audio). Default: 8000.
Local Bind Address	Local IP address used for SIP communication.
Backup Bind Address	Bind address to use in redundant mode.
Public Bind Address	Public-facing bind address, if needed.
Media Debug Enabled	When enabled, call audio is recorded to the temp directory.

Auditing

Property Name	Description
Audit Profile	Audit profile to record call events and acknowledgments. Acknowledgments are also stored in the Alarm Journal.

Call Lifecycle

Voice notifications proceed through these steps:

1. An alarm event reaches a voice-enabled Notification block in a pipeline.
2. Users with valid phone contact types and matching schedules are selected.
3. Calls are placed one at a time from the queue.
4. When the user answers, they're prompted to press a key or enter a PIN (if required).
5. The alarm message is delivered.
6. The user can acknowledge, ignore, or repeat the message.
7. Call results are logged and the next call proceeds.

Notification Security PIN

You can require users to enter a personal identification number (PIN) before they hear a voice alarm notification. This helps ensure that the person receiving the alarm is authorized to do so—especially in shared phone environments.

Voice alarm notifications typically allow users to acknowledge alarms by pressing **1**. Requiring a PIN prevents unauthorized acknowledgment.

Add a PIN to a User Profile

To assign a PIN to a user:

- 1. In the Gateway, go to **Platform > Security > User Sources**.
- 2. Under your user source (such as **default**), click **Manage Users**.

User Sources

Filter user sources...

View20

Verify User Source

Create User Source +

☐	NAME	DESCRIPTION	TYPE	
☐	default	This user source profile was automatically created during a startup	internal	Edit Rename Duplicate Manage Users Disable Delete
☐	opcua-module	OPC UA clients will authenticate against this profile by default.	internal	

Showing 2 of 2 items

- 3. Locate the user and click **Edit**.

×

default Manage Users

Users

Roles

Create User +

USERNAME	ROLES	SCHEDULE	
admin	Administrator	Always	
Bob		Always	Edit Assign Roles Delete

- In the user editor, confirm that the **Phone** contact field is populated.
- Enter a numeric **Security PIN** and click **Save Changes**.

CONTACT

User Contact Information

Add +

TYPE	VALUE
Email	admin@mycompany.com
Phone	555-555-5555

EXTENDED PROPERTIES

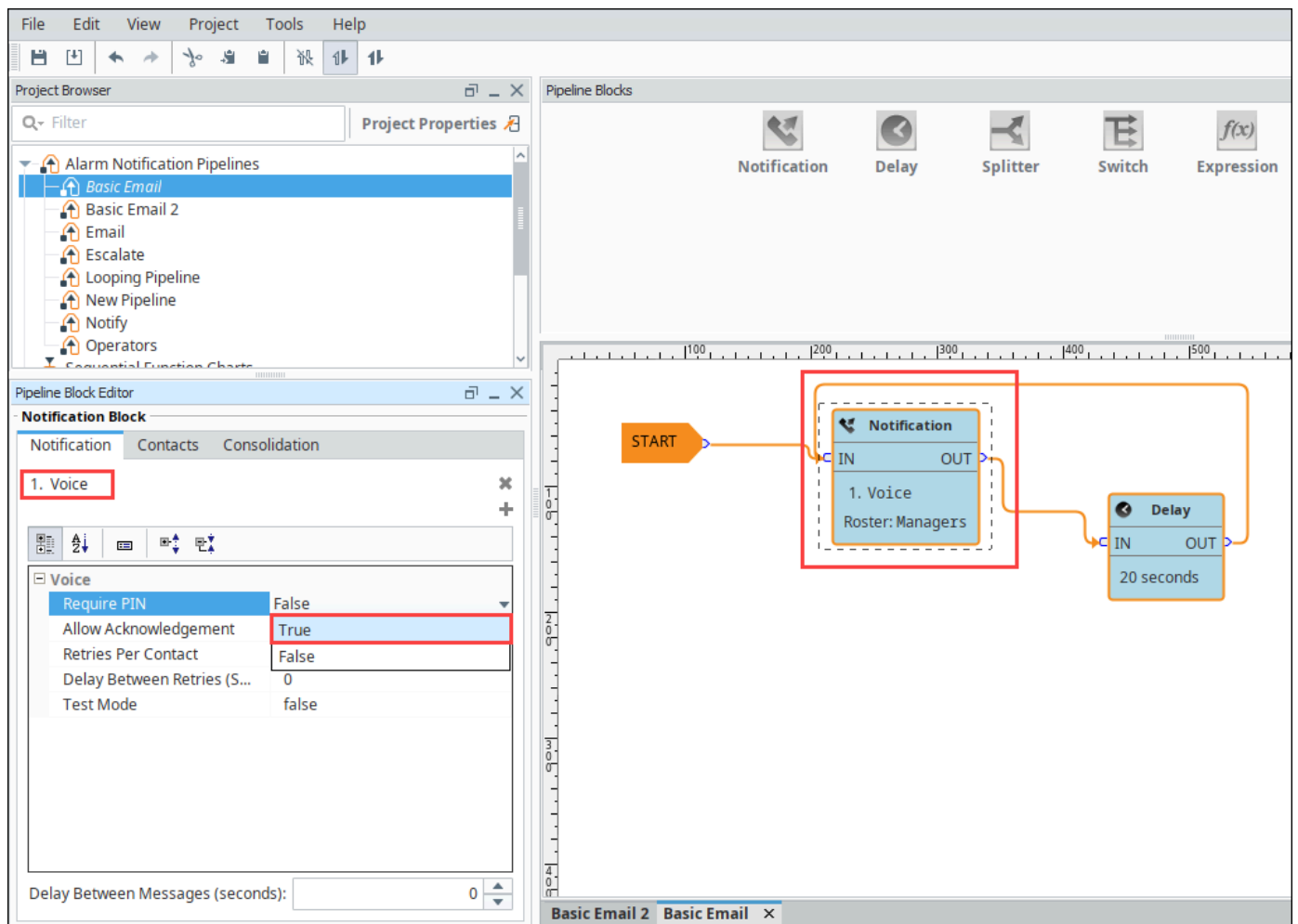
Security PIN

8765432

Require PIN in the Notification Block

Once users have PINs, configure your pipeline to require them:

- In the **Designer**, open your **Alarm Notification Pipelines**.
- Select the **Notification Block** in the workspace.
- On the **Notification** tab, select the **Voice** profile.
- Set **Require PIN** to **True**.



Now, when a user receives a call, they'll be prompted to enter their PIN followed by the pound key before the message plays.

Voice Notification Scripts



INDUCTIVE
UNIVERSITY

Voice Notification Profile

[Watch the video](#)

Voice Notification scripts define what users hear when they receive an alarm call. These scripts use **text-to-speech (TTS)** voices provided by separate language modules, allowing you to deliver alarm messages in multiple languages and with different voices.

Ignition includes default scripts for supported languages.

NOTE

Scripts define spoken phrases used during voice alarm notifications. These are not scripting code, but text that is read aloud to users over the phone.

Install TTS Voice Modules

The Voice Notification module requires at least one TTS voice module.

Available TTS voices include:

Voice Name	Language
Katherine	English
Laura	Italian
Sara	Spanish
Suzanne	French
Alex	German

For detailed download and installation steps, see the [Voice Notification Downloads](#) Knowledge Base article.

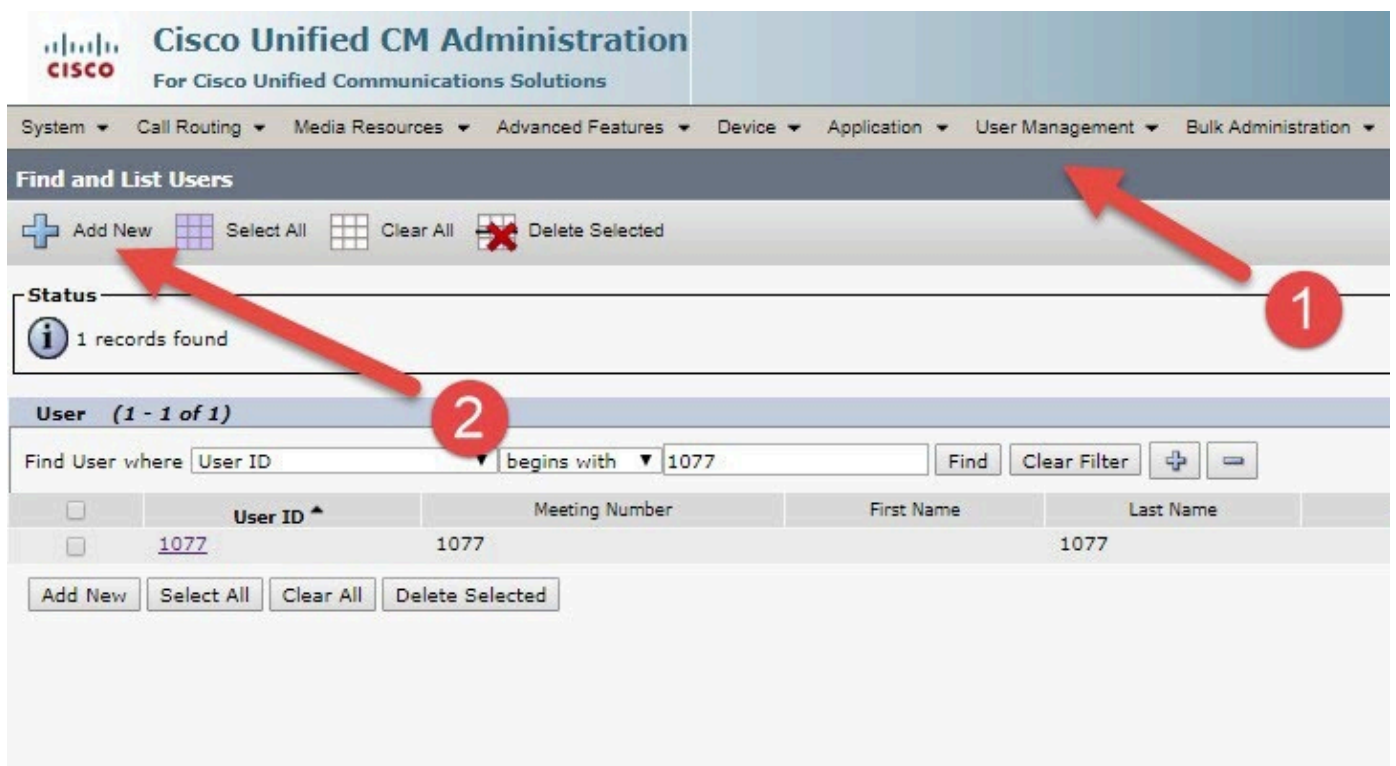
Registering Voice Alarm Module to CUCM

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.

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



2. Create a user with digest credentials.

i NOTE

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

Cisco Unified CM Administration
For Cisco Unified Communications Solutions

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

End User Configuration

Save Delete Add New

Status

Status: Ready

User Information

User Status	Enabled Local User	
User ID*	1077	
Password		Edit Credential
Confirm Password		
Self-Service User ID	1077	
PIN		Edit Credential
Confirm PIN		
Last name*	1077	
Middle name		
First name		
Display name		
Title		
Directory URI		
Telephone Number		
Home Number		
Mobile Number		
Pager Number		
Mail ID		
Manager User ID		
Department		
User Locale	English, United States ▾	
Associated PC/Site Code		
Digest Credentials		
Confirm Digest Credentials		
User Profile	Use System Default("Standard (Factory Default) Us ▾	View Details

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

Cisco Unified CM Administration
For Cisco Unified Communications Solutions

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

Find and List Phones

Add New Select All Clear All Delete Selected Reset Selected Apply Config to Selected

Status

551 records found

Phone (1 - 50 of 551)

Find Phone where ▾ begins with ▾ [Find](#) [Clear Filter](#)

Select item or enter search text ▾

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: 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.



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

NOTE

Depending on your system, the **Media Termination Point Required** option may need to be checked.

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

Remote Gateway Notification



INDUCTIVE
UNIVERSITY

Remote Alarms

[Watch the video](#)

The Remote Gateway Notification profile allows one Gateway to send alarm notifications using a notification profile hosted on a different Gateway in the same Gateway Network. This lets you centralize email, SMS, and voice delivery infrastructure while distributing alarm generation across remote sites.

This is especially useful when remote Gateways lack direct access to email servers or telephony hardware. The remote Gateway triggers the alarm; the central Gateway handles the delivery.

! INFO

The **Alarm Notification module** must be installed and licensed on all participating Gateways. However, notification-specific modules (like SMS or Voice) are only required on the Gateway that executes the notification.

! INFO

Both Gateways must be connected via the Gateway Network.

Create a Remote Gateway Notification Profile

Follow these steps to create a Remote Gateway Notification Profile.

1. In the Gateway, go to **Platform > Alarming > Notification**.
2. Click **Create Profile +**.
3. Select **Remote Gateway Notification**, then click **Next**.

Create Alarm Notification Profile

Create Alarm Notification Profile

1. Profile Selection

2. Profile Setup

☐ **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.

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

Cancel

Next

4. Enter the required values using the following example:

- **Name:** Remote
- **Gateway:** IP address or name of the remote Gateway.
- **Notification Profile:** Email 1

GENERAL

*Required

Name *

Remote

Enter a name for the alarm notification profile.

Description

Enabled

☒ Set whether this alarm notification profile is enabled or disabled within the Gateway.

REMOTE GATEWAY

Gateway

172.17.0.1

Notification Profile

Email 1

ADVANCED

*Required

Retry Delay (Milliseconds) *

10000

Delay, in milliseconds, between retries of a failed attempt to synchronize with the remote profile.

Max Queued Alarm Transitions *

10000

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.

5. Click **Create Alarm Notification Profile**.

6. The profile will appear in the list and show a status of **Connected** or **Registered** once the connection is successful.

Once created, this profile can be used in any local pipeline or notification block.

Remote Gateway Notification Profile Settings

General

Property Name	Description
Name	Name of the notification profile.
Description	Optional description of the profile.
Enabled	Whether the profile is currently active.

Remote Gateway

Property Name	Description
Gateway	IP address or hostname of the remote Gateway in the Gateway Network.
Notification Profile	Name of the alarm notification profile on the remote Gateway to be used for sending notifications.

Advanced

Property Name	Description
Retry Delay	Time to wait (in milliseconds) before retrying a failed synchronization with the remote Gateway. Default: 10000 .
Max Queued Alarm Transitions	Maximum number of alarm transitions to hold in the queue while the remote Gateway is unavailable. Default: 10000 .

Assign a Remote Pipeline to an Alarm

After creating the profile, you can assign a remote pipeline to an alarm directly from the Tag Editor.

1. In the Designer, open the **Tag Browser** and double-click the tag you want to configure.

Tag Browser				
↺ ↻ 🔍 📁 🔌 🔄 🔗 🔧				
▶ Random				
▶ ReadOnly				
▶ Realistic				
▶ Sine				
▼ Writeable				
▶ WriteableBoolean1 OPC	☐		Boolean	
▶ WriteableBoolean2 OPC	☐		Boolean	
▶ WriteableDouble1 OPC	0		Double	
▶ WriteableDouble2 OPC	0		Double	
▶ WriteableFloat1 OPC	0		Float	🔔
▶ WriteableFloat2 OPC	0		Float	
▶ WriteableInteger1 OPC	0		Integer	
▶ WriteableInteger2 OPC	0		Integer	

2. Scroll to the **Alarms** section and click the **Edit** icon for the alarm.
3. In the alarm editor, scroll to the **Notification Settings** section.
4. Under **Active Pipeline**, select a pipeline from the remote Gateway (e.g., Remote/Pipeline1).
5. Click **Commit** to save your changes.

Tag Editor

WriteableFloat1 > Alarms

default

Alarms

Fault1 - Above Setpoint, High

High Alarm - Above Setpoint, Critical

Low Alarm - Below Setpoint, Medium

Fault1

🔍

📄

🔧

🔄

🔗

+

🗑️

Deadband	0	🔗	^
Deadband Mode	Absolute	▼	
Active delay (seconds)	0	🔗	
Clear delay (seconds)	0	🔗	
▼ Notification			
Ack Pipeline		▼	🔗
Active Pipeline	Core/Complex Pipeline	▼	🔗
Clear Pipeline		▼	🔗
▼ Phone Call Settings			
Custom Message			🔗
▼ Email Notification Properties			
Custom Message			🔗
Custom Subject			🔗
▼ SMS Notification Properties			
Custom Message			🔗

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

Commit

Revert

Once saved, the tag will send alarm notifications using the remote Gateway's pipeline.

 INFO

Only the Gateway executing the notification must have the necessary notification modules installed (Email, SMS, or Voice).

You can also use the profile in a **Notification Block** within an alarm pipeline.

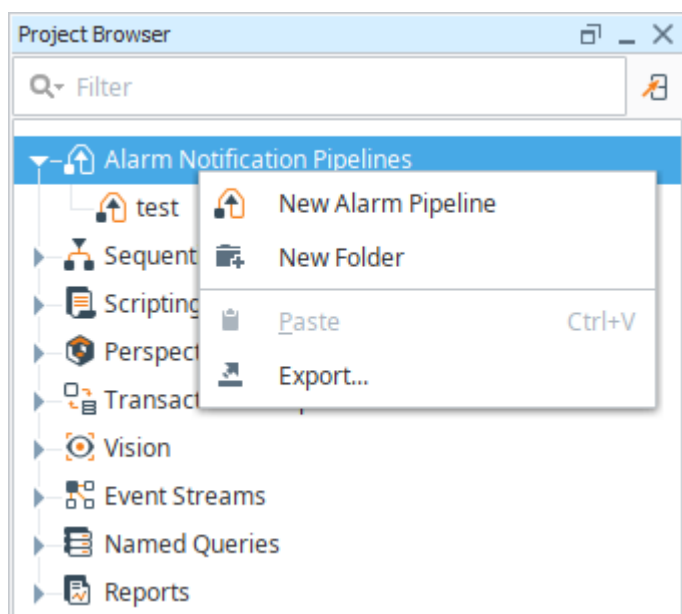
Alarm Notification Pipelines

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

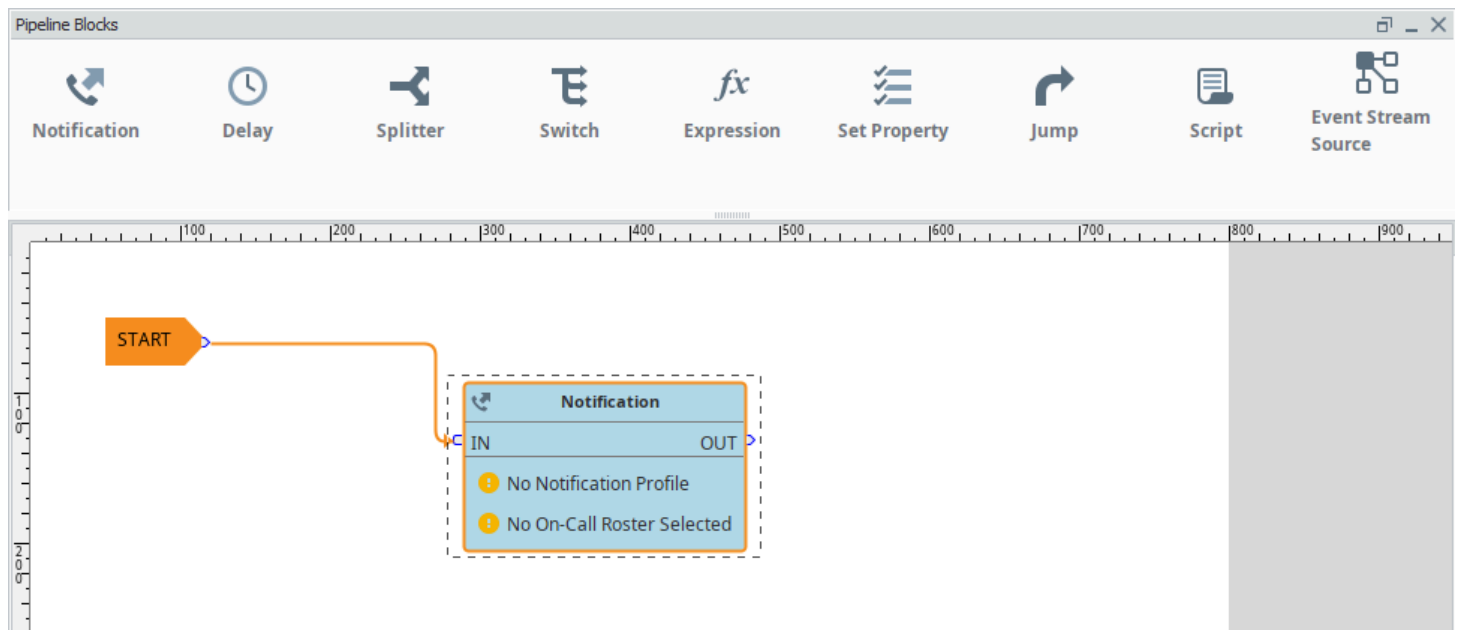
Before you create a pipeline in the Designer, you need to configure at least one **alarm notification profile** and at least one **roster** in the Gateway. Each user in the roster must have valid contact information, such as an email address or phone number, and that information must match the address used to deliver notifications in order for acknowledgements to be processed successfully. Then in the Designer, you create the Pipelines under the **Global** node of the **Project Browser**.

Alarm Pipelines, unlike other types of resources such as windows and Transaction Groups, are global resources, and are not part of a project. A defined pipeline runs in the Gateway, and you can see the same pipelines available to edit regardless of the project that is currently open. If you already have a remote Gateway configured to support alarm notifications, you can expand the already significant power of alarm pipelines by also exposing alarm pipelines that are configured on the **Remote Ignition Gateway** system.



When you create a new pipeline, or select one to edit, the Designer Switches to the Pipeline Workspace. The workspace is 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, you can write the output of a downstream block back to the input of an earlier block, creating a loop. Be careful 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 **Services > Alarming > Pipelines** screen on the Ignition Gateway. It is possible to forward events from one pipeline to another 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 that 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, and 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.

Display and Modify the Properties

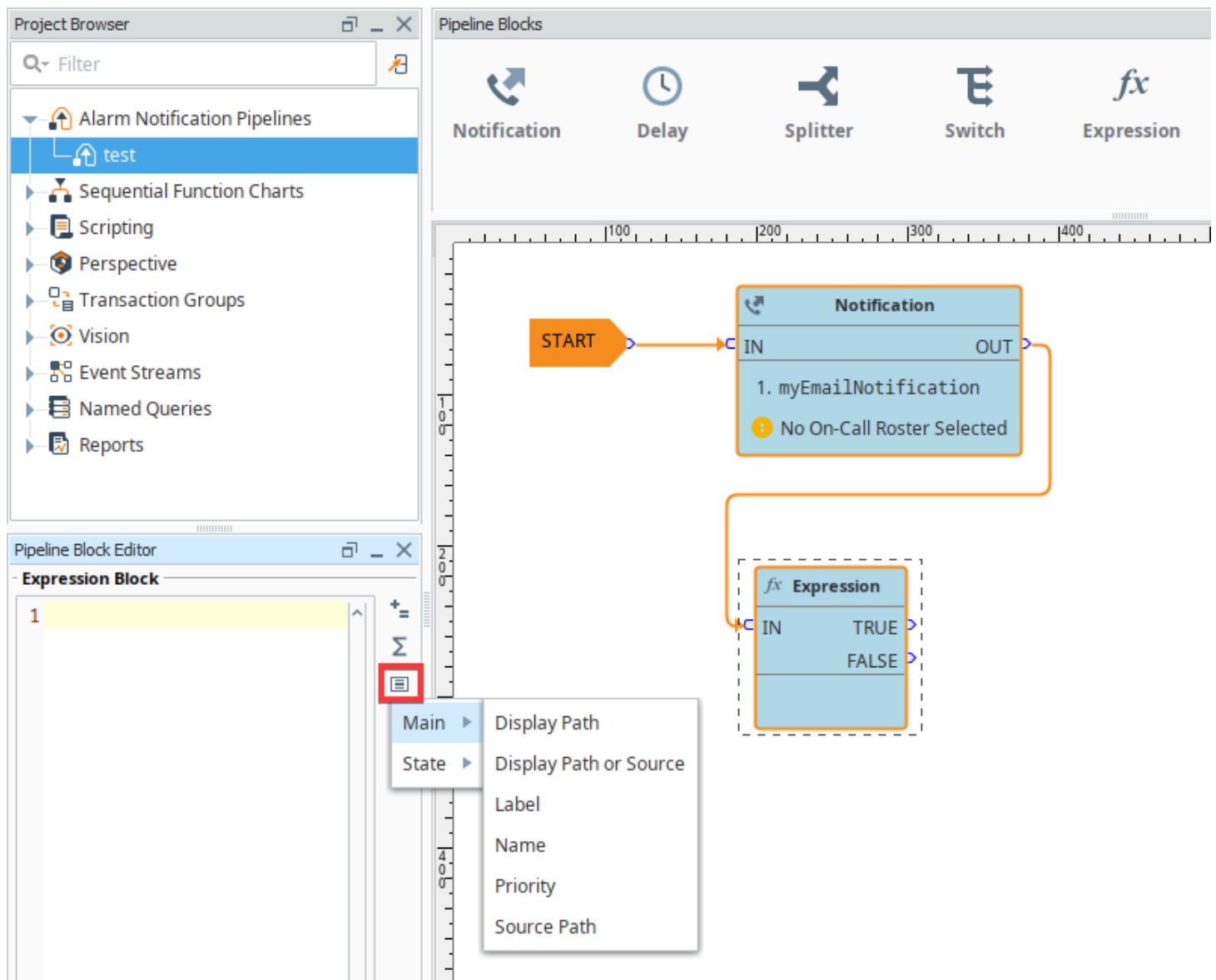
Click on the gray area of the Pipeline workspace or anywhere there isn't a block.

In the lower-left corner of the Designer, the Pipeline Properties are displayed.

Property	Description
Dropout Condition	Dropout conditions 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, cleared, or acknowledged. They are:

- Display Path
- Display Path or Source
- Label
- 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 Property Reference](#) section for information on alarm properties and how they are used.

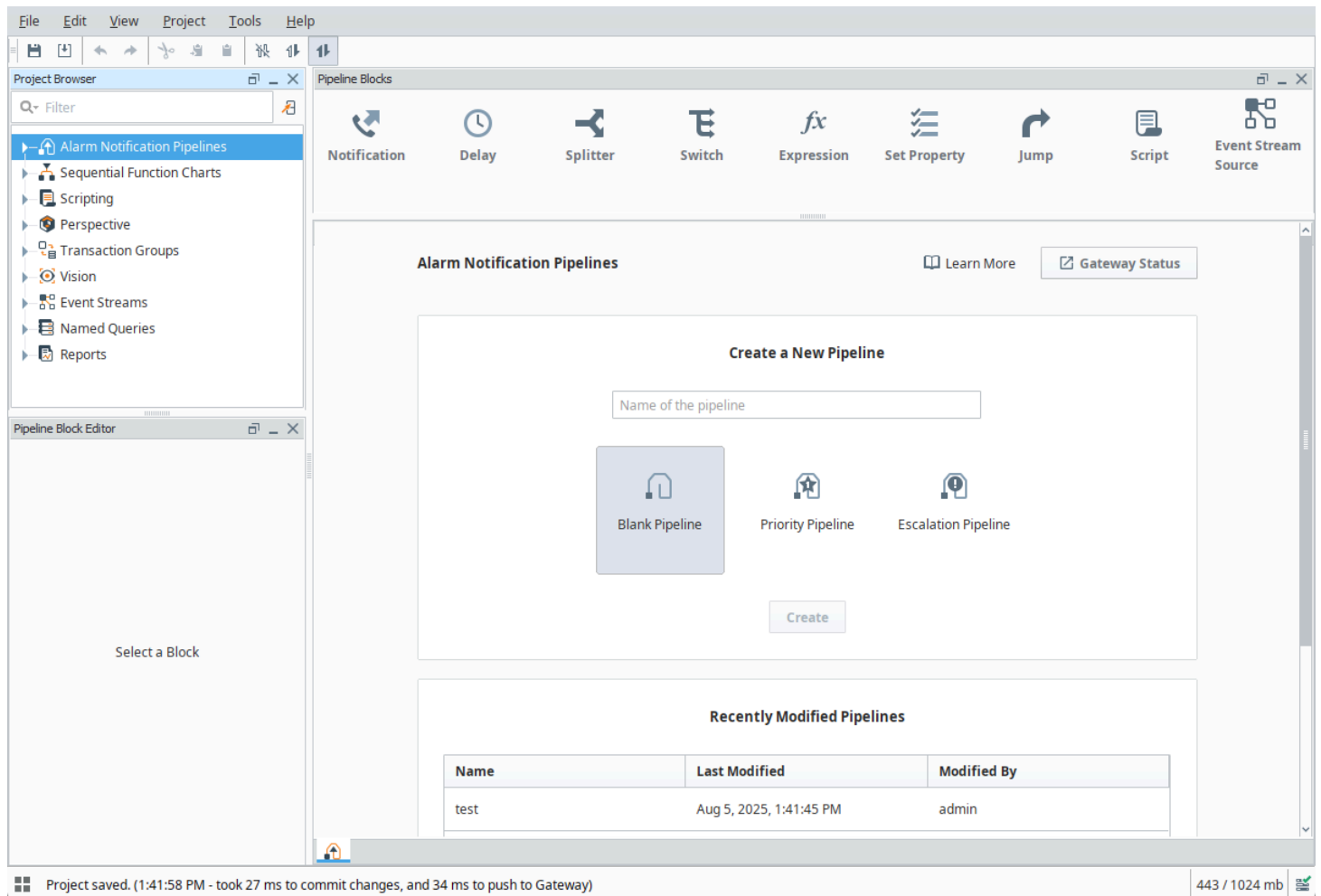
Alarm Pipeline Designer Interface

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.

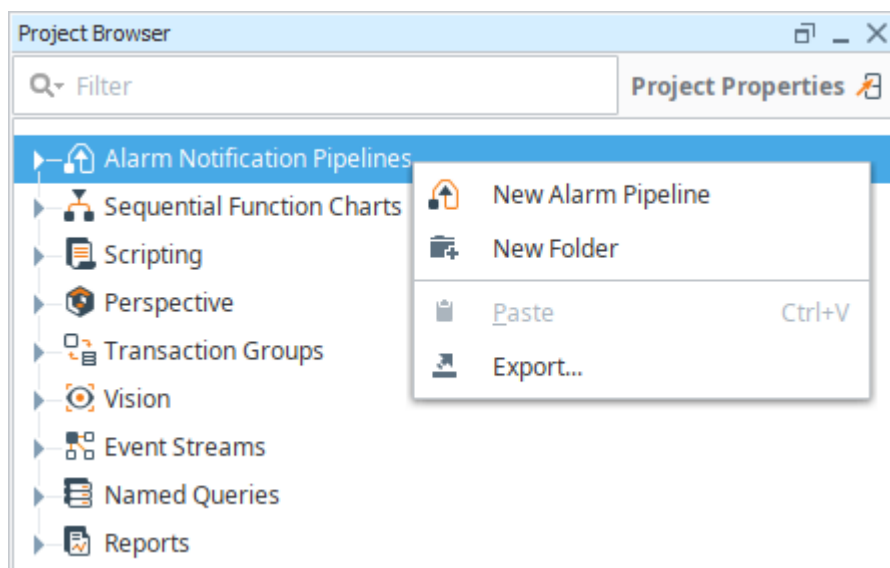
On the Alarm Notification Pipeline Welcome tab, you'll notice that there are some basic template pipelines to help get you started creating your own alarm notification pipelines. You can choose from the Priority and Escalation pipelines, or have the option to start from the Blank pipeline. These pipelines 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.



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, import pipelines from other projects, and export your existing pipelines. Like other resources, **pipelines are inheritable**, so a pipeline can be configured in one project, and inherited by another.



Pipeline Blocks

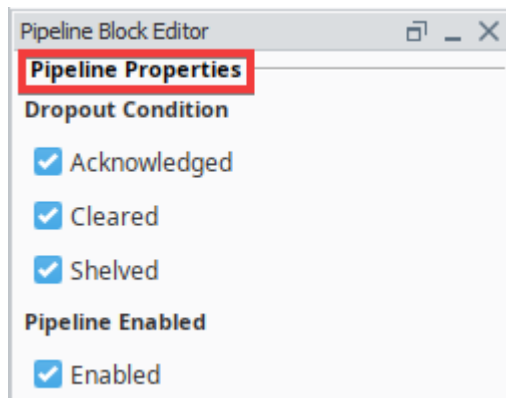
Pipeline Blocks are the building blocks of the Alarm Pipelines and are located at the top of the Designer. Each pipeline block has unique functionality that performs tasks such as sending notifications, alarm escalation, setting a delay, and evaluating an expression. Simply drag any of the pipeline blocks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



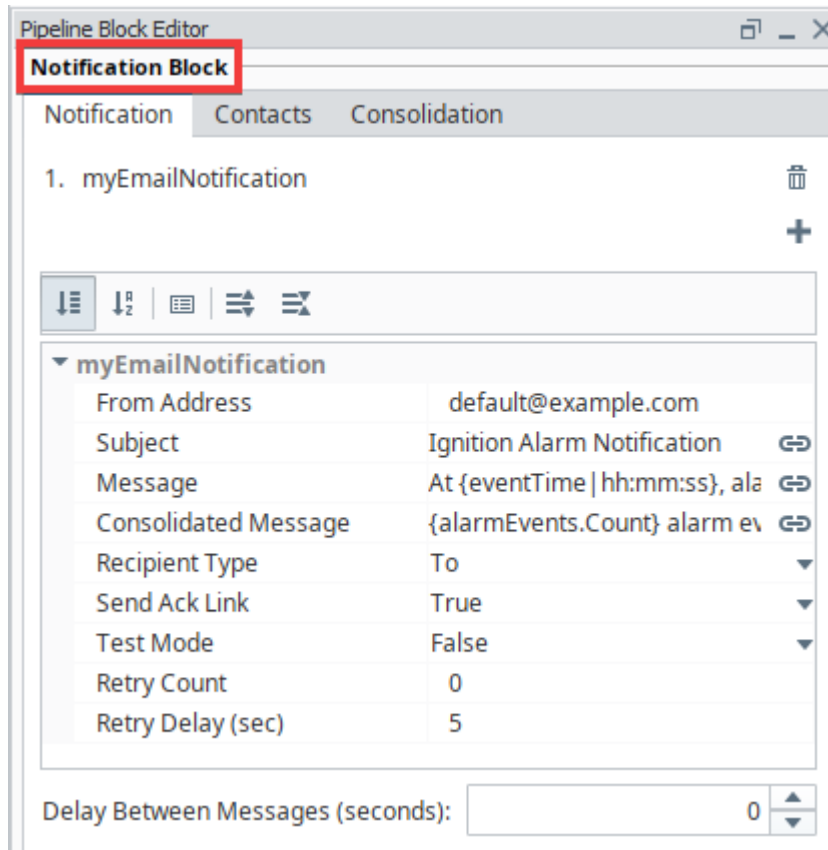
Pipeline Block Editor

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 images below show the Pipeline Block Editor properties based on function.

Pipeline Properties



Notification Block



The screenshot shows the 'Pipeline Block Editor' window. The 'Notification Block' tab is selected and highlighted with a red box. Below the tab, there are three sub-tabs: 'Notification', 'Contacts', and 'Consolidation'. The 'Notification' sub-tab is active, showing a list of notification blocks. The first block is '1. myEmailNotification'. Below this list is a table of properties for the selected block. The properties are: From Address (default@example.com), Subject (Ignition Alarm Notification), Message (At {eventTime | hh:mm:ss}, ala), Consolidated Message ({alarmEvents.Count} alarm ev), Recipient Type (To), Send Ack Link (True), Test Mode (False), Retry Count (0), and Retry Delay (sec) (5). At the bottom, there is a 'Delay Between Messages (seconds):' field with a value of 0.

myEmailNotification	
From Address	default@example.com
Subject	Ignition Alarm Notification
Message	At {eventTime hh:mm:ss}, ala
Consolidated Message	{alarmEvents.Count} alarm ev
Recipient Type	To
Send Ack Link	True
Test Mode	False
Retry Count	0
Retry Delay (sec)	5

Delay Between Messages (seconds): 0

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 the Operators roster via email. If an operator doesn't respond to the notification within 5 minutes, a second email will be sent to the Managers roster. 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.

FileEditViewProjectToolsHelp

Project Browser

FilterProject Properties

Alarm Notification Pipelines

test

Sequential Function Charts

Scripting

Perspective

Transaction Groups

Vision

Event Streams

Named Queries

Pipeline Block Editor

Notification Block

Notification

Contacts

Consolidation

1. myEmailNotification

myEmailNotification

From Address

default@example.com

Subject

Ignition Alarm Notification

Message

At {eventTime | hh:mm:ss}, al

Consolidated Message

{alarmEvents.Count} alarm e

Recipient Type

To

Send Ack Link

True

Test Mode

False

Retry Count

0

Retrv Delay (sec)

5

Delay Between Messages (seconds):0

Pipeline Blocks

Notification

Delay

Splitter

Switch

Expression

Set Property

Jump

Script

Event Stream Source

START

Notification

IN

OUT

1. myEmailNotification

Roster: Operators

Delay

IN

OUT

5 minutes, 0 seconds

Notification

IN

OUT

1. myEmailNotification

Roster: Managers

test

BlockComponent (562,245) [190x120]

387 / 1024 mb

Simple Pipeline

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

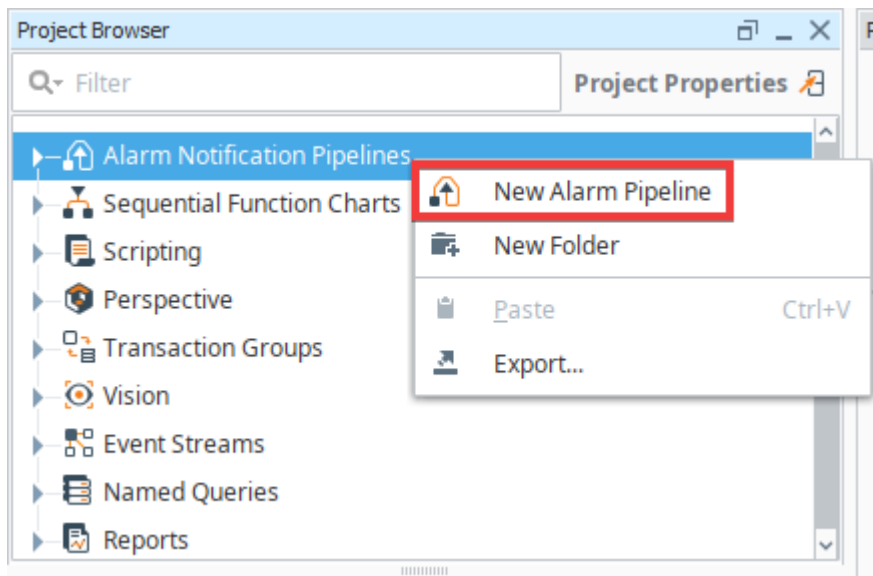
ⓘ CONFIGURE ALARM NOTIFICATION PROFILE AND ON-CALL ROSTER

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.

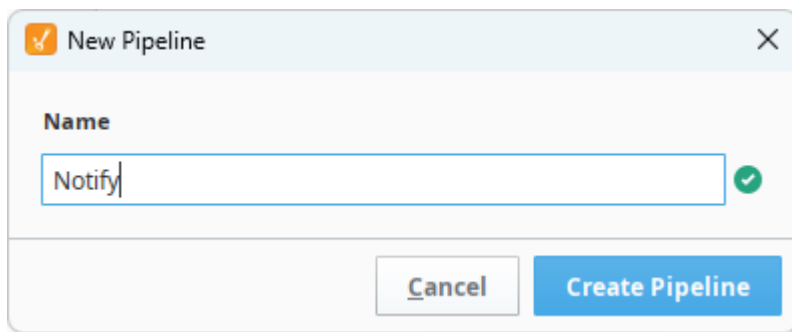
Create an Alarm Notification Pipeline

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

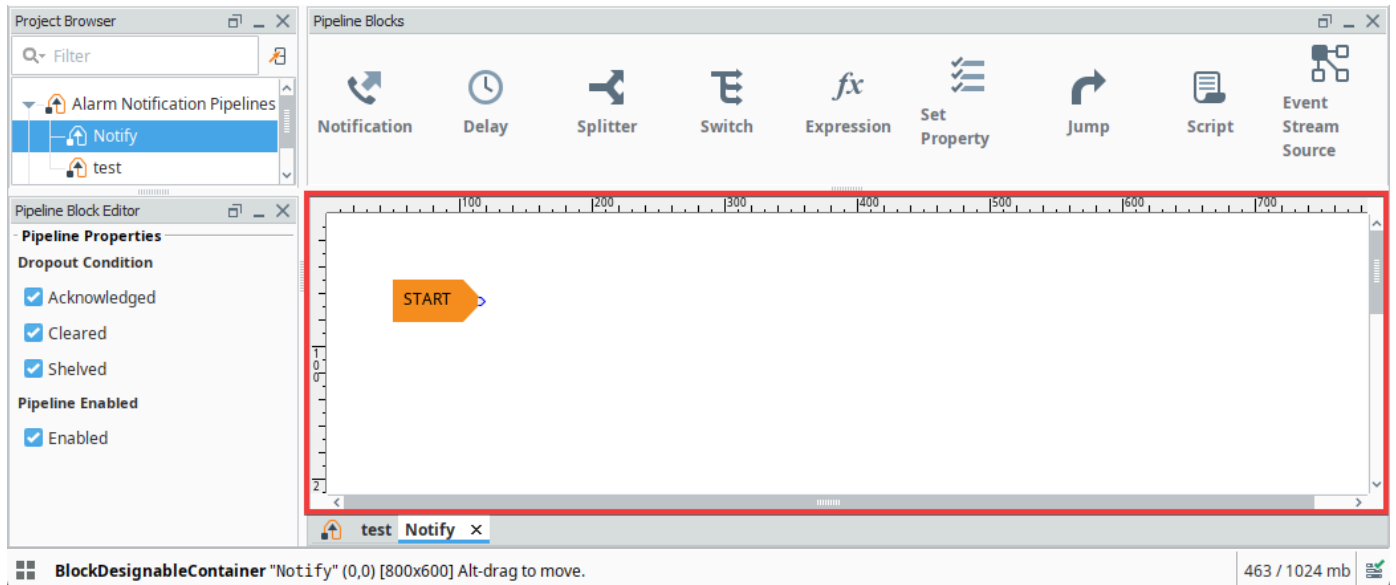
1. In the **Designer**, create a new pipeline in the **Project Browser** by right-clicking on Alarm Notification Pipelines, then clicking **New Alarm Pipeline**.



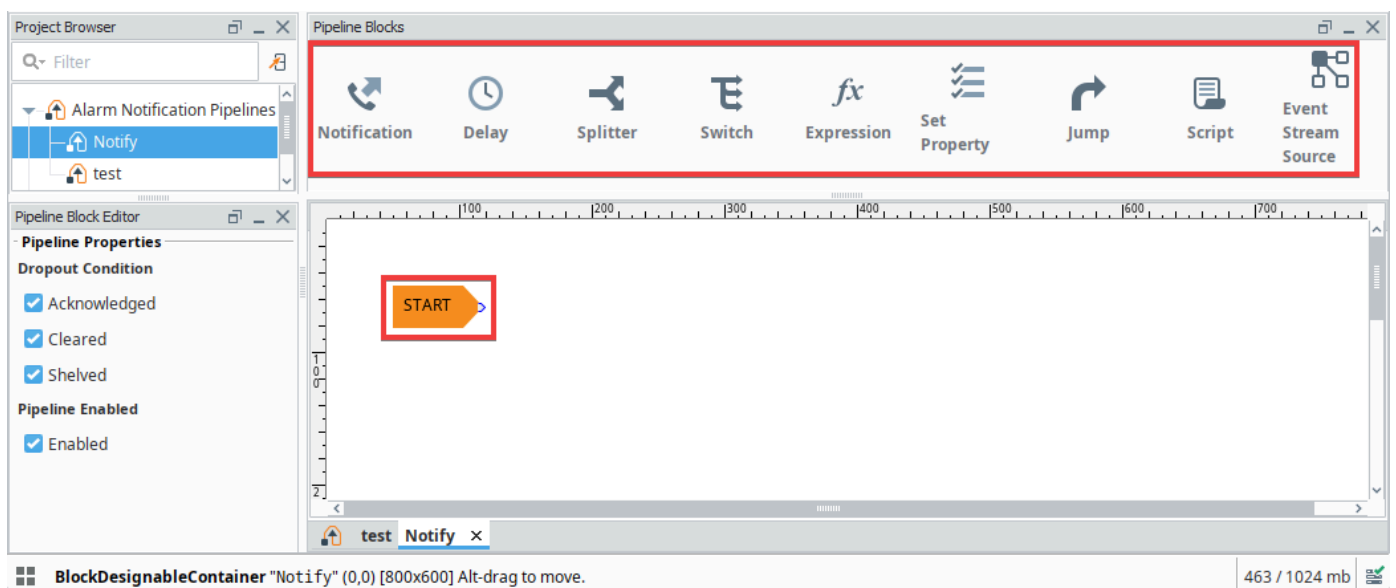
2. You will be prompted to enter a name for the new pipeline. It is good practice to use the name to describe what the pipeline does. In this case, we are naming the pipeline "Notify". Enter the name you are using, then click **Create Pipeline**.



3. After creating a new pipeline, the Designer switches to the Pipeline Workspace. The workspace is essentially a blank canvas and can contains various **Pipeline Blocks**.

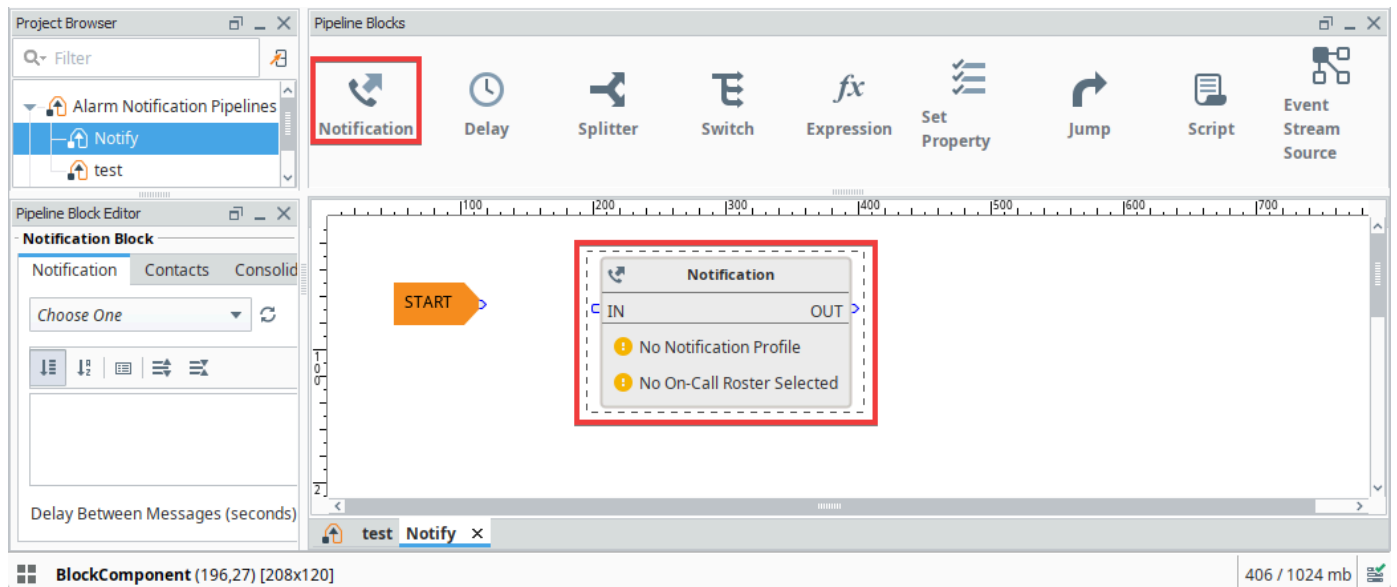


Every pipeline has a Start block which you will see when you create your pipeline. When an alarm is triggered, it automatically goes to the Start block, and from there you can use other pipeline blocks to determine what you want to do with that alarm. All the available pipeline blocks are shown at the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

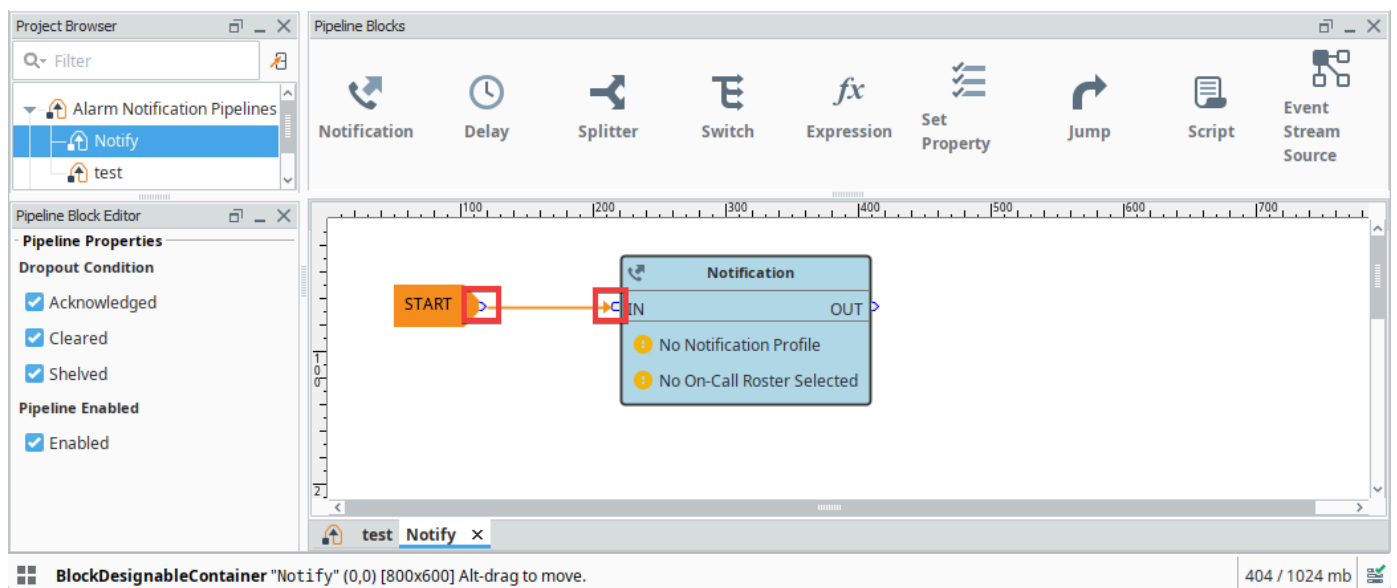


4. The most basic pipeline you can create is to simply notify someone right away when an alarm occurs. Drag the Notification block on to your canvas and place it next to the Start block. Note

that the Notification block is not automatically connected.



5. Drag the output pin from the Start block to the input pin of the Notification block to connect the two blocks. When an alarm occurs, it will go directly into the Start block and then into the Notification block.



6. On the Notification block, you need to specify the **Alarm Notification Profile** that you are going use and the **On-Call Roster** containing who you are going to send the alarm notification to. These are configured on your Gateway.
7. To specify which Alarm Notification Profile you want to use in your pipeline, click on the Notification block. The Notification block will open in the Pipeline Block Editor. Choose which Notification Profile you want to use. In this example, we will use **myEmailNotification**.

Pipeline Block Editor

Notification Block

Notification Contacts Consolidation

Choose One ▼ ↻

<None>

myEmailNotification

Delay Between Messages (seconds): 0

8. In the Pipeline Block Editor, under the **Contacts** tab, click the **On-Call Roster** dropdown. Select the roster you want to notify about this alarm. In this example, we will choose **myRoster**.

Pipeline Block Editor

Notification Block

Notification **Contacts** Consolidation

Roster Type

Direct ▼

On-Call Roster

<None> ▼ ↻

<None>

Managers

Operators

myRoster

You have 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 myRoster 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**. See how to make your alarm use a pipeline on the [Adding Pipelines to Tags](#) page.

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

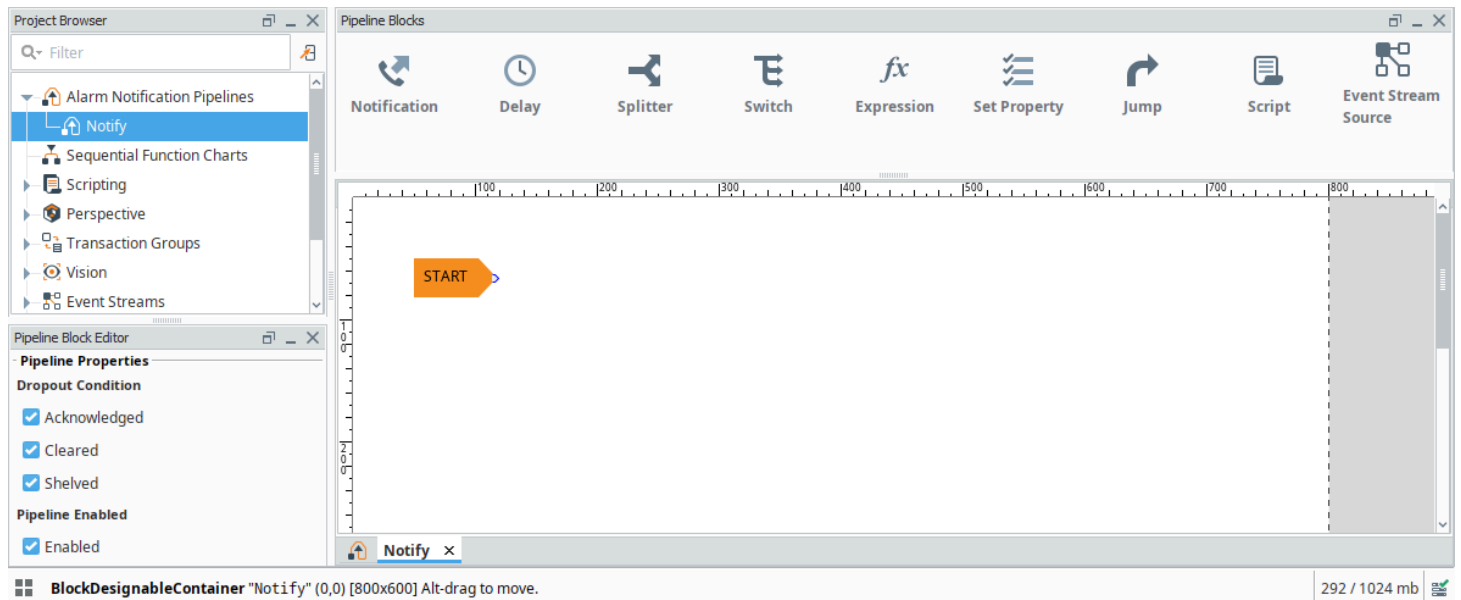
The **Pipeline Blocks** are described individually in this section, but here is a brief overview:

- **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.
- **Event Stream Source** Transmits alarm event data to an **event stream** with an event listener source.

Start Block

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

pipeline.



Notification Block

The **Notification block** is the block that controls the actual notification. When an alarm event reaches this block in the pipeline, some sort of notification will go out. In it, you can 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.

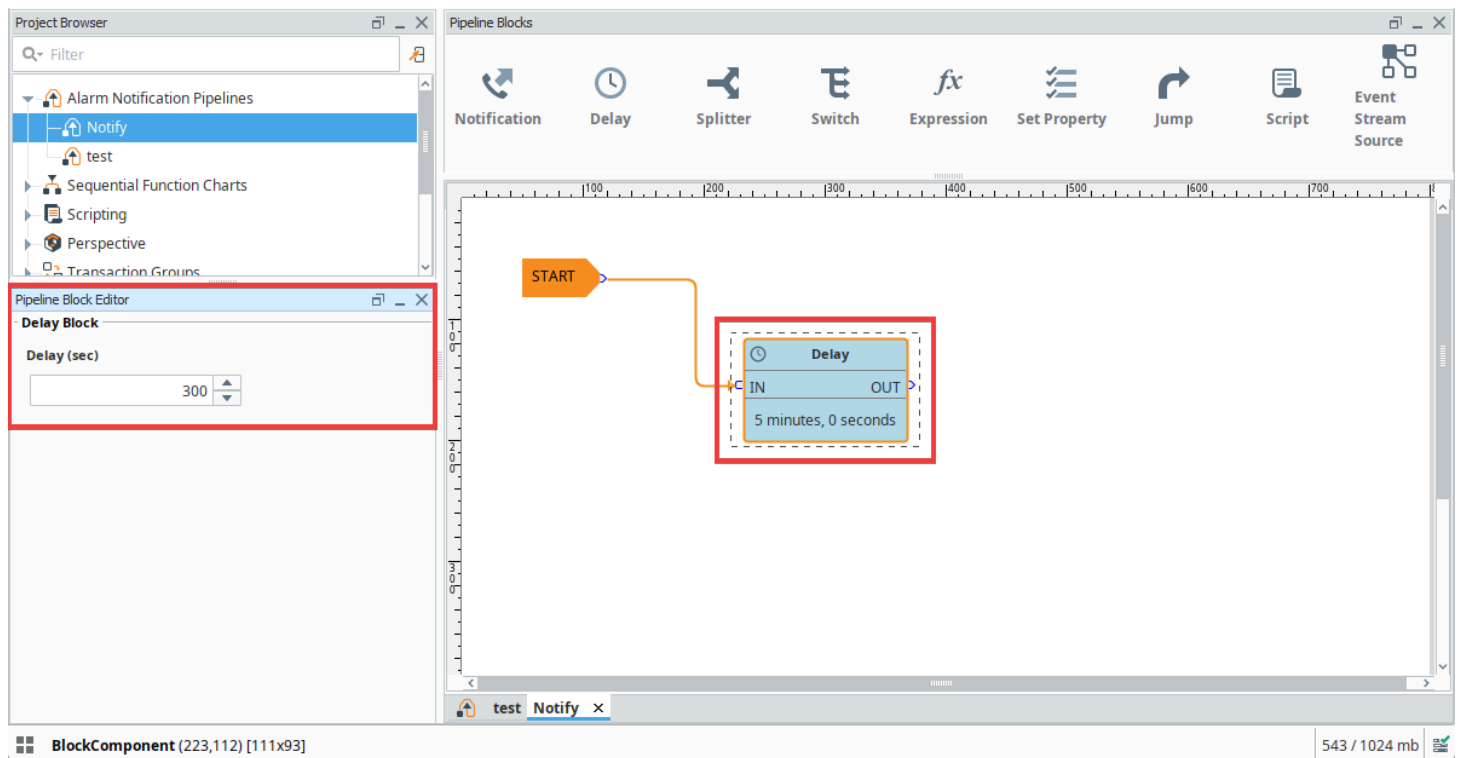
Delay Block

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).

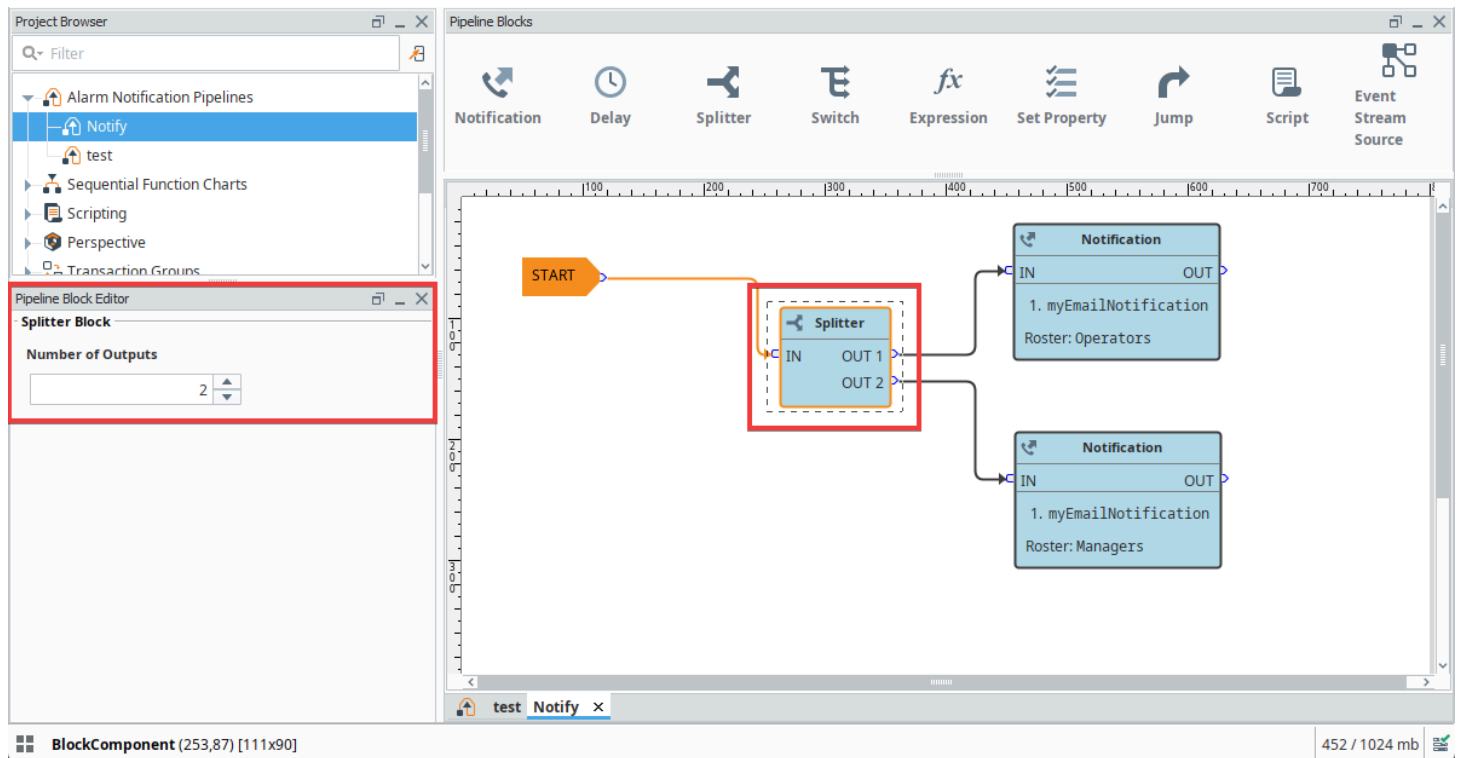
Delays are often also used to control flow in a pipeline. For example, in an "escalation" scenario, a notification block might be used to send emails. Since emails are sent instantly, and acknowledgment occurs "asynchronously" (the user must see the email and click the link or log into the system), a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

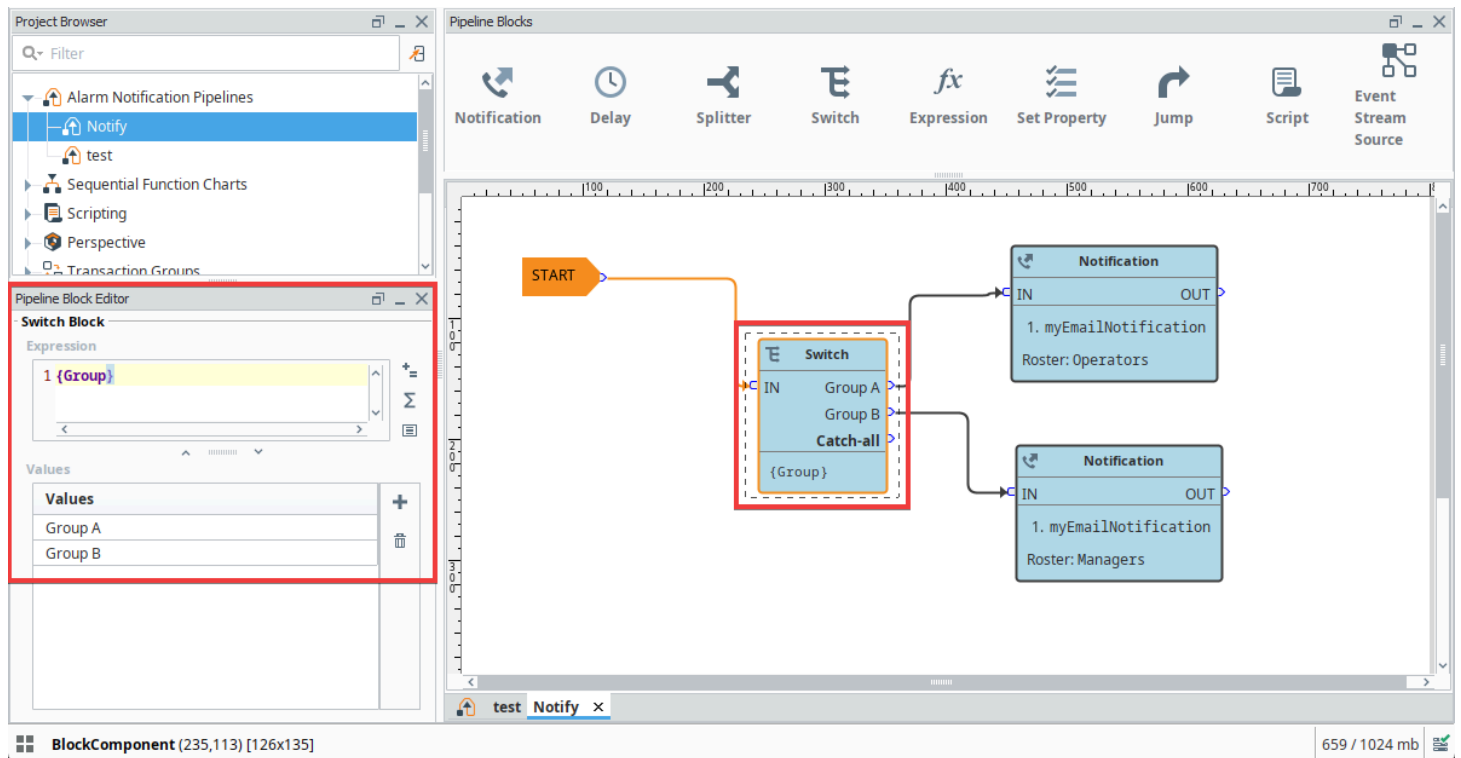


Switch Block

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.

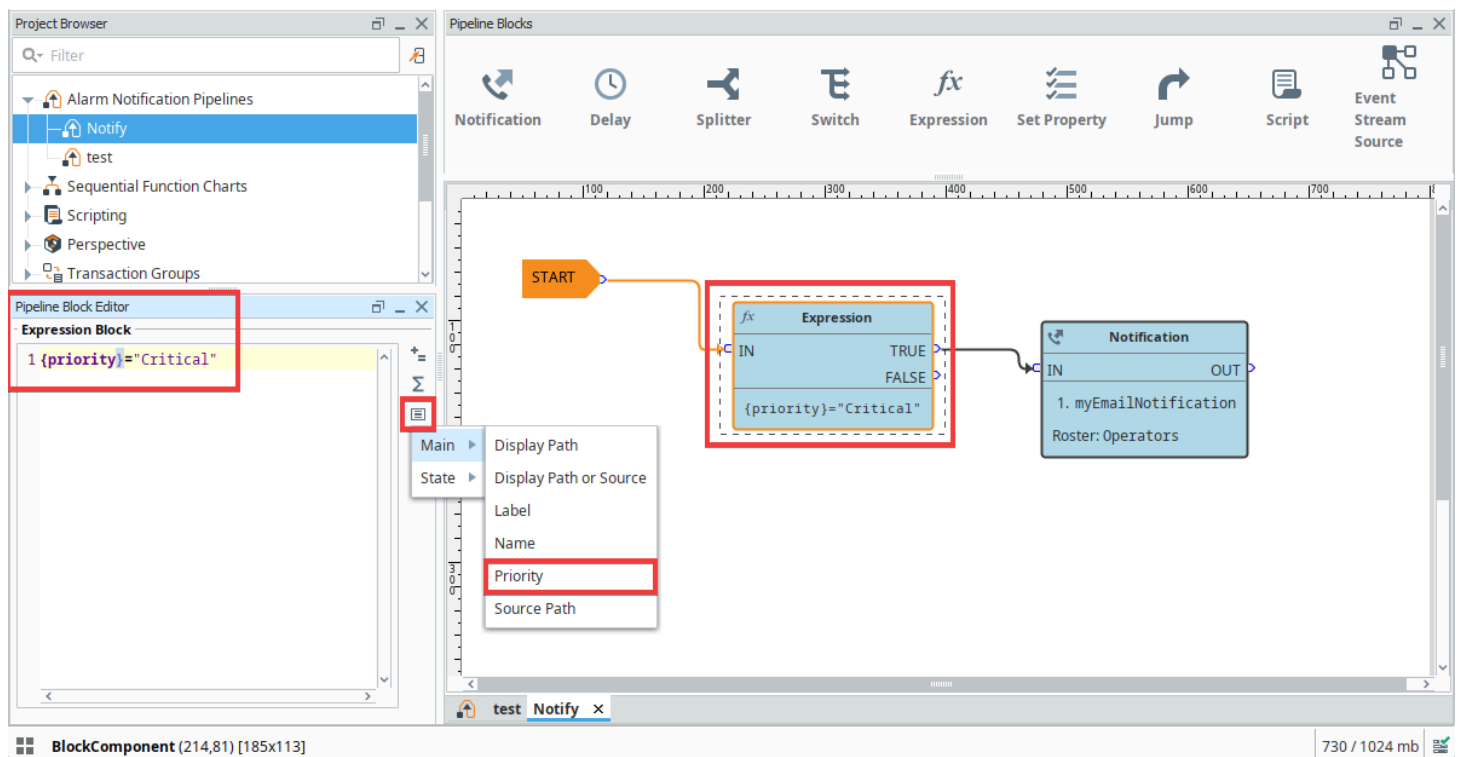
`isPropertyDefined("isEscalated")` This Expression checks if a property exists on the event. The `isEscalated` property is not a system defined property. Instead, in this example, it might be set using a Set Property block before forwarding back to the start of the pipeline. The first time through, this

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).

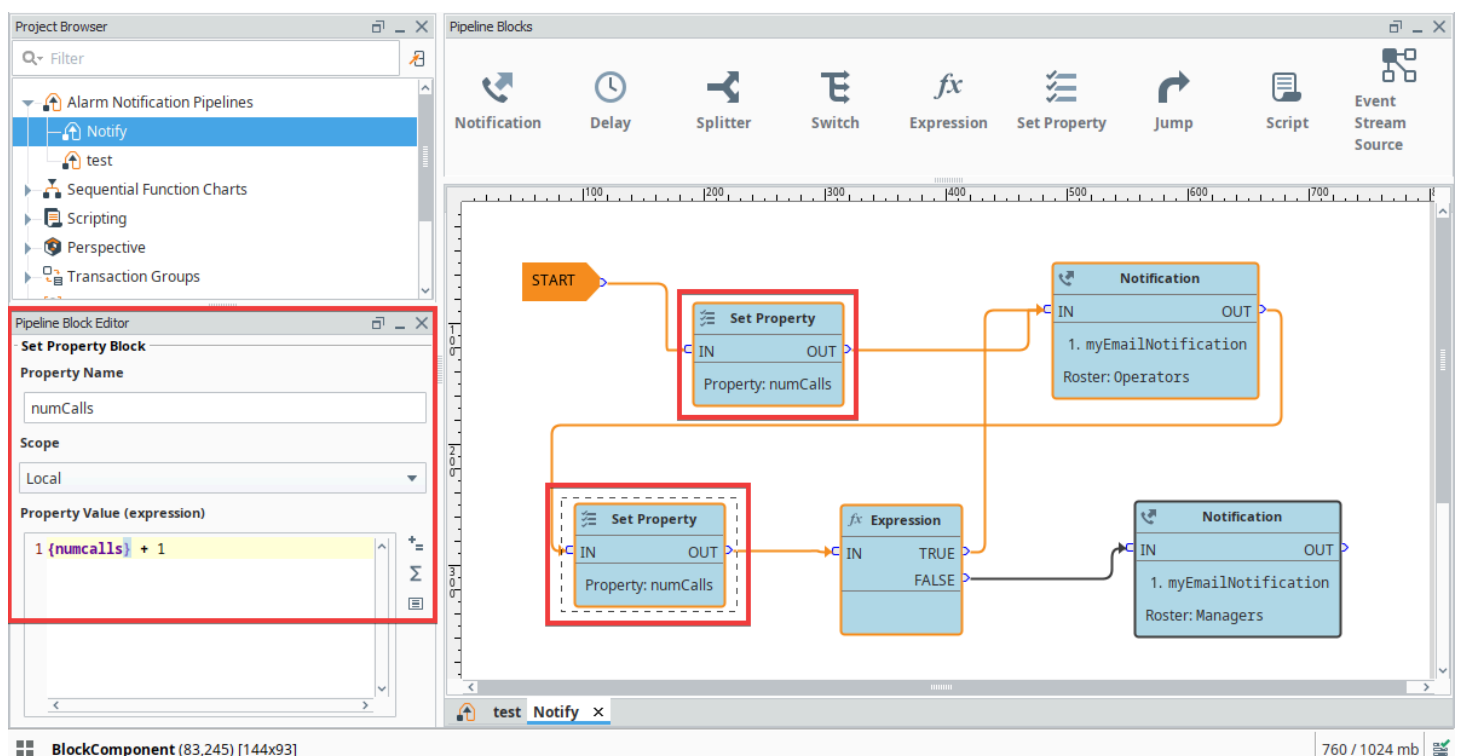
The Set Property block would look like: Property Name: counter Expression: `coalesce({counter}, 0) + 1`

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 Managers. 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 Managers on-call roster.

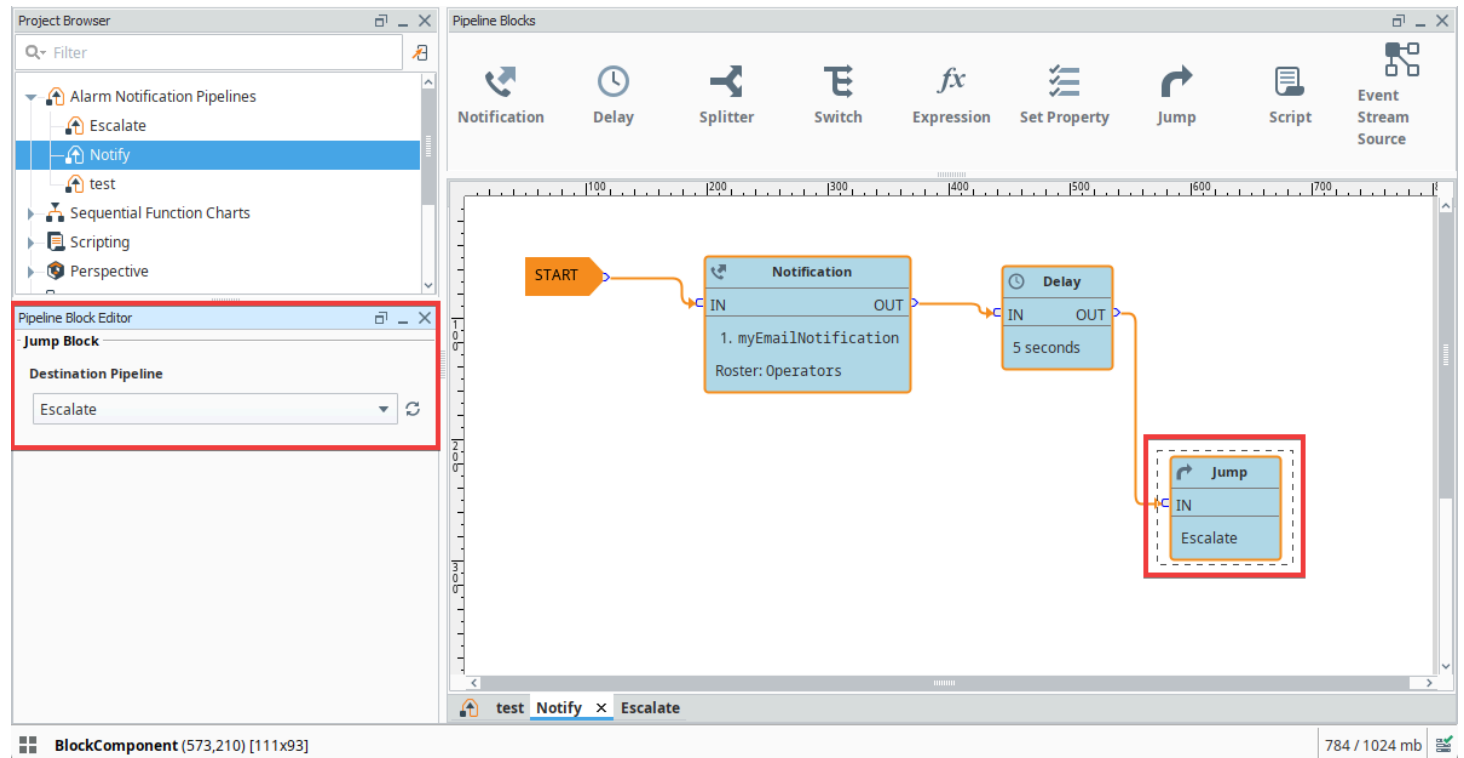


Jump Block

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**.

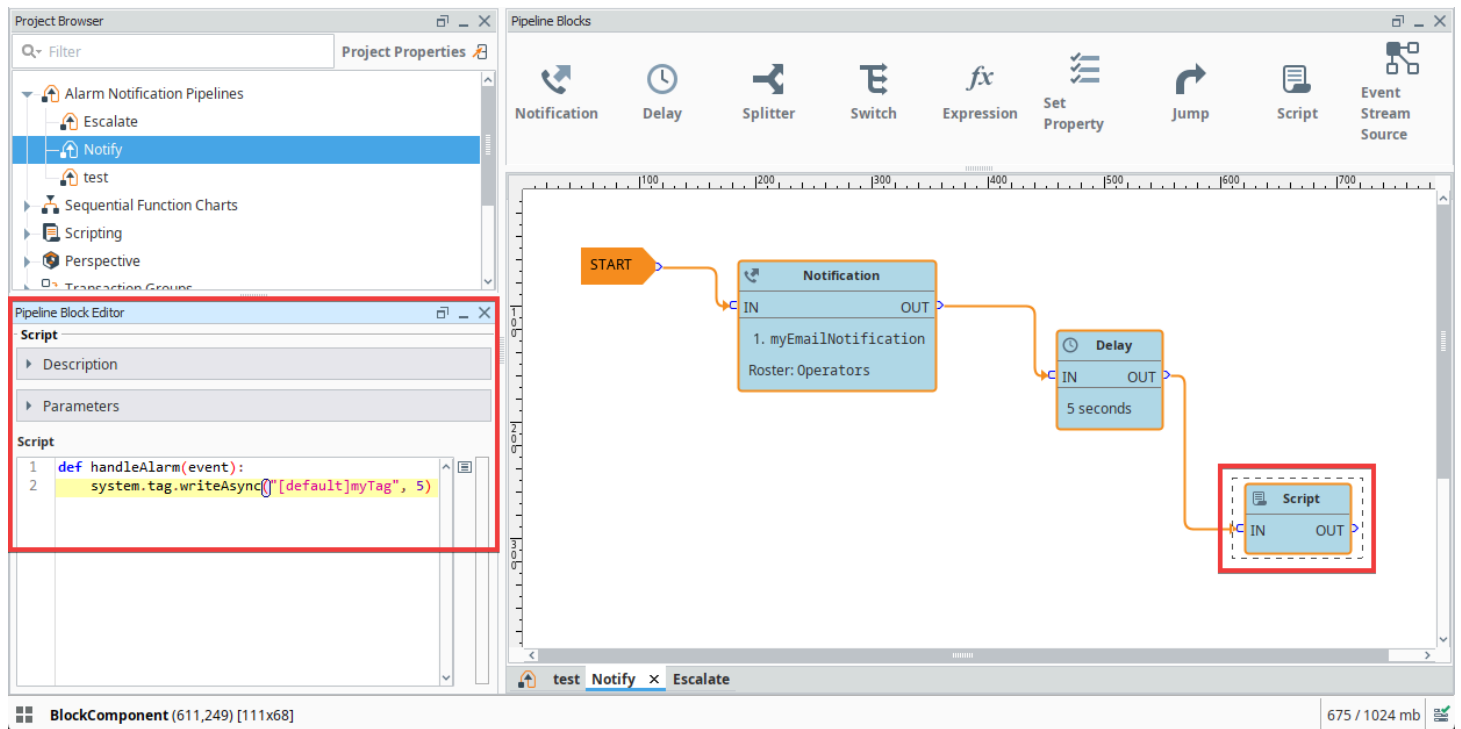
In this simple example, the Jump block forwards the alarm to the Escalate pipeline and ends there.



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.

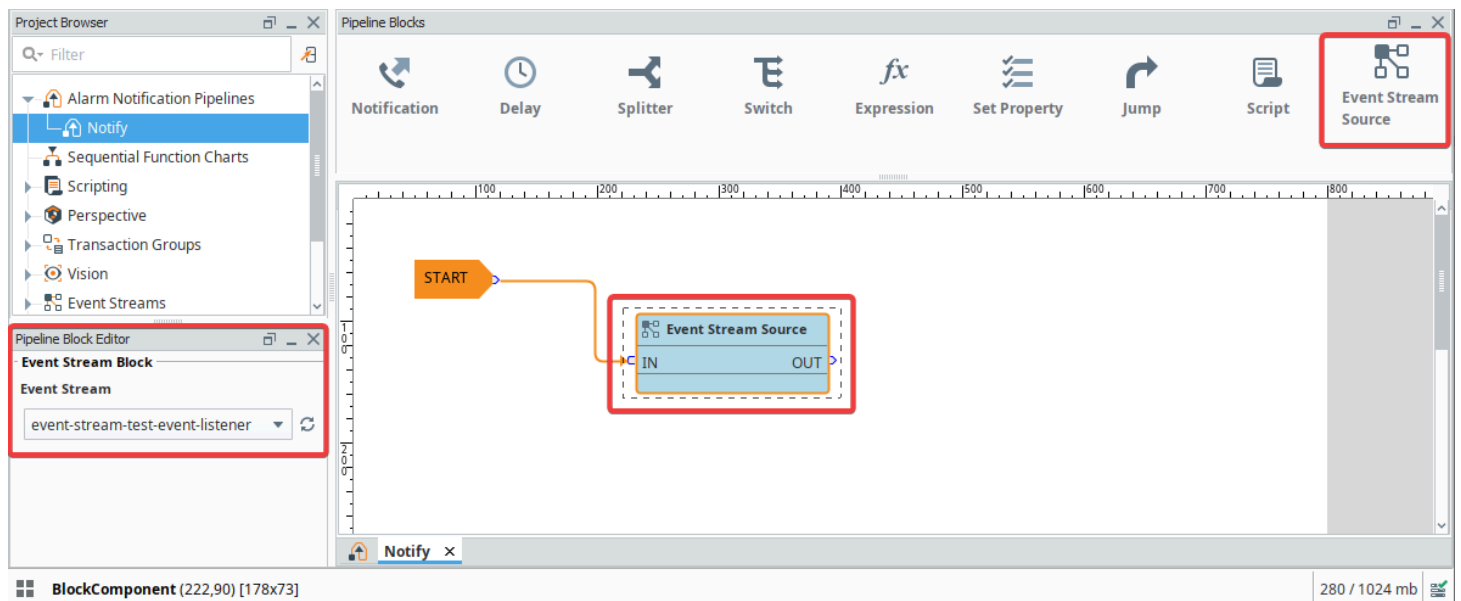
The script block provides access to an event object, which is simply a **ScriptableBlockPyAlarmEvent** object.



Event Stream Source Block

The **Event Stream Source Block** allows you to send alarm event data to an event stream that is using an event listener as a source. Then, based on handler setup, the event stream will manipulate the alarm data as desired.

The alarm event data from this pipeline block will be an `AlarmEventObject`, containing extractable metadata similar to other [event stream sources](#).



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.

Pipeline Block Editor

Notification Block

Notification

Contacts

Consolidation

<None>

Delay Between Messages (seconds): 0

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 <code><html></code> tag at the beginning of the Message property. An end tag (<code></html></code>) 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 section for more details.
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 acknowledgement 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.

Pipeline Block Editor

Notification Block

NotificationContactsConsolidation

1. Email 1

▼ Email 1

From Address	default@example.com	
Subject	Ignition Alarm Notification	↔
Message	At {eventTime hh:mm:ss}, ala	↔
Consolidated Message	{alarmEvents.Count} alarm ev	↔
Recipient Type	To	▼
Send Ack Link	True	▼
Test Mode	False	▼
Retry Count	0	
Retry Delay (sec)	5	

Delay Between Messages (seconds):

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. 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: Contact User1 (wait Delay Between Retries time) > Retry User1 (wait Delay Between Retries time) > Retry User1 (wait Delay Between Messages time) > Contact User2. 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. Caution: 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.

Pipeline Block Editor

Notification Block

Notification
Contacts
Consolidation

1. Voice

+

Voice

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

Delay Between Messages (seconds):
0

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 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 acknowledge code.
Test Mode	If enabled, logs messages that would have been sent to the Gateway console, and does not actually send them.

Pipeline Block Editor

Notification Block

Notification
Contacts
Consolidation

1. SMS

+

SMS

Message

At {eventTime | hh:mm:ss}, ala

Consolidated Message

{alarmEvents.Count} alarm ev

Acknowledge Message

To acknowledge, reply '%s'

Test Mode

No

Delay Between Messages (seconds):
0

Twilio Settings

Notifications can also be sent with the **Twilio Alarm Notification** module. To use Twilio, you must first set up one of the three available **Twilio Notification Profiles**. This does not require any hardware, but the Gateway must have Internet access, and a Twilio account with the required capabilities must be created (www.twilio.com).

SMS Notification Settings

Setting	Description
From Number	Number from which the alarm notification is sent.
Message	Message to send. Default is <code>At {eventTime hh:mm:ss}, alarm {name} at {displayPath} transitioned to {eventState}</code> . Click the binding  icon to access other alarm properties that can be used for the message.
Consolidated Message	The message sent when consolidation is turned on, and more than one alarm is being sent. See the Notification Block section for more details.

Pipeline Block Editor

Notification Block

Notification

Contacts

Consolidation

1. Twilio_Notification_Profile

+

↓

↓₂

☰

↕

↔

▼ Twilio_Notification_Profile

From Number

Message

Consolidated Message

At {eventTime|hh:mm:ss}, ala

{alarmEvents.Count} alarm ev

Delay Between Messages (seconds):

43

▲

▼

Voice Notification Settings

Setting	Description
From Number	Number from which the alarm notification is sent.
Max PIN Attempts	The maximum number of attempts allowed for a user to correctly enter a PIN when pin validation is enabled.
Call Script	The list of messages used by the text-to-speech engine to generate the call message. See the Call Script Options section of the Twilio Voice Notification page for more details.

Pipeline Block Editor

- Notification Block

Notification

Contacts

Consolidation

1. Twilio_Voice

▼ Twilio_Voice

From Number	
Max PIN Attempts	3

▼ Call Script

Greeting	Hello, this is an Ignition alarm ...
PIN Challenge	Please enter your pin, followe...
Non-PIN Challenge	Please press any key to contin...
Invalid PIN	Invalid pin. Please re-enter yo...
PIN Challenge Failed	PIN Entry failed. Goodbye.
Event Summary	There are %d events.

Delay Between Messages (seconds):

WhatsApp Notification Settings

Setting	Description
From Number	The number associated with the Messaging Service used to send WhatsApp messages.
Message	Message to send. This content will serve as the free-form message. Click the binding  icon to access other alarm properties that can be used for the message.
Consolidated Message	The message sent when consolidation is turned on, and more than one alarm is being sent. See the Notification Block section for more details.
Template	The Template ID of your pre-approved Templates, found in the Twilio Console under Develop > Messaging > Content Template Builder. On Twilio, it is advised to use Utility Templates over Marketing Templates.
Template Parameters	Optional parameters to set your template variable values. Each value is keyed by an integer value, which increments per parameter value (i.e., <code>{{1}}</code> , <code>{{2}}</code> , <code>{{3}}</code>).

Notification Block

Notification

Contacts

Consolidation

1. test_WhatsApp

▼ test_WhatsApp

From Number

Message

Consolidated Message

Template

Template Parameters

Third Message {alarmEvents.Count}

{alarmEvents.Count} alarm events have occurre

HX1ac52aed6284e5a48f7579

User From Designer; {name}

Delay Between Messages (seconds):

0

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

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

Contacts

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.



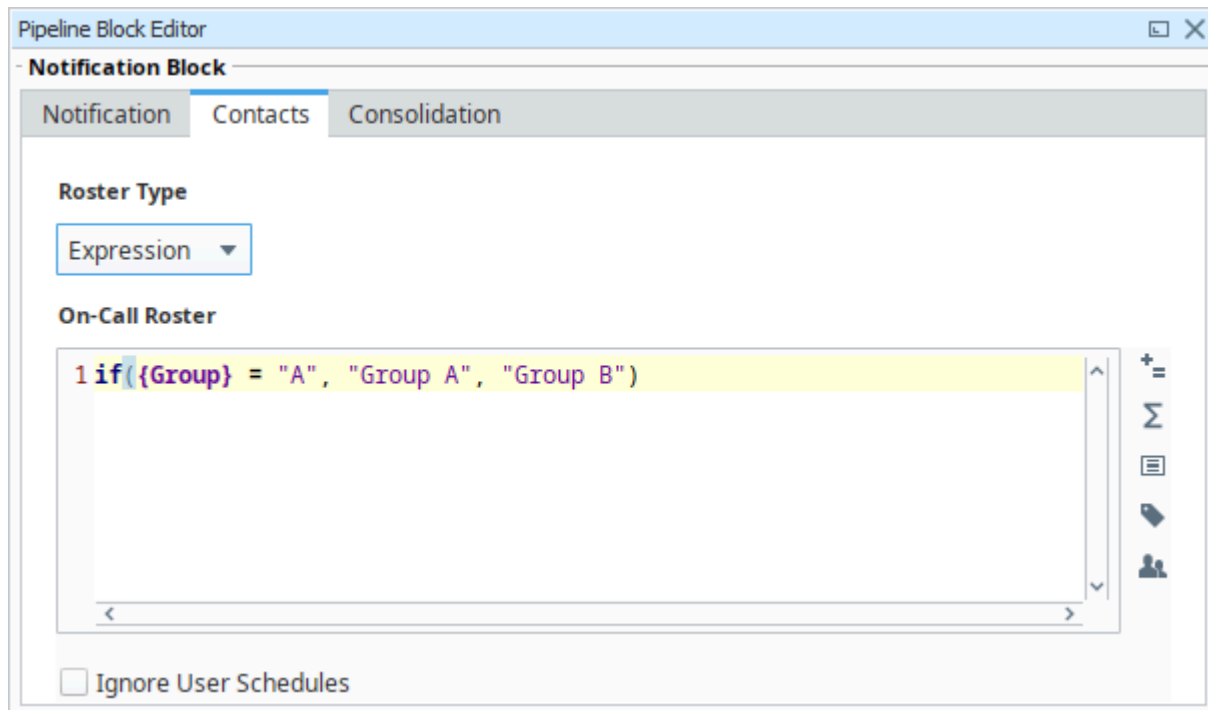
Direct

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.

A screenshot of the 'Pipeline Block Editor' window. The title bar says 'Pipeline Block Editor' with standard window controls. Inside, there's a tabbed interface with 'Notification Block' selected. Below the tabs are three sub-tabs: 'Notification', 'Contacts', and 'Consolidation'. The 'Roster Type:' dropdown is set to 'Direct'. Below that, the 'On-Call Roster' dropdown is open, showing a list of options: '<None>', 'Managers', 'Operators', 'Test', 'User Group 1', and 'User Group 2'. A refresh icon is visible to the right of the dropdown.

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.

Using 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")
```

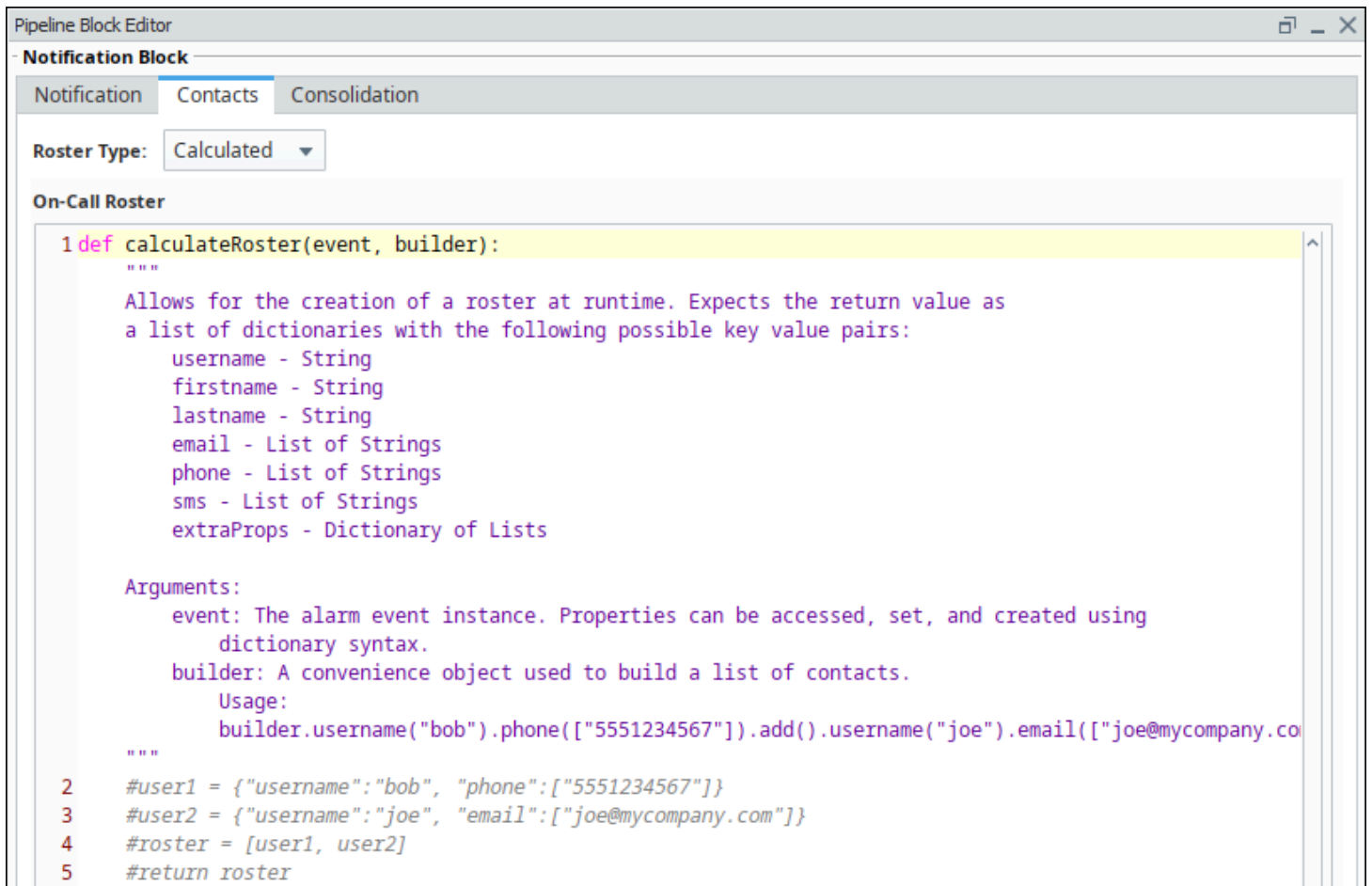
However, if the quotes were removed from the roster names, the expression would fail with a syntax error:

```
//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 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.



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.

Alarm Properties

```
# 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
username	A username for this particular user. Note: This can be anything that you want, the user does not have to exist in the system. It is used for acknowledgement and auditing purposes, so it is important you create a unique username for each user.
firstname	A first name for this particular user.
lastname	A last name for this particular user.
email	A list of possible email address for the user.
phone	A list of possible phone numbers for the user for voice notification purposes.
sms	A list of possible SMS numbers for the user for SMS notification purposes.
whatsapp	A list of possible numbers for the user for WhatsApp notification purposes.
extraProps	Extra properties that can be added to a user. This is useful when dealing with custom functionality. The extraProps key can be used to set arbitrary user properties. This is useful for setting a property not mentioned above. For example, if you needed to set the language of the calculated roster to Spanish so that the Voice Notification module uses a Spanish voice agent, you can be configured as follows: <pre>user1 = {"username":"bob", "phone":["5550100"], "extraProps":{"language":"Spanish"}}</pre> The <code>extraProps</code> key can be used to set the following user properties. Technically, you can set any user property. See the User object. The value for each key must be a list. • badge • language • schedule

Name	Description
	In addition, <code>extraProps</code> can be used to set user properties that are registered in the system. For example, the SMS module registers a user pin setting, which can be set with the following code: <pre>user1 = {"username":"bob", "phone":["5550100"], "extraProps":{"u</pre>

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

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

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

# return the user data
return userList
```

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

User Data Created with the Builder

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

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

Looping through Users

```
# This would grab a table from the database that contains a username, email address, and sms number.
# We must specify the database connection name because this is in the gateway scope.
data = system.db.runPrepQuery(
    "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.runPrepQuery(
        "SELECT username, emailaddress, smsnumber FROM user_info_table",
        [],
        "DB"
    )

    # loop through the returned query data
    for row in data:
        # add each user to the builder

builder.username(row["username"]).email([row["emailaddress"]]).sms([row["smsnumber"]])

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

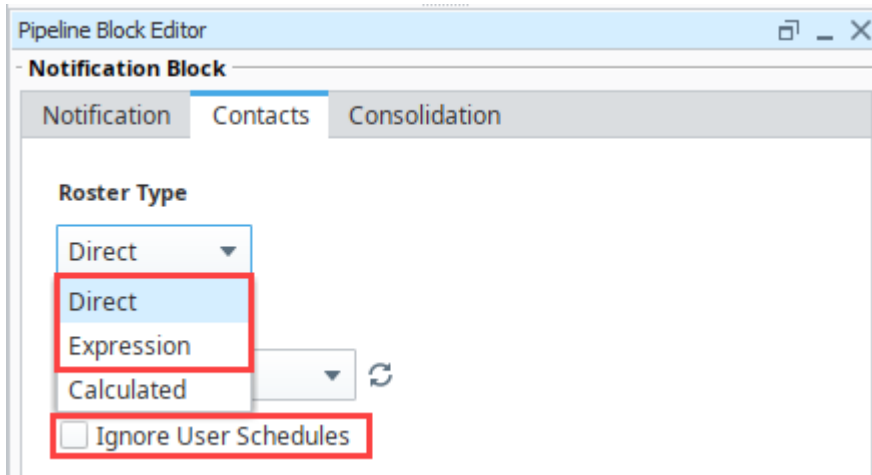
```
if({Group}= "A","Group A","Group B")
```

Ignore User Schedules

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.



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.

Pipeline Block Editor

Notification Block

Notification Contacts Consolidation

☒ Consolidation Enabled

Delay (sec)

15

Frequency (sec)

60

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.

Pipeline Block Editor

Notification Block

Notification Contacts Consolidation

1. Papercut

Consolidated Message

← Back

```
1 {alarmEvents.Count} alarm events have occurred:
2 {{Alarm "{name}" at "{displayPath}" transitioned to {eventState} at {eventTime|hh:mm:ss}}}
```

Delay Between Messages (seconds): 0

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 `<b>` 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:

 PAPER CUT

LogRulesOptionsExit

Ignition Alarm Notific.
10/13/2020 3:33:10 PM (831)

Fromdefault@example.com

Tostuff@thing.com

Date10/13/2020 3:33:10 PM -07:00

SubjectIgnition Alarm Notification

MessageHeadersBodySectionsRaw

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

FORWARDDELETE (1)

Papercut v5.1.76
<https://github.com/ChangemakerStudios/Papercut>

Pipeline - Filter on Alarm Priority

NOTE

This page assumes that you already know how to [configure alarms](#), set up [alarm notifications](#), and create [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**.



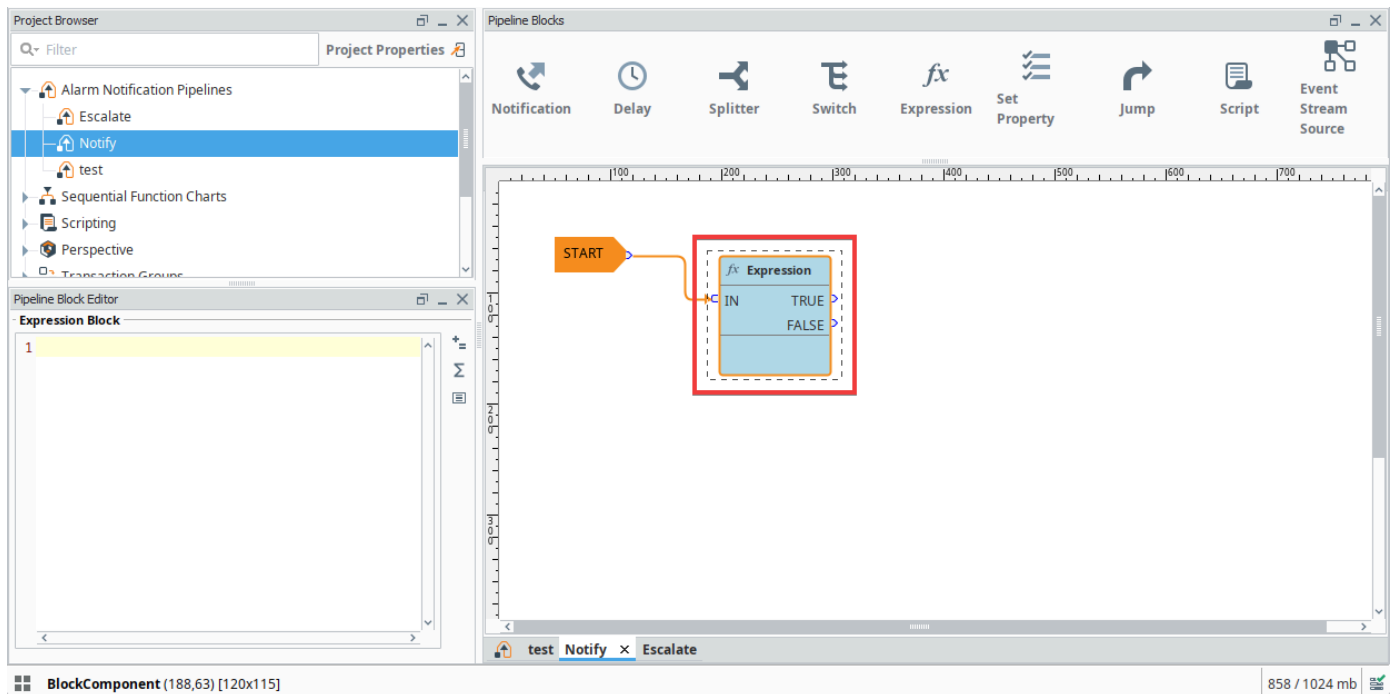
INDUCTIVE
UNIVERSITY

Pipeline - Filter on Alarm Priority

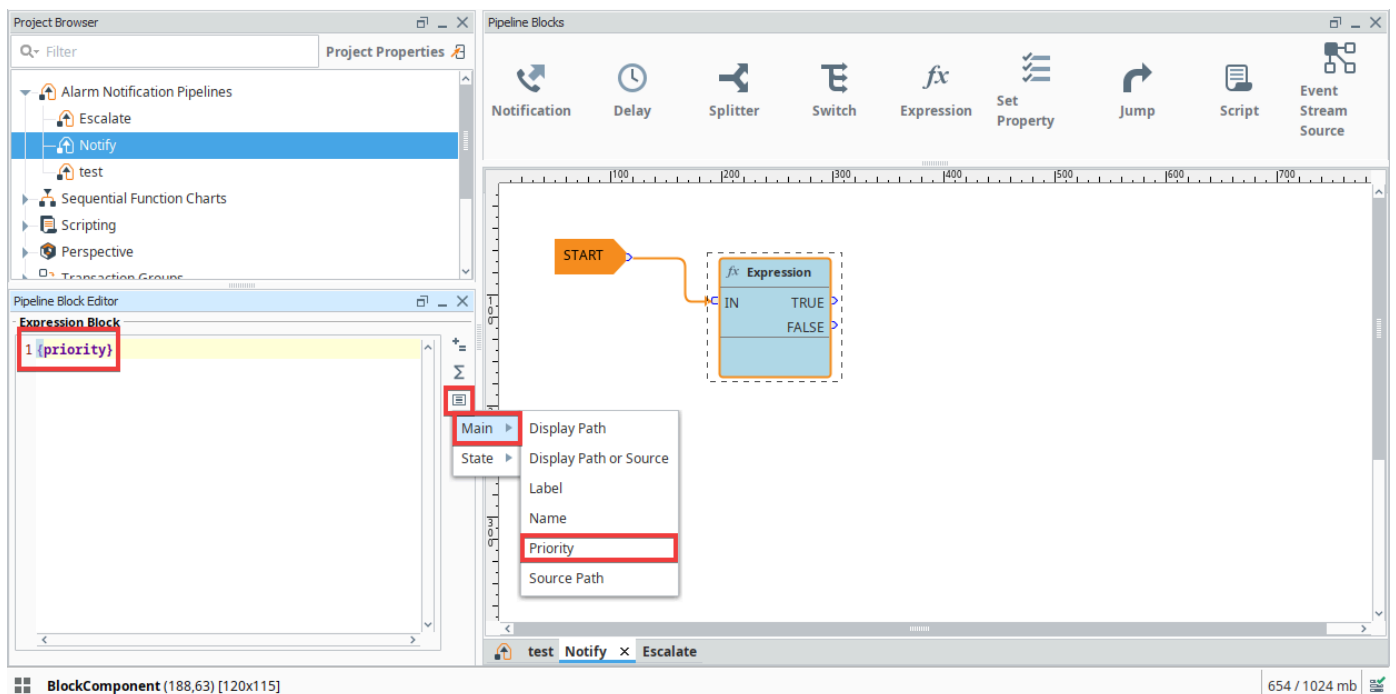
[Watch the video](#)

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 Block Editor, then click on 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.

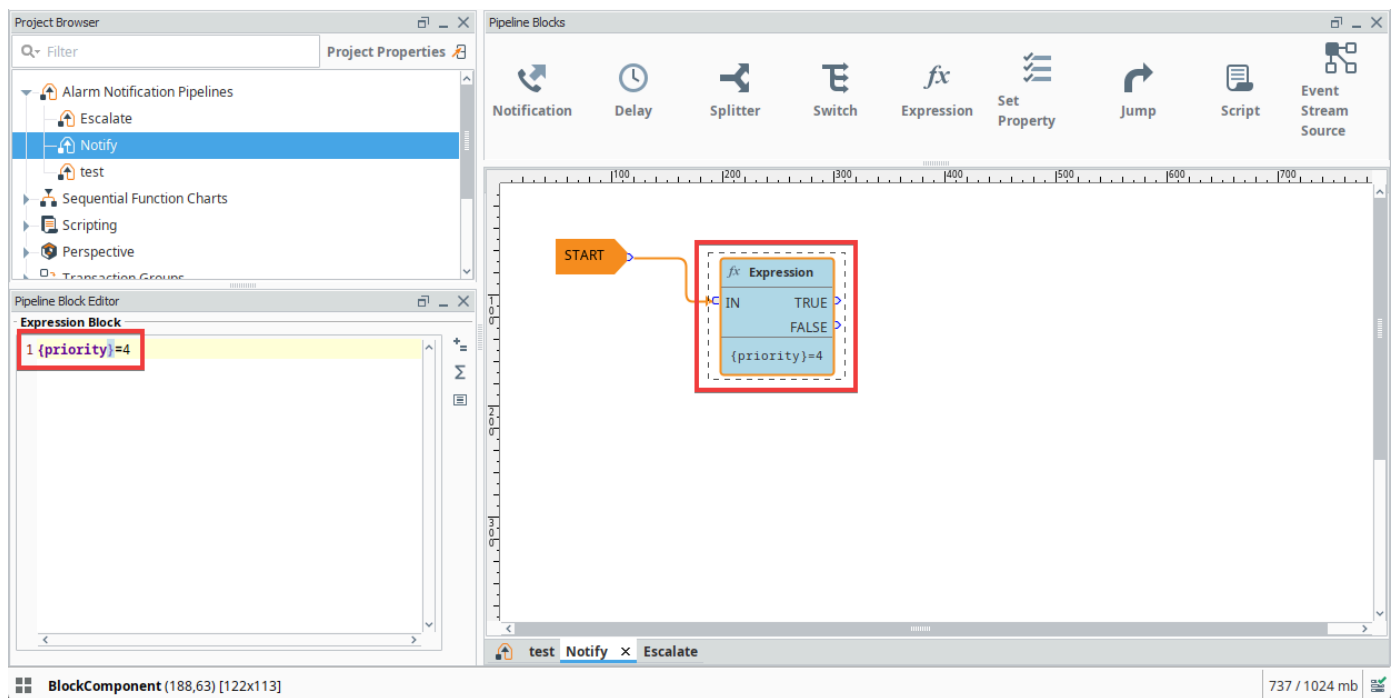


5. Alarm priorities have predefined integer values for filtering and sorting. Because this value is an int, you can test against it using =, <, >, >=, or <=. In this example, we are going to filter for all alarms with a priority of **Critical**. Critical has a value of 4.

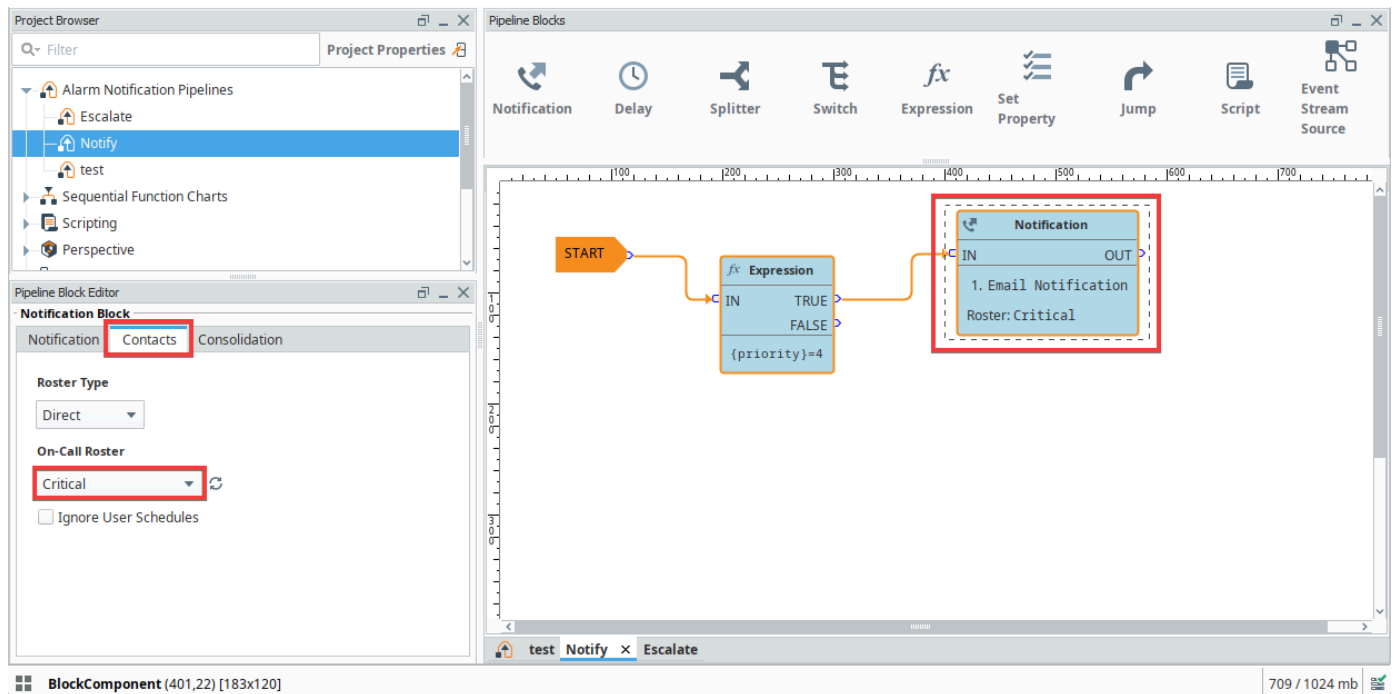
- 0 = Diagnostic
- 1 = Low
- 2 = Medium
- 3 = High
- 4 = Critical

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

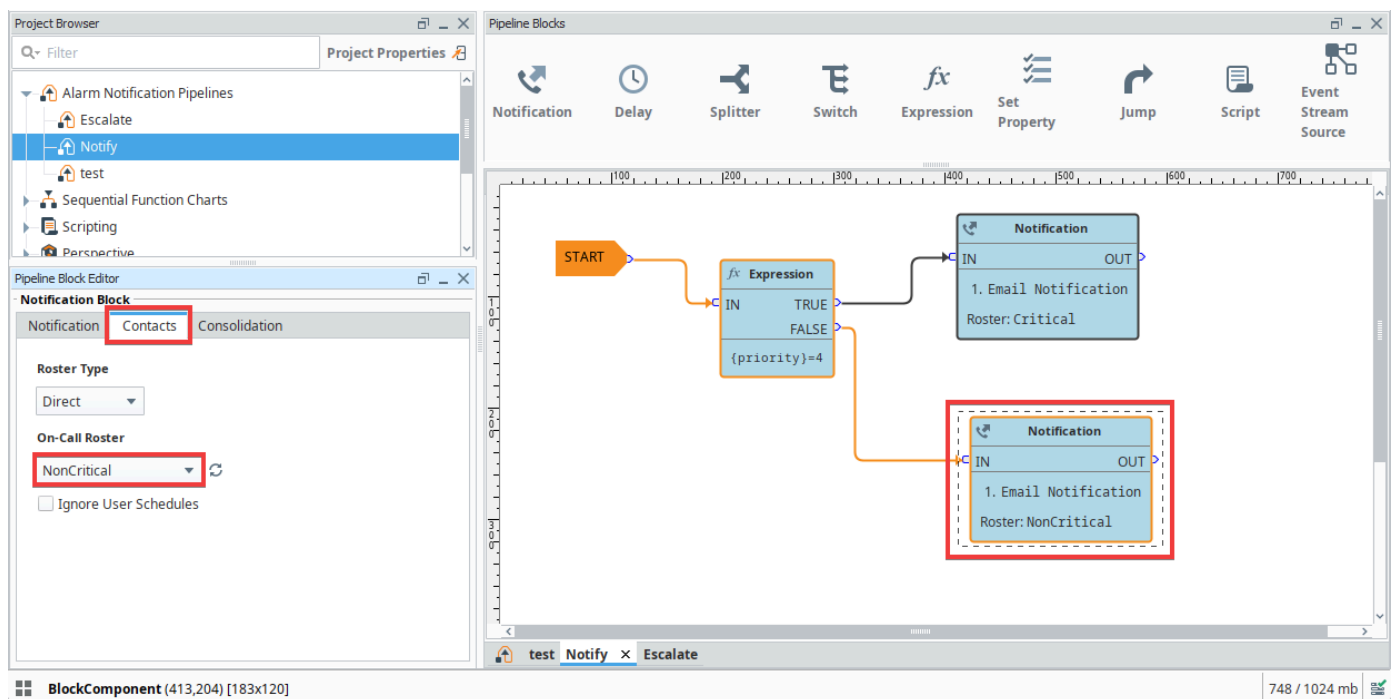
If the alarm priority is critical, the alarm will go to the output pin of the **True** value in the Expression block. If the priority is not critical, the alarm will go to the output pin of the **False** value. Notice, we are doing something different for each of these conditions.



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



9. 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 NonCritical list.



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

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.



INDUCTIVE
UNIVERSITY

**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.

Tag Editor

WriteableFloat1 > Alarms

default

Alarms

Fault1 - Above Setpoint, High

High Alarm - Above Setpoint, Critical

Low Alarm - Below Setpoint, Medium

High Alarm

Properties

2

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	

Active Pipeline

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

Commit

Revert

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.

Tag Editor

WriteableFloat1 > Alarms

default

Alarms

Fault1 - Above Setpoint, High

High Alarm - Above Setpoint, Critical

Low Alarm - Below Setpoint, Medium

High Alarm

Properties

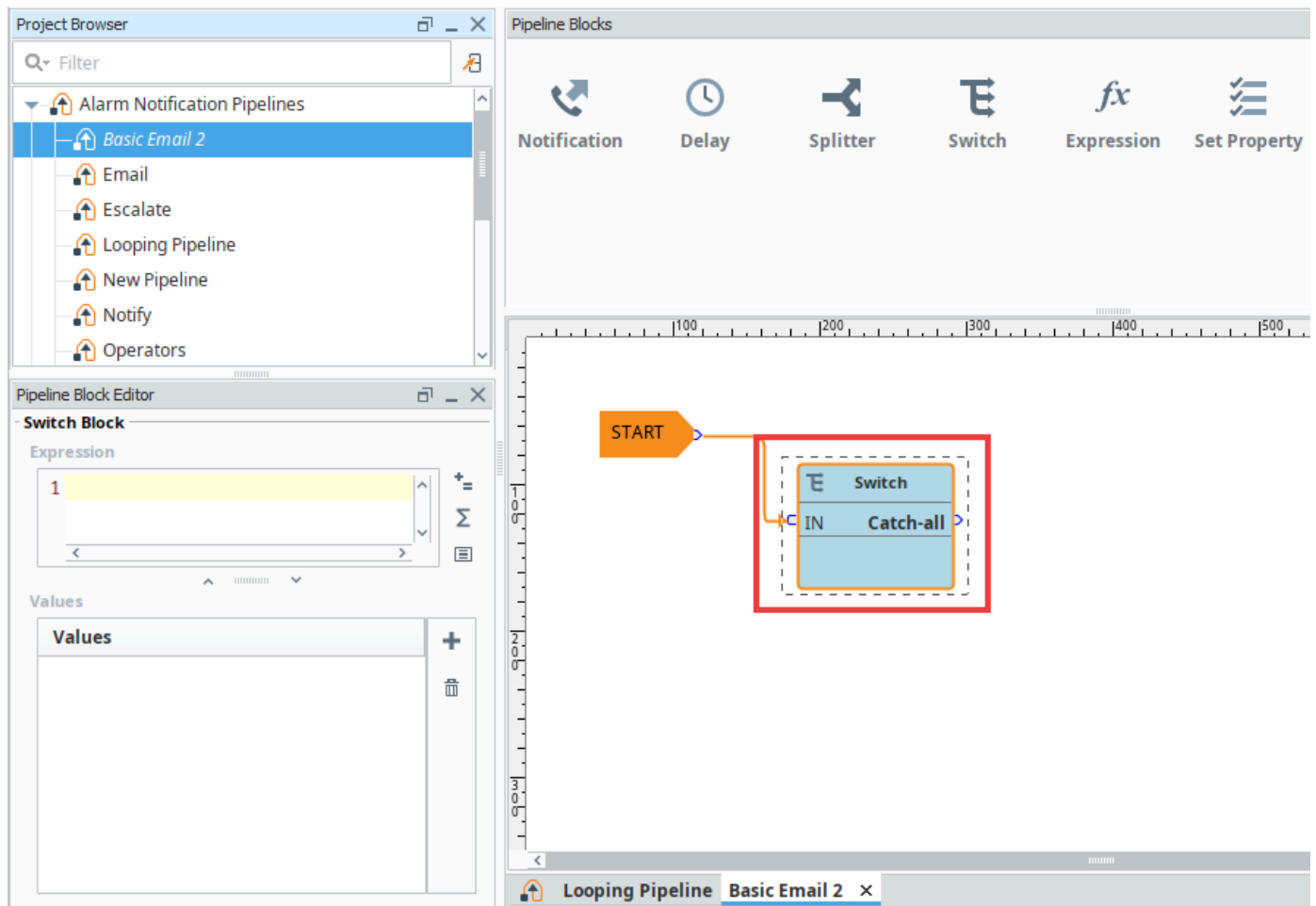
Inclusive	true	
Any Change	false	
Deadbands and Time Delays		
Deadband	0	
Deadband Mode	Absolute	
Active delay (seconds)	0	
Clear delay (seconds)	0	
Notification		
Ack Pipeline		
Active Pipeline	Examples_2019_11_and_Later/Email	
Clear Pipeline		
Phone Call Settings		
Custom Message		
Email Notification Properties		
Custom Message		
Custom Subject		
SMS Notification Properties		
Custom Message		
Associated Data		
Group	Group A	

Group

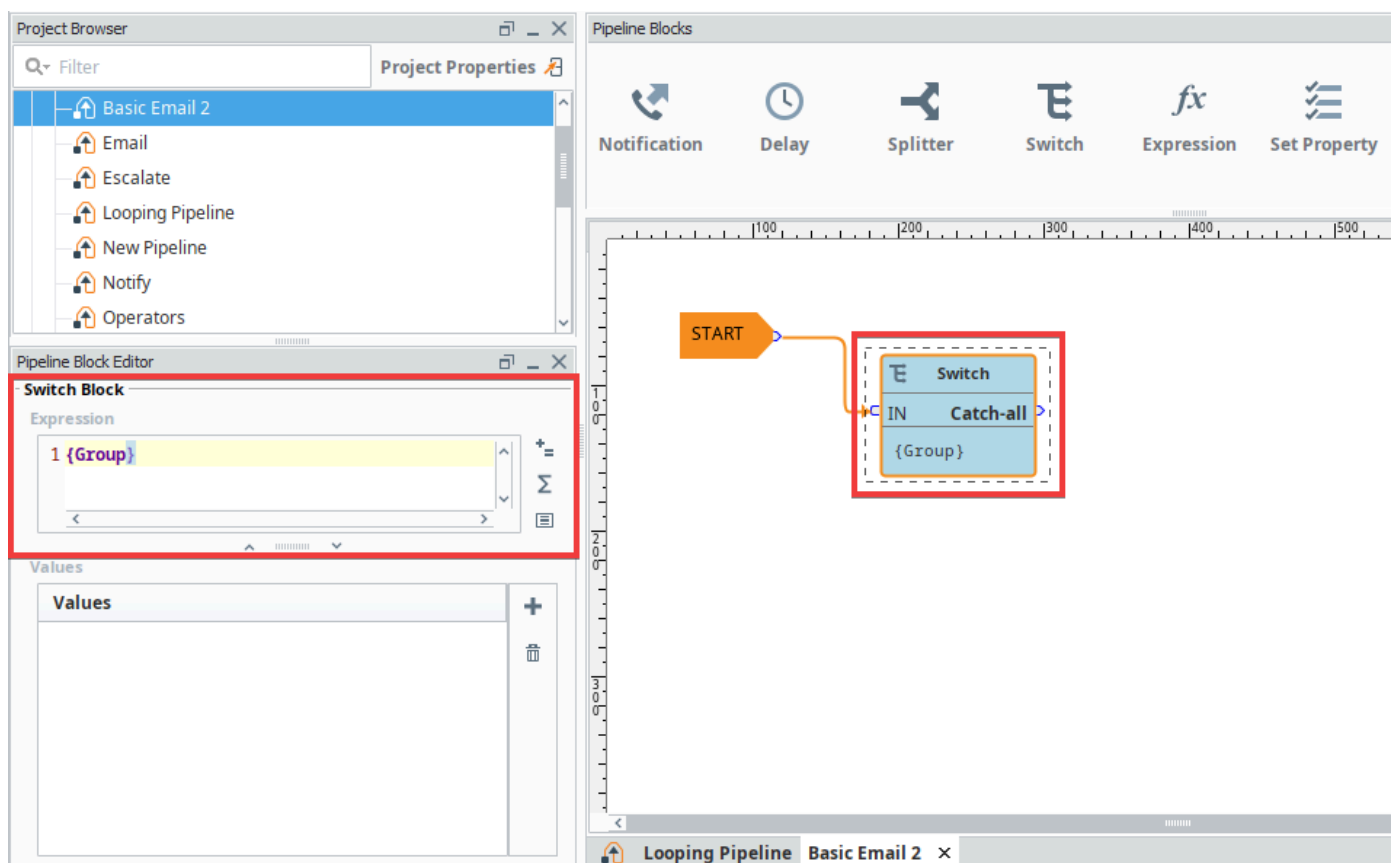
Commit

Revert

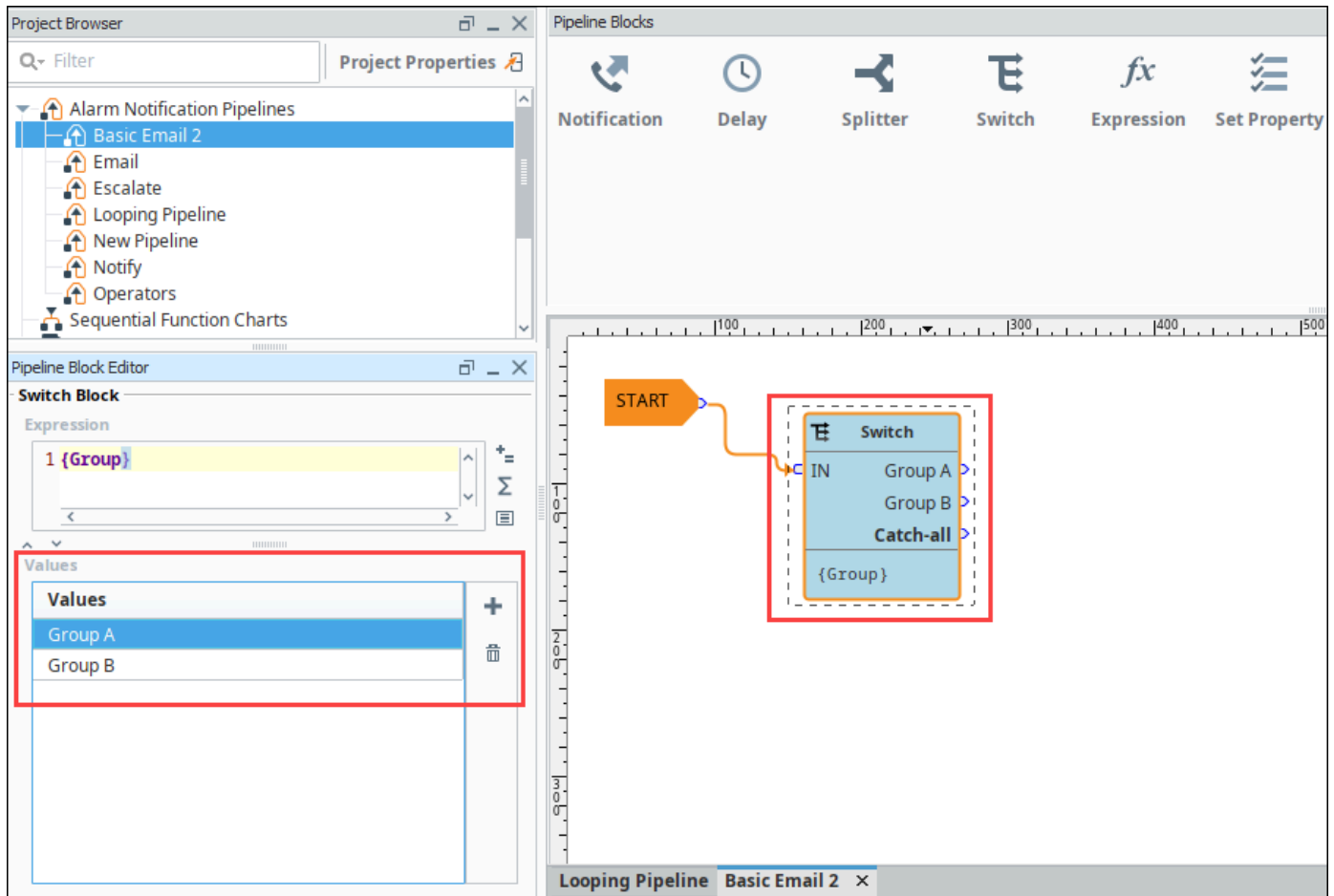
- For this example, drag in the **Switch block** to your pipeline, and connect the **Start block** to the **IN** of the Switch block. There are two blocks that can be used for filtering, either the Switch block or the Expression block. The Switch block allows you to switch on a value and do something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



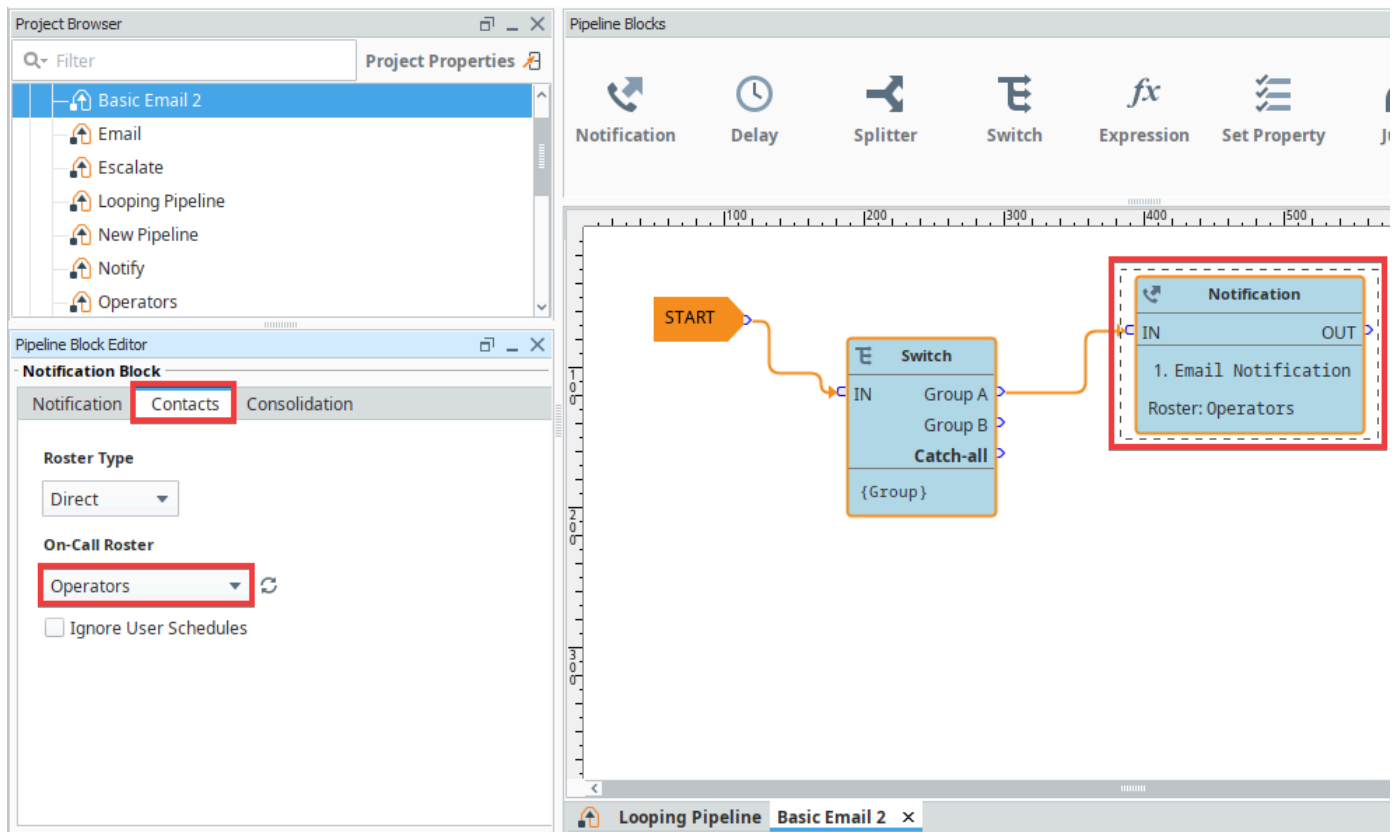
6. In the **Pipeline Property Editor**, you have to manually enter the expression because you will not find any of the associated data properties in the dropdown list. Enter the name of the associated data property in curly braces, such as `{Group}`. The **Group** associated data property acts as a placeholder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



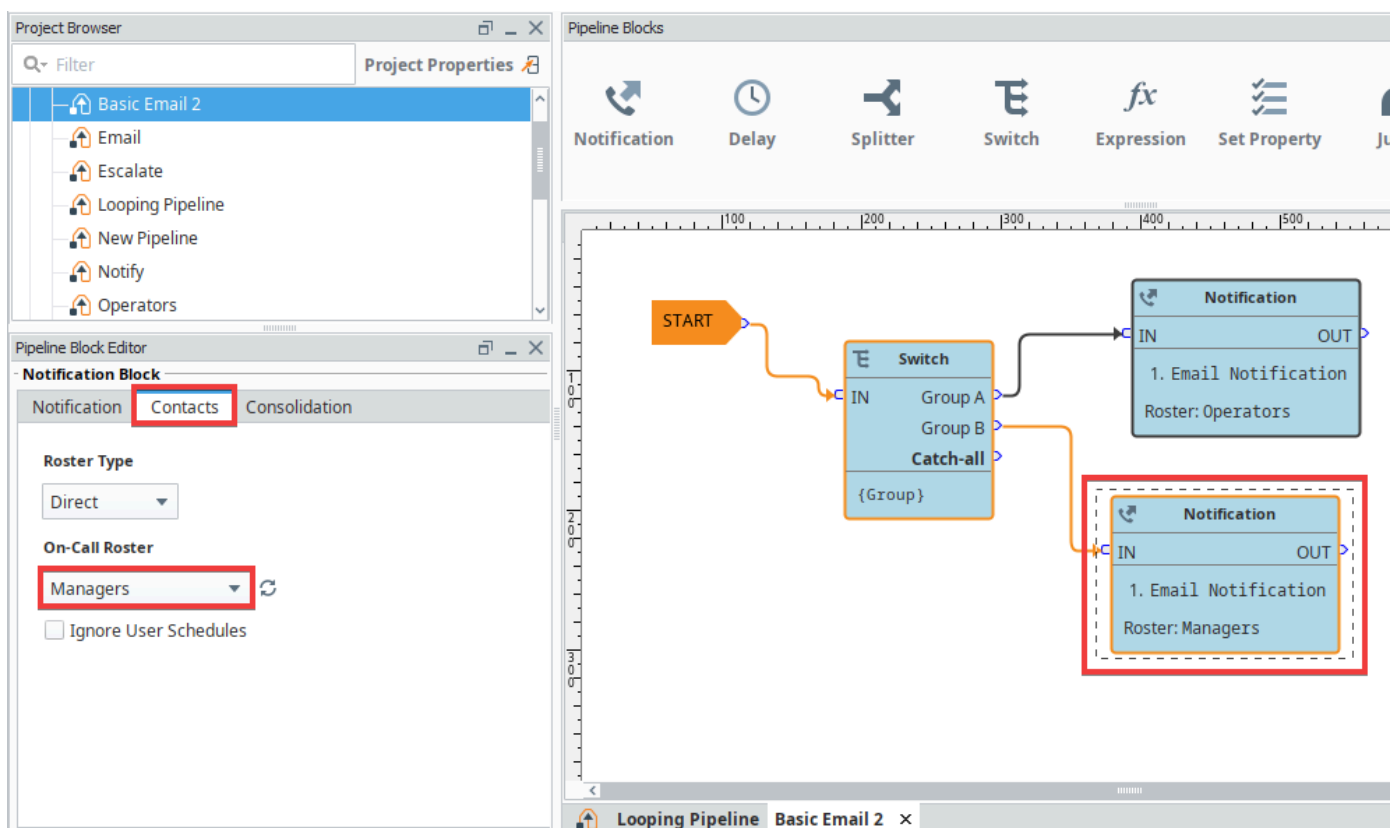
7. Next, you need to add some values to the expression. You can add as many values that you have values for. Click on the **Add +** icon and add two values: **Group A** and **Group B**. Once you enter both values, you will have outputs in the Switch block for **Group A**, **Group B**, and by default,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



8. 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.
9. In the Pipeline Block Editor, under the Notification tab, choose **Email Notification** as your notification type.
10. 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 Notification** as your notification type, and specify a different On-call Roster to notify a different group of people. We will be using the Managers roster in this example.



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

Pipeline - Simple Loop

THIS SECTION ASSUMES THAT ALARMS ARE CONFIGURED

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

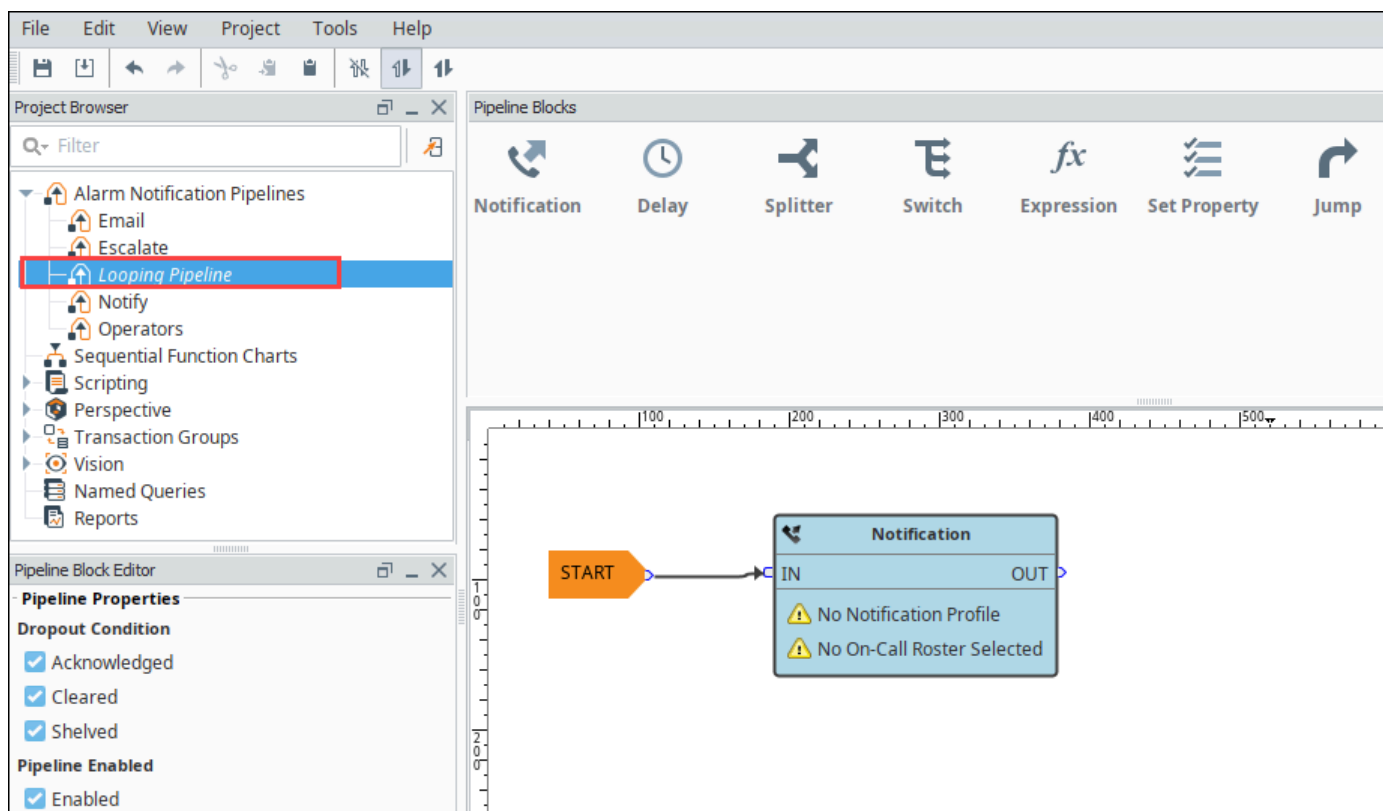
How to Create a Looping Alarm Notification

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.

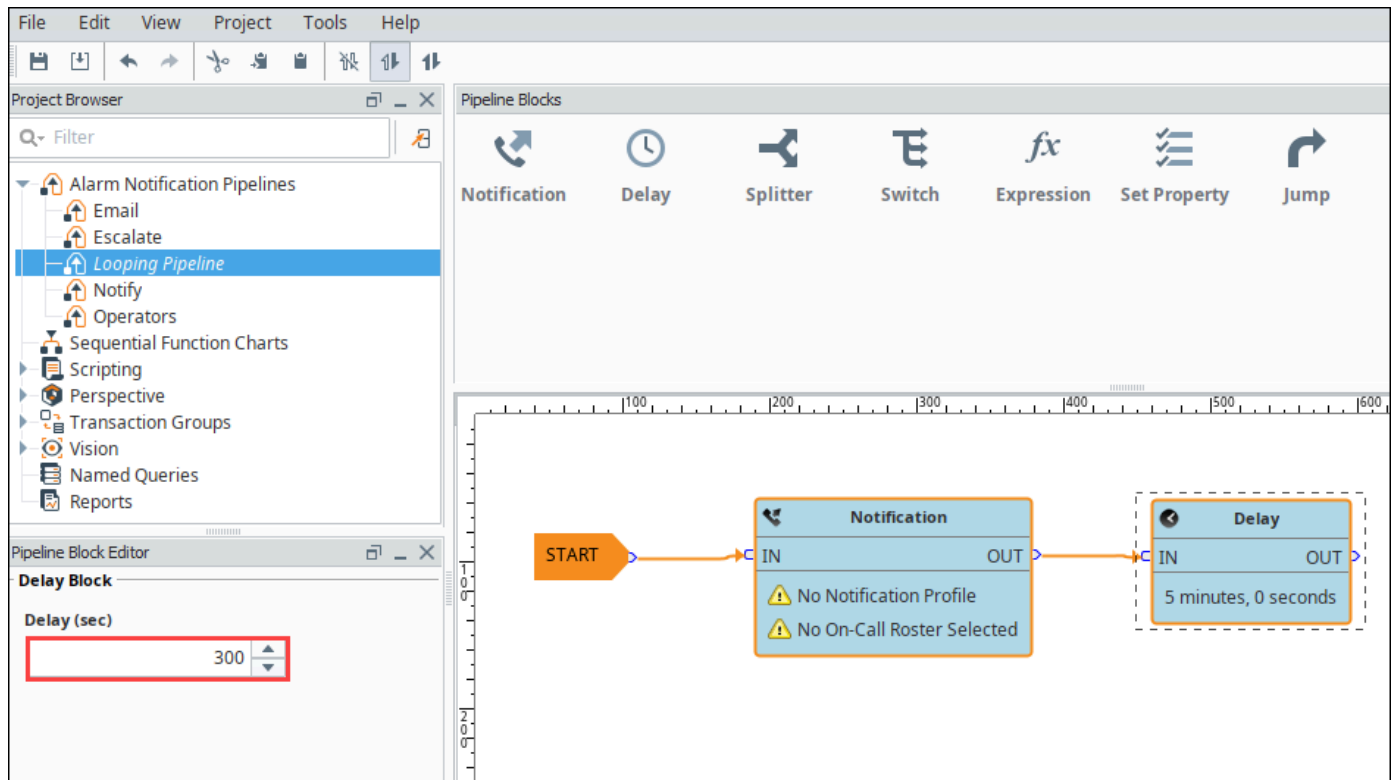
There are two **pipeline blocks** that we need to create a loop. The **Notification Block** to send a notification, and the **Delay Block** to create a pause between each notification.

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

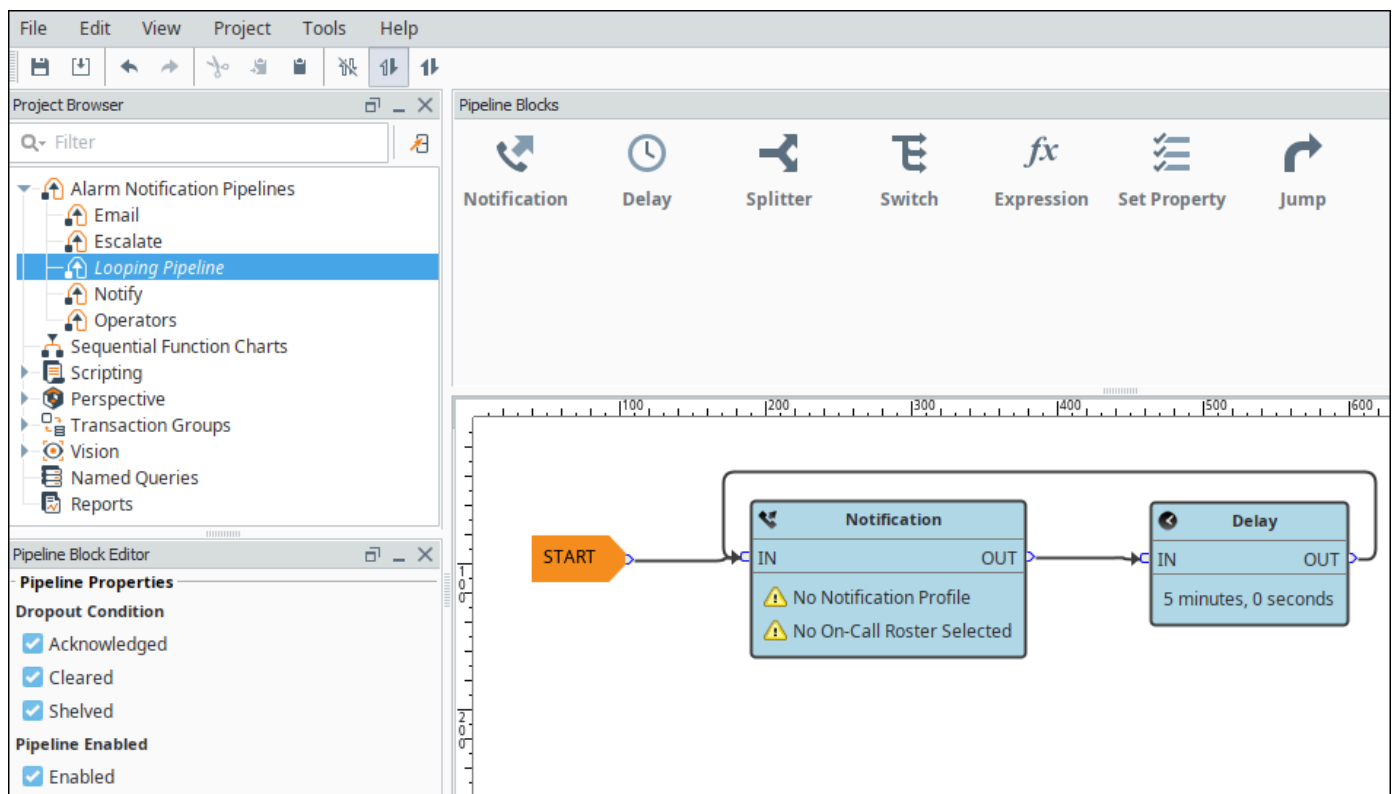
1. Drag in the **Notification Block** in to your pipeline, and connect the **Start** to the **IN** handle of the Notification block. You will need to set up your Notification block, but you can see more about that on the [Notification Block](#) page. This will cause the pipeline to notify immediately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

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 Managers On-Call Roster.

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



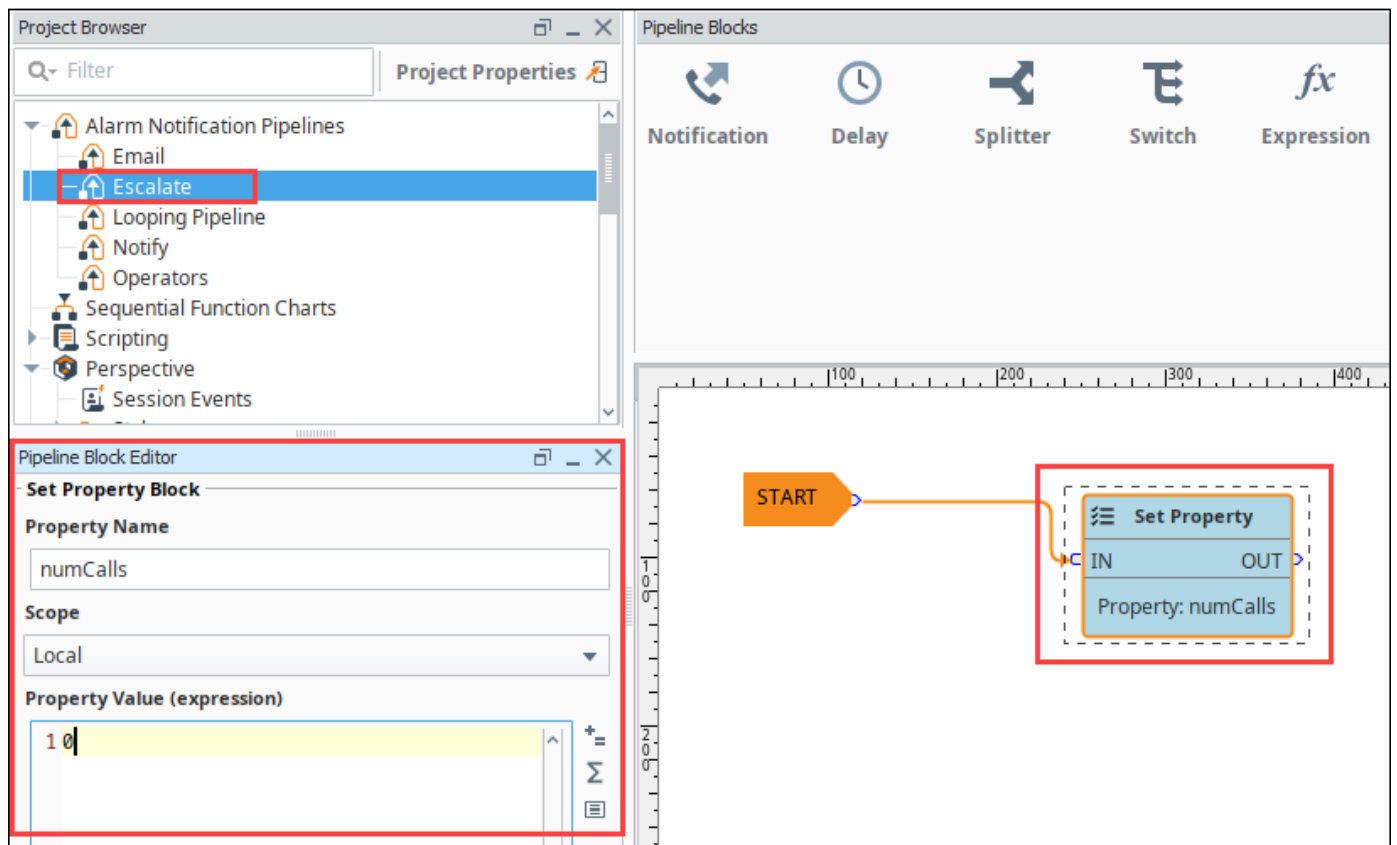
INDUCTIVE
UNIVERSITY

Pipeline - Escalation

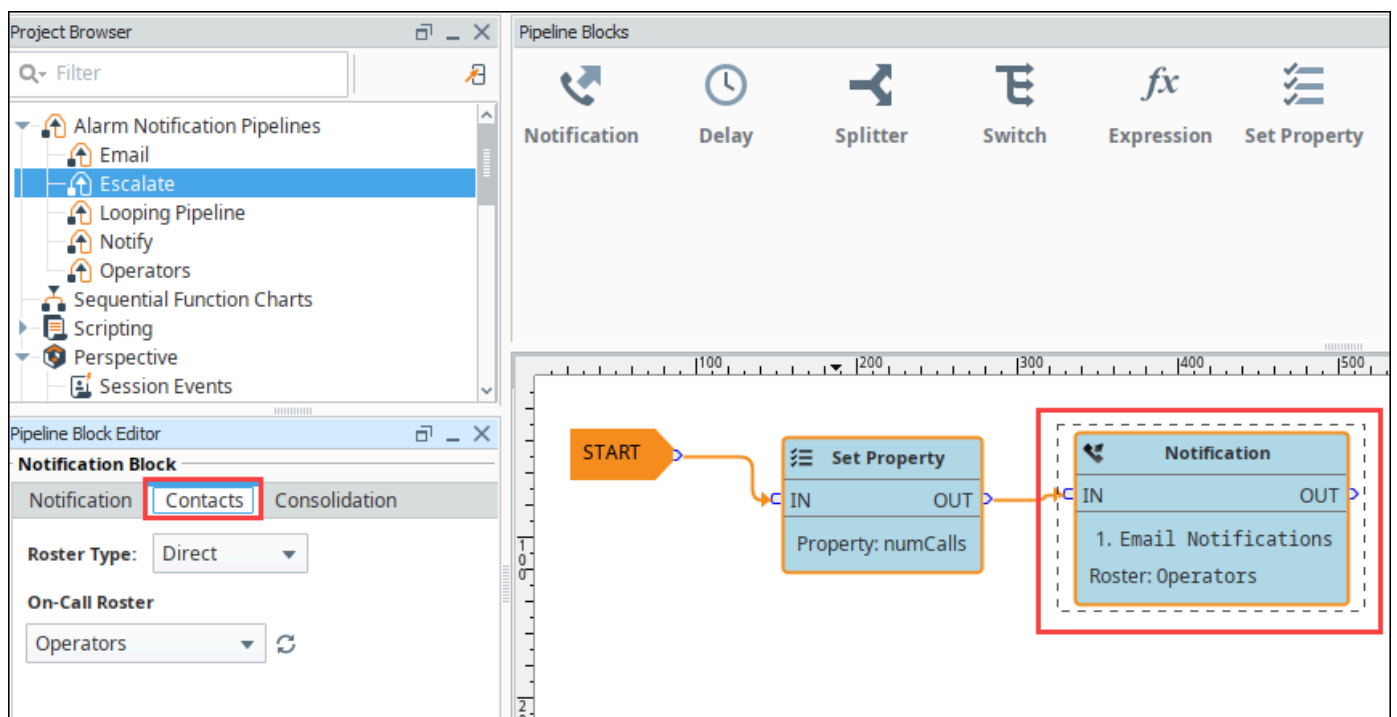
[Watch the video](#)

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"**.

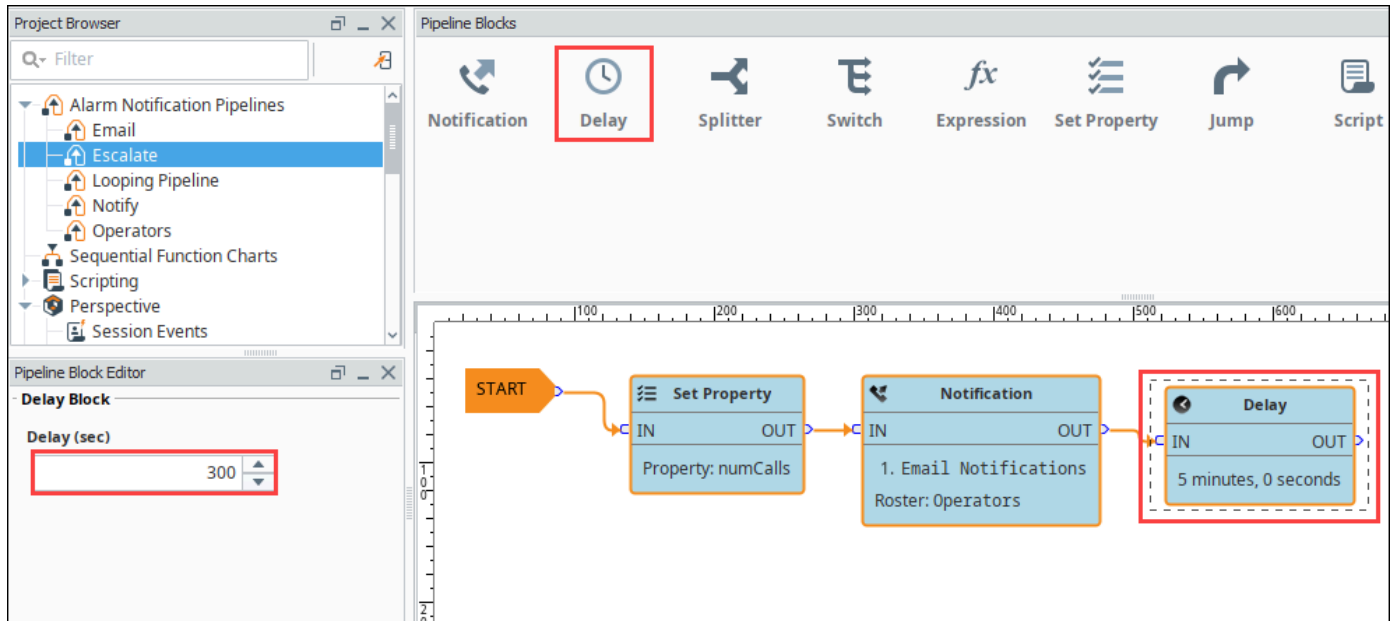


4. Drag the Notification block to the Pipeline Workspace and connect the OUT from the Set Property block to the IN of the Notification block.
5. In the Pipeline Block Editor, set the Notification to use **Email Notifications** and set the Contacts to use the **Operators** on-call roster.



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.

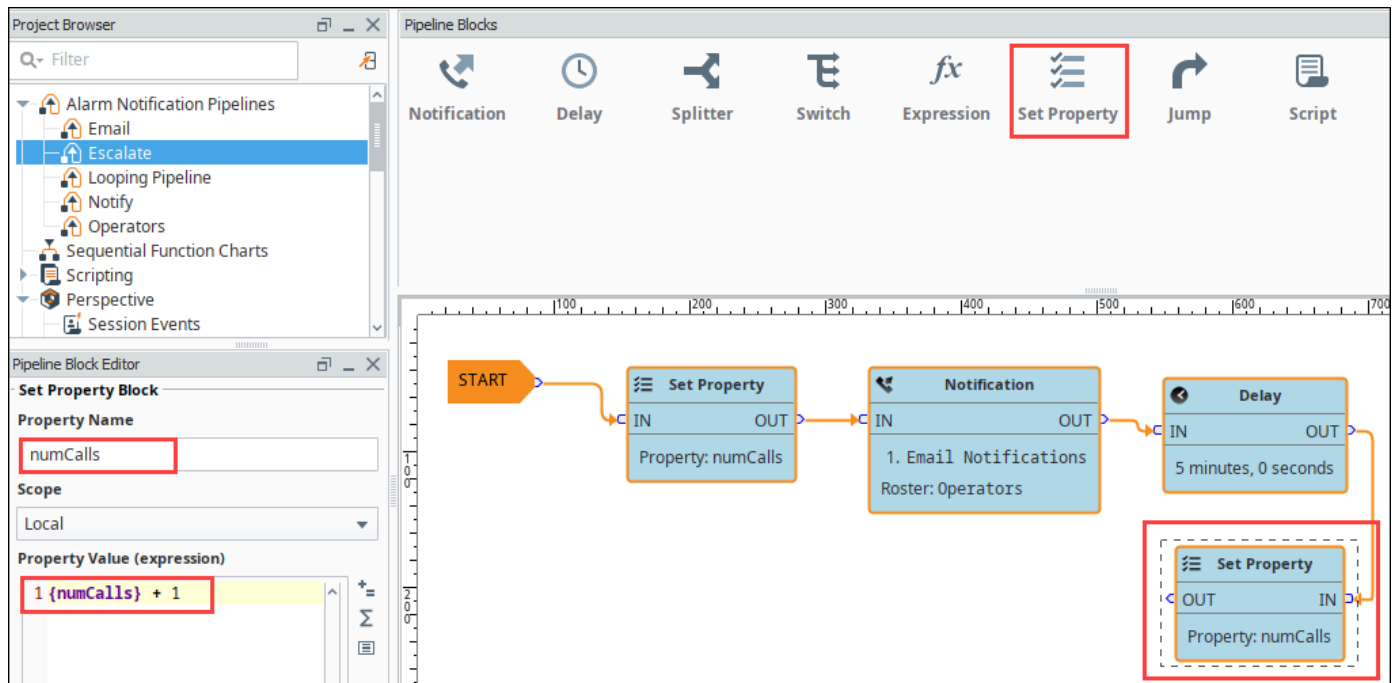


8. If the alarm is still active and not acknowledged, the counter will increment by 1. In order to increment the counter, drag in another Set Property block and connect the **OUT** of the Notification block to the **IN** of the Set Property block. You can right click on the Set Property block to **reverse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

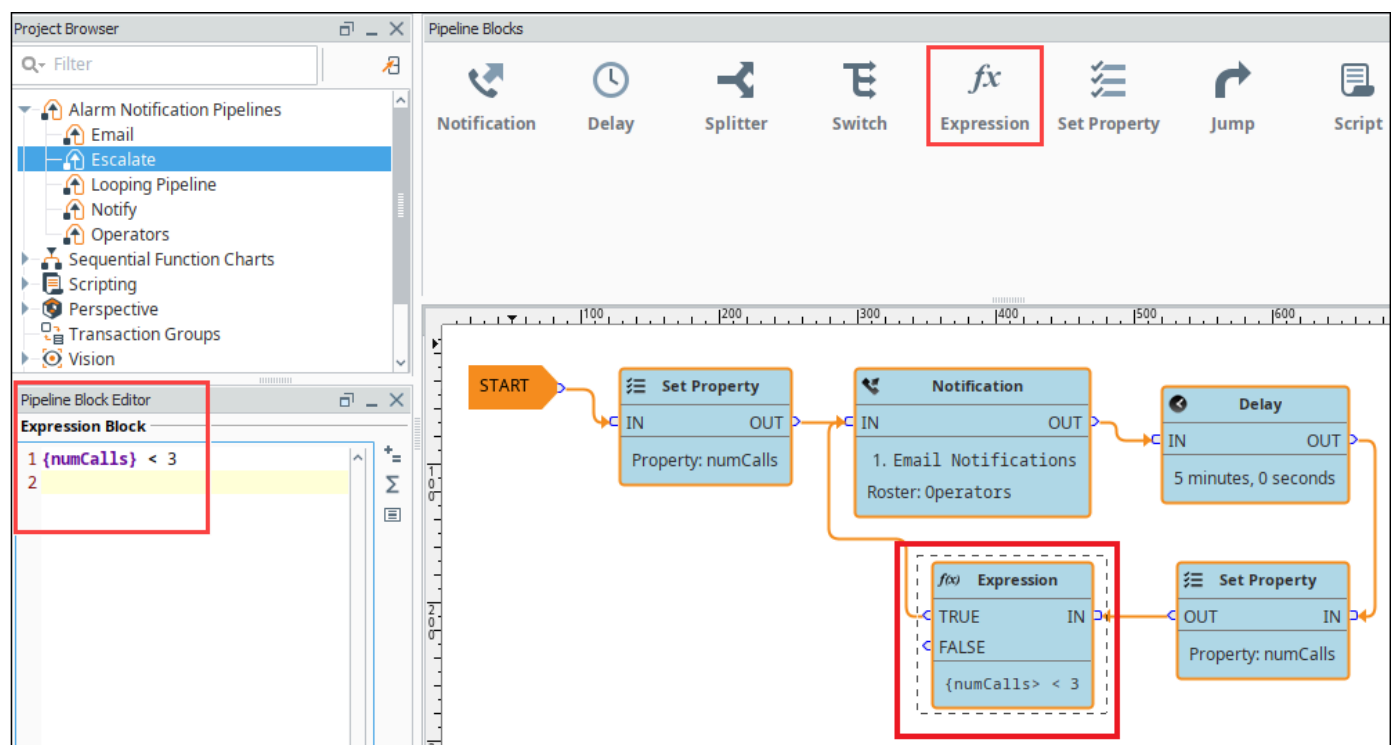
i NOTE

The Set Property Block's **Property Value** uses Ignition's Expression language. More information on the Expression language can be found on the [Expression Language 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.



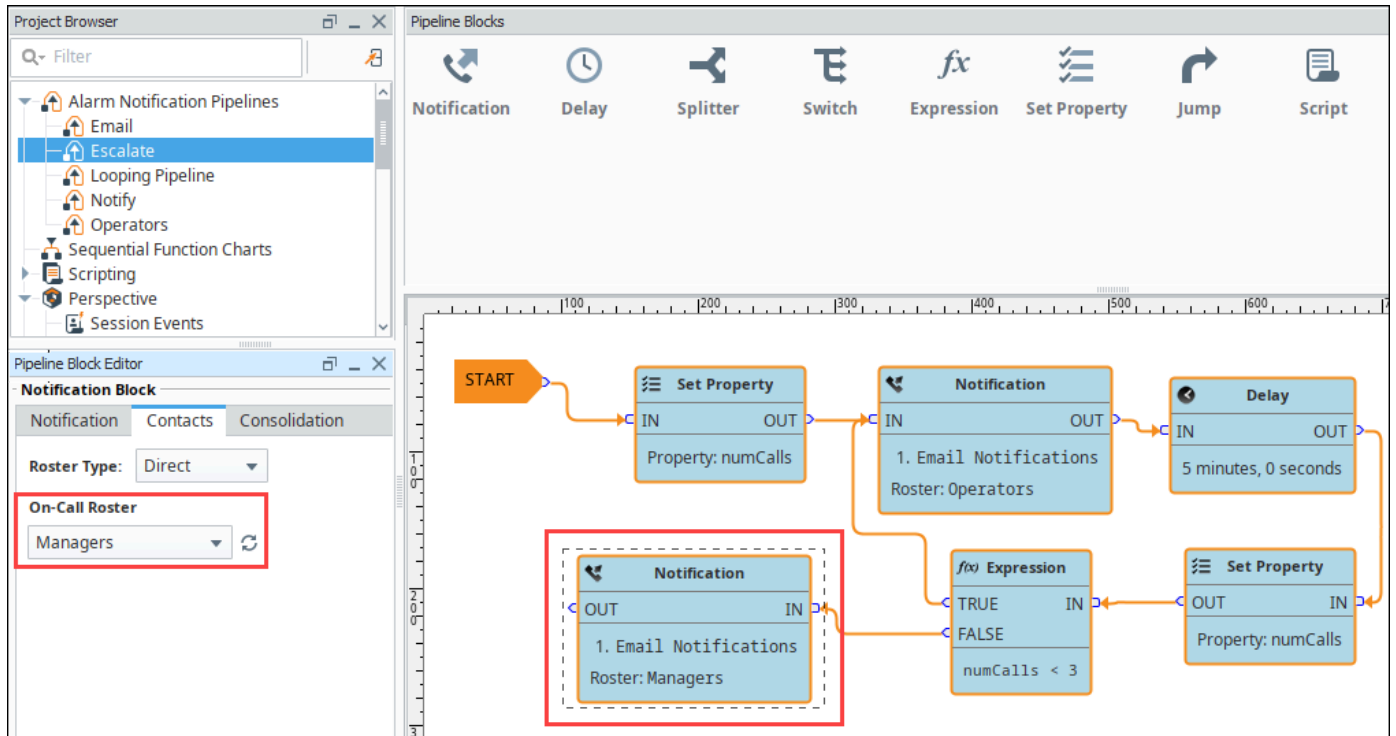
10. The Expression block will check to see if **numCalls** is **< 3**. Drag in an Expression Block to your pipeline workspace and right click on the Expression block to reverse the orientation.
11. Connect the **OUT** of the Set Property block to the **IN** of the Expression block. Since the value of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

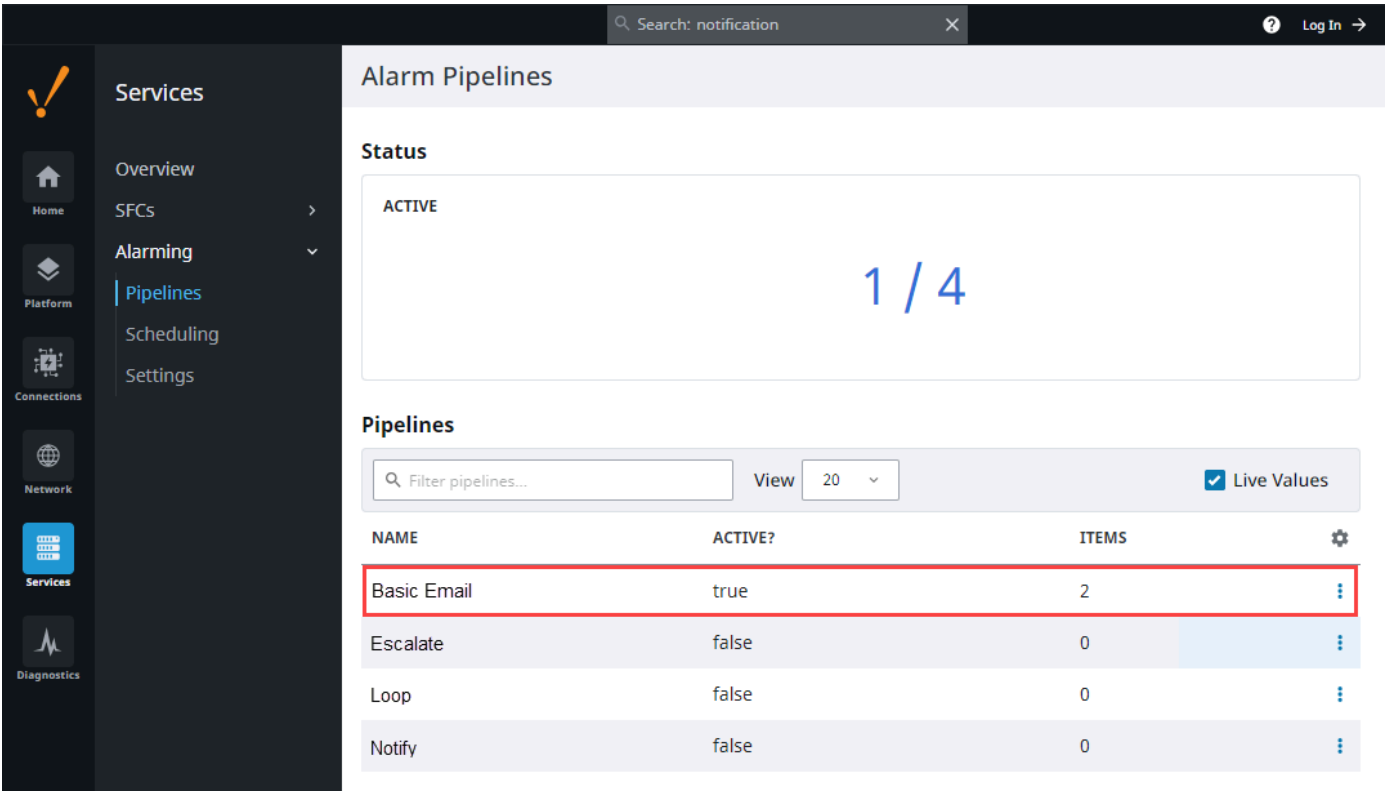
You can also use the [Jump Block](#) as your escalation instead of a second Notification block. This will allow you to jump into another complete pipeline when nobody answers.

Alarm Pipelines Status

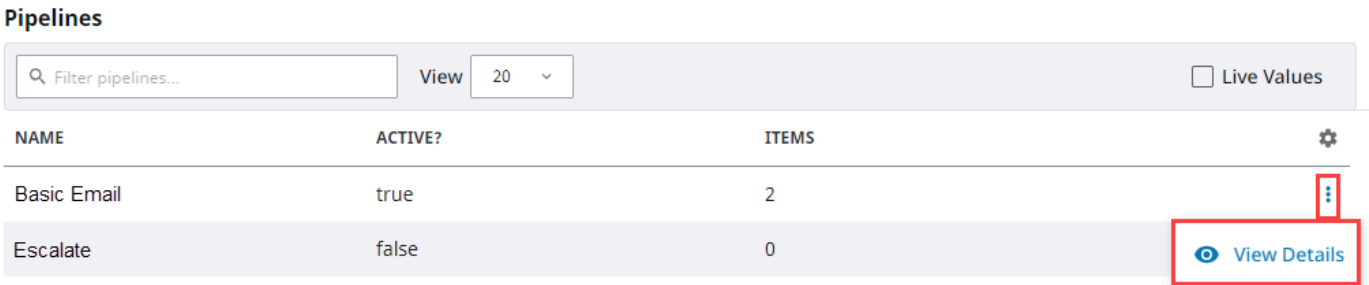
Monitoring the Pipeline Status

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 **Services** tab of your Gateway Webpage and on the left side of the navigation bar, click on **Alarming > Pipelines**. The Alarm Pipelines page will open and display the status of all your Alarm Pipelines. This example shows four pipelines with one active pipeline for Basic Email and three inactive pipelines.



2. To see a detailed status of any of your alarm pipelines, click the **three dots menu** and select **View Details**.



3. A Details panel will appear for the pipeline and show the Source, Display Path, Pipeline Block, Status, and the length of time stuck in a pipeline block for each active alarm. You can **Cancel** the

alarm to delete it from the pipeline, and view the logs by clicking on the **Pipeline Logs** tab.

NOTE

The Pipeline Logs list is a filtered log viewer that only displays logs pertaining to this particular pipeline. A list of all logs can instead be found on the [Diagnostics > Logs](#) page.

< Pipeline Status

Pipeline Logs

>

🔍 Filter statuses...

View 20 ▾

☒ Live Values

SOURCE	DISPLAY PATH	BLOCK	TIME IN BLOCK	STATUS	⚙️
prov:default:/tag:gensim/ Writeable/WriteableFloat1: /alm:Fault1		NotificationBlock	11 sec	🟢 EVALUATION COMPLETE	⋮

How long an alarm stays in the in the Alarm Pipeline Status list 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 can typically be left as is, 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

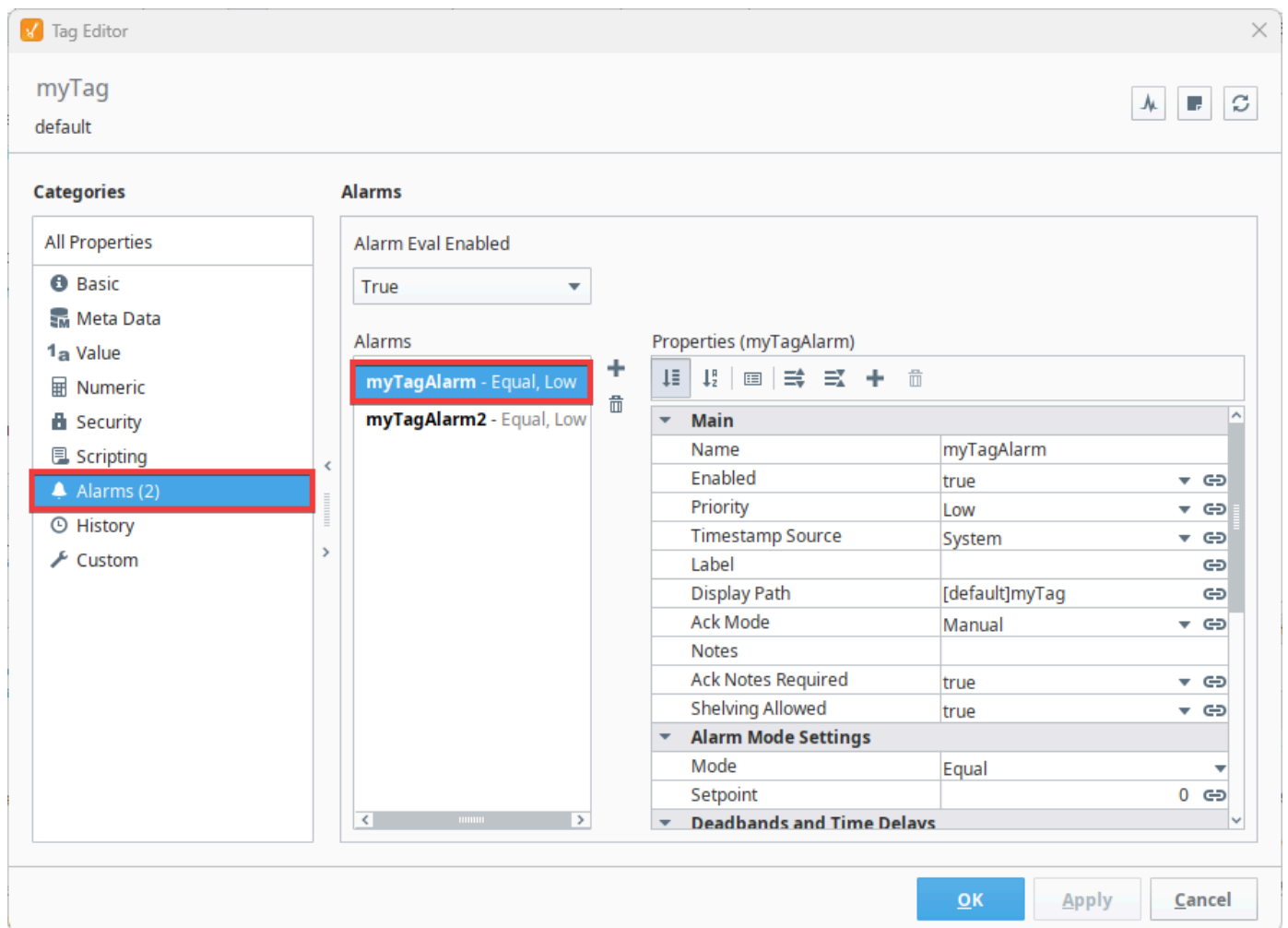
The SMS Notification has a property that allows you to create a custom text that will get sent out when this alarm is triggered.

Voice

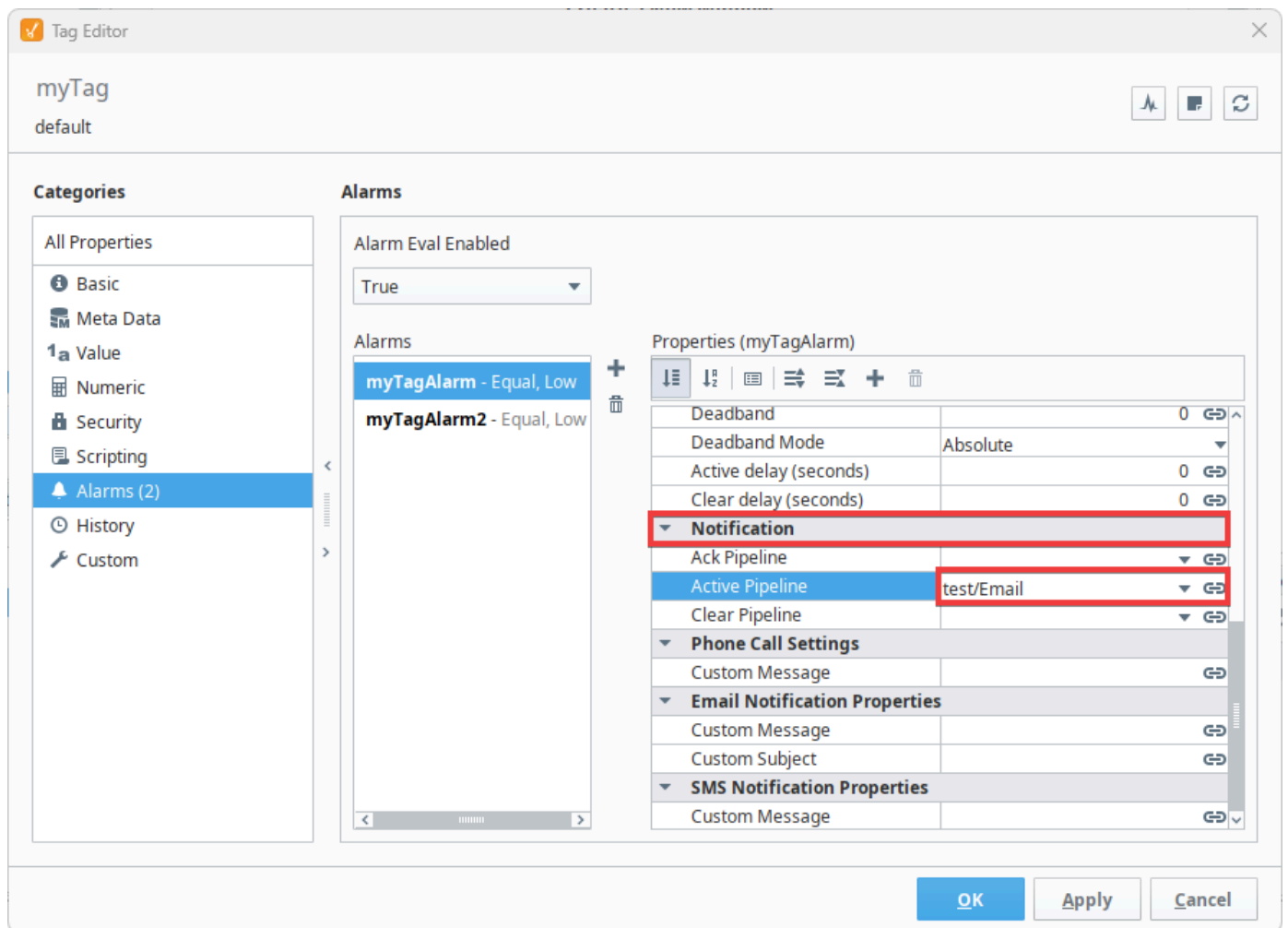
The Voice Notification system allows you to create a custom message on alarms as well. The custom message will only replace the Active Message portion of the voice scripts, and will play the defined script for all other prompts.

Associate a Pipeline with the Alarm

1. Double-click on a tag that already has an alarm configured. The **Tag Editor** is displayed and you can access its properties.
2. Click on the Alarms category and select the alarm you want to edit. **Create** one if you don't have an alarm configured already.



3. Scroll down to the **Notification** section, then select the desired alarm pipeline from the dropdown for any of the alarm status states (Ack, Active, Clear). Typically, operators want to know when an alarm goes active. This example uses the Email Pipeline when the alarm status becomes Active.



4. Click **OK**.

Now, every time the tag's alarm goes Active, it will use the logic in the Basic Email pipeline to notify the operators.