

# Cindev Integrator Tools



## Compatible Control4 Systems:

Designed to work with OS 3.4.2+

## Driver Download, Change Log & Documentation:

<https://drivercentral.io/platforms/control4-drivers/utility/integrator-tools>

---

## Content

- [Overview/Features](#)
- [Dealer Notes](#)
- [Agent Setup](#)
- [Notification Methods](#)
- [Monitored Events](#)
- [Agent Actions \(Composer Commands\)](#)
- [Adding Variables to Messages \(Variable Injection\)](#)
- [Support](#)
- [License Purchase & Assignment](#)
- [Showroom and Free Trial](#)

- [Warranty & Disclaimer](#)
  - [Developer Information](#)
- 

## Overview

The **Cindev Integrator Tools Agent** is a versatile utility designed to enhance the programming efficiency of Control4 systems. It streamlines workflows, simplifies project setup, and provides powerful tools for integrators to manage and customize system behavior.

---

## Dealer Notes

- Driver is an *Agent* and so is located in the **Agents** Section of Control4.
  - This Agent can only be used on Control4 OS3.4.0 and higher.
  - If you delete the Announcement Agent/Announcement the driver was using, you will need to redo setup on Text-To-Voice.
  - *Text-To-Voice*: If *Output Type* is set to **Announcement** will be created by the driver, however Dealers will need to set the rooms the announcement should be triggered in to use.
  - The following Legacy Cindev Agents that have been replaced by this Agent. We recommend removing these older agents and re-working your programming to use Integrator Tools.
    - dealer\_agent
    - voice\_agent
    - email\_plus\_agent
    - login\_agent
- 

## Agent Setup

The Agent is designed to work without any setup, however some features (like email) do require dealer input to function properly.

- Add Agent to Project.
- Go through the different Notification Methods and configure them as needed.

- Use Composer Programming to send Custom Alerts and Trigger Custom Actions.

---

## Agent Notification Methods

### Email

This notification method allows you to send an email or group of emails from a Control4 Project.

- Does not require 4Sight or Connect.
- Daily Email Limit: 50 per day. Note: Email addresses grouped together count as one email.
- Email Groups can be as many email addresses as you want. Separate by a , or space between email addresses.
- Agent is using *SendGrid V3 Email Send API*

### Settings

- *Default Subject* The subject line of emails for Owner, Manager and Other. Dealers can customize the subject line in Composer Programming.
  - Default: **My [CITY] Smart Home**
- *Default Subject (Integrator)* The subject line for emails to the Integrator Email Group. Can be custom in Composer Programming.
  - Default: **Control4 Project: [PROJECT\_NAME]**
- *Default From Name* The Name displayed on the From section of the Email.
  - Default: **Control4 Smart Home**
- *Default From Email* This is where the email is coming from.
  - Default: **controllerMACAddress@cidev.com**
  - This value cannot be changed
- *Default Reply Email* The email address that will get reply responses to emails from the driver.
  - Default: **None**
  - Example: Service department email is (service@cidev.com). When emails from this driver get sent to customers, the customer can then **reply** back to this email address instead of **Default From Email**.

## Composer Commands

- *Send Custom - Email Alert* This will send an email to either a specific email group or a single address.
  - Message: The message/body of the email to be sent.
  - Can also be HTML (*Message is HTML* must be set to Yes).
  - Supports *Variable Injection*
  - To [Owner, Integrator, Manager, Other]: The group to send the email to.
  - To Custom: A single or set of email addresses if you do not wish to use Email Groups.
  - Snapshots: The camera snapshots to include with the email.
  - Supports Multiple Camera Snapshots.
  - Subject: A custom subject line to be used with this email. Leave blank for Default.
  - Supports *Variable Injection*
  - From Name: A custom From Name value to be used with this email. Leave blank for Default.
  - Reply Email: A custom Reply Email Address to be used with this email. Leave blank for Default.
  - Message is HTML: Set to **Yes** if the Message body should be encoded to support HTML characters.

## Push

This notification method uses the Control4 Push Notification Agent and native Control4 Push Alerts available from the MyHome App.

- Requires an active 4Sight or Connect subscription from Control4 and the MyHome App from Control4 to be installed on the customers device.
- No limit to how many can be sent per day.
- A single camera snapshot can be included with any Notification.
- By Default, all customers/devices that have the Control4 App receive *Alarm* and *Alert* notifications.
  - To adjust a devices notification levels: <https://customer.control4.com>

## Settings

- *Default Category* The Default Category that will be used for Push Notifications.
  - Default: **Security**
  - Options: Lighting, Security, Listen and Watch, Comfort
- *Default Severity* The Default Severity that will be used for Push Notifications.
  - Default: **Alert**
  - Options: Alarm, Alert, Info
- *Default Subject (Push)* The Default Subject line that will be used for Push Notifications.
  - Default: **My [CITY] Smart Home**
  - Supports *Variable Injection*

## Composer Commands

- *Send Custom - Push Alert* This will send a push notification with customizable parameters.
  - Message: The message/body of the push notification to be sent.
  - Supports *Variable Injection*
  - Category: The custom Category type for this push notification.
  - Snapshot: The camera snapshot to be included with the push notification.
  - Supports One Snapshot only. Driver allows for multiple snapshots to fix ComposerPro deselection bug.
  - Subject: A custom subject line to be used with this email. Leave blank for Default.
  - Supports *Variable Injection*

## Text to Voice

This notification method allows dealers to send text to voice alerts via a variety of output types.

- Does not require 4Sight or Connect.
- Daily Voice Limit: 12000 per day. A character is any ASCII character (including punctuation and spaces) in the message.
- Agent is using Google Test-To-Speech API.

## Voice Alert Output Types

- **Announcement Agent**

- Uses the *Announcement Agent* to play a audio.wav file in specific digital media rooms in the project and/or touchscreens.
- The Agent can create the announcement template, however dealers will need to define which rooms and volume levels via the Announcement Agent.

- **C4 T3/T4 Touchscreens**

- Plays an audio.wav file on specific touchscreens in the project.
- The alerts are not in sync. There is not a way to have all touchscreens play the audio file at the same time.
- T3 Touchscreens using WiFi sometimes have issues properly playing audio files. To fix, reboot the T3 Touchscreen.

- **Sonos V2 Rooms**

- Plays an audio.wav file to the connected Sonos rooms.
- Must be using Sonos Network Driver version 104 or newer and Sonos Zone Player Driver 99 or newer
- Alerts to multiple rooms are not in sync. We recommend choosing an announcement are per floor or area to prevent echo.

## Settings

- *Voice Example* A website containing all the possible voice languages and models.
- *Voice Language* The Voice Locale to be used by the Agent.
  - Default: **en-US**
  - Options: Dynamically populated based on API.
- *Voice Gender* A filter for the voice models to to choose from
  - Default: **Female**
  - Options: Male, Female
- *Voice Model* The current voice model to be used by the driver.
  - Default: **Neural2-C**
  - Options: Dynamically populated based on API.

## Composer Commands

- *Send Custom - Voice Alert* This will read a message and play the audio via the selected output methods.
  - Message: The message to be transcribed and converted into an audio.wav file.
  - Supports *Variable Injection*
  - Touchscreen: Specific touchscreens to play the audio file on.
  - Touchscreen Volume: The volume level to play the audio file at.
  - Announcement: If **Yes** the Agent will play the audio file to the announcement agent template (if setup)
  - Sonos Rooms: A custom set of Sonos Rooms to play the audio file.
  - Sonos Volume: Custom Volume level for the selected Sonos Rooms.
  - Voice Model: A custom voice model for this alert.

## Navigator Popup

This notification method allows dealers to create a popup message on Control4 Navigators.

- Does not require 4Sight or Connect.
- Can use Touchscreens or EA3/Core Controller HDMI Outputs
  - If using an EA3/Core for the popup, dealers will need to select/choose the Control4 GUI first then trigger the event.
- A camera snapshot or static image can be included with the popup.

### Settings

- *Default Navigators* The Control4 UIs that will display the popup message.
- *Default Display Time* The default amount of time (seconds) the message will be displayed on the selected Navigators.
  - Default: **30**
  - Note: If set to **0** the popup will be displayed until a *Refresh Navigators* or significant Control4 action... Like a reboot.
- *Default OK Button* Sets if the default popup should have an *OK* button the customer can click. If this is set to **Hidden** the popup will be displayed and

disappear after the *Display Time* amount.

- Note: This can be used as a simple touchscreen lockout feature.
- *Default Image URL* The default image URL to be used with the popup.
  - Default: **None**
  - Note: Recommend images no bigger than 300x300px and be .png file type.
  - Note: image url should be in the format of:  
**http://ipOrDomain/imagePath.png**

## Composer Commands

- *Send Custom - Navigator Popup Alert* This will display the Message on selected Control4 Navigators.
  - Message: The message to be displayed.
  - Supports *Variable Injection*
  - Snapshot: The camera snapshot to be included with the notification.
  - Supports One Snapshot only. Driver allows for multiple snapshots to fix ComposerPro deselection bug.
  - Image URL: An optional image to be included with the popup message. This overrides a Camera Snapshot.
  - Navigators: Choose specific navigators for this popup message. Leave blank to use Default Navigators.
  - OK Button: Select a custom OK Button setting for this notification.
  - Display Length: Selects a custom display length for this notification.
  - **-1** will use Agent *Default Display Time* value
  - **0** will keep the notification until OK is pressed or cleared.
  - All other values will be the amount of time to display the notification (seconds).

---

## Monitored Events

The Cindev Integrator Tools Agent automatically monitors most service and project level events and gives dealers the ability to quickly assign who gets which alerts. Below are the monitored events and specific notification and dealer notes about each.



Note: All Events can send any of the Notification Methods. Only Notifications that are active by default will be displayed below for brevity.

## Project Loaded

This event happens when the project loads and was the same version before the reload. Normally this occurs have a Controller/Director Reboot.

### Default Notifications

- Default Message: **[PROJECT\_NAME] has loaded**
- Email Integrator: **Yes**

## Project Updated

This event happens after a project OS has been updated.

### Default Notifications

- Default Message: **[PROJECT\_NAME] has been updated to [OS\_VERSION]**
- Email Integrator: **Yes**

## Driver Updated

This alerts occurs when a driver has been updated to a different version. Reloading the driver will not cause this event to occur.

### Default Notifications

- Default Message: **Driver [FILE] has been updated to v[DRIVER\_VERSION]**
- Email Integrator: **Yes**

### Additional Injectable Variables

- FILE: The C4Z filename of the driver that triggered the event.
- DRIVER\_VERSION: The driver version of the driver that triggered the event.

## Driver Added

This event occurs when a driver has been added to the project.

### Default Notifications

- Default Message: **[DRIVER\_NAME] ([FILE]) has been added to the project**

- All Default Notifications Set to **No**

### Additional Injectable Variables

- DRIVER\_ID: The device ID value for the driver that was added.
- DRIVER\_NAME: The name of the driver that was added.
- DRIVER\_VERSION: The driver version of the driver that was added.
- FILE: The C4Z filename of the driver that was added.

## Driver Removed

This event occurs when a driver has been removed from the project.

### Default Notifications

- Default Message: **[DRIVER\_NAME] ([FILE]) has been removed from the project**
- All Default Notifications Set to **No**

### Additional Injectable Variables

- DRIVER\_ID: The device ID value for the driver that was removed.
- DRIVER\_NAME: The name of the driver that was removed.
- FILE: The C4Z filename of the driver that was removed.

## Composer Login

This event occurs when a technician connects to Director from ComposerPro.

- Supports Remote and Local connections

### Default Notifications

- Default Message: **[USER] has logged into ComposerPro [LOCATION]ly**
- Push Alert: **Yes**
- Email Integrator: **Yes**

### Additional Injectable Variables

- USER: The email address of the user that logged into ComposerPro
- LOCATION: The login location type. *Local* or *Remote*

# Composer Logout

This event occurs when all logged in ComposerPro users have been logged out.

- All users must be logged out for at least 3 minutes
- After 8 hours, Agent stops tracking logged in user.

## Default Notifications

- Default Message: **All users have logged out of ComposerPro**
- All Default Notifications Set to **No**

# Lighting Hardware Issue

This event occurs when a monitored lighting device has one of the following issues:

## Monitored Variables

- SHORT\_CIRCUIT\_DETECTED
- OVER\_TEMPERATURE
- OVER\_RATED\_WATTAGE

## Monitored Drivers

- *din\_rail\_8\_channel\_dimmer\_module.c4i*
- *din\_rail\_8\_channel\_fpd\_module.c4i*
- *combo\_dual\_dimmer.c4i*
- *0\_10v\_dimmer.c4i*
- *combo\_dimmer.c4i*
- *adaptive\_phase\_dimmer.c4i*
- *forward\_phase\_dimmer.c4i*
- *switch\_gen3.c4i*
- *fan\_speed\_controller.c4i*
- *puck\_switch\_gen3.c4i*
- *combo\_switch.c4i*
- *puck\_dimmer\_gen3.c4i*

## Additional Injectable Variables

- DRIVER\_NAME: The name of the driver that had the alert.
- ISSUE: The kind of issue (Monitored Variable Name) that occurred.

## Climate Hardware Issue

This event occurs when a monitored climate device has an issue.

### Monitored Variables

- Thermostat Error

### Monitored Drivers

- *control4\_thermostat\_c4therm.c4z*

### Additional Injectable Variables

- DRIVER\_NAME: The name of the driver that had the alert.
- ISSUE: The kind of issue (Monitored Variable Name) that occurred.

## Low Battery

Each device handles battery level/alerting differently. Cindev has spent years analyzing this device data and built a monitoring system designed to work with any battery device in Control4. Below is the alert flow:

- Once a day, the Agent will scan the project and review battery levels
  - Driver will check levels of each monitored battery
  - Each device level value is then validated against its device type threshold

**If** device level is low and not in *Alert*, set device to **Warning**

- After 5 failed scans, set device to **Alert**
- Failed scan counter resets to 0 anytime a normal battery level is detected

After all monitored devices are checked... Drivers found that are in **Alert** State will trigger the *Low Battery* Event for each driver once.

Below is the list of C4I/C4Z driver filenames the battery agent will monitor. If you do not see the device listed and would like it added to this agent, please submit a ticket with the c4i/c4z filename to: <https://help.drivercentral.io>

### Control4 Hardware

- [control4\\_sr250.c4i](#)
- [control4\\_sr260.c4i](#)
- [control4\\_sr150.c4i](#)
- [control4\\_srh1150.c4i](#)
- [control4\\_thermostat\\_c4therm.c4z](#)
- [control4\\_wirelesscontact.c4i](#)
- [control4\\_imr10a-p\\_driverworks.c4i](#)
- [control4\\_imr10a\\_driverworks.c4i](#)
- [C4-Z2C.c4z](#)
- [control4\\_wirelesspir.c4i](#)
- [control4\\_neeo\\_remote.c4z](#)
- [control4-halo-remote.c4z](#)

## Card Access Hardware

- [contact\\_cardaccess\\_sps10\\_driverworks.c4i](#)
- [cardaccess\\_IMR10A-P\\_DriverWorks.c4i](#)
- [cardaccess\\_IMR10A\\_DriverWorks.c4i](#)
- [cardaccess\\_wirelesspir.c4i](#)
- [cardaccess\\_wirelesscontact.c4i](#)
- [contact\\_cabridge\\_micro\\_dws.c4i](#)
- [contact\\_cabridge\\_water.c4i](#)
- [cardaccess\\_sensorbridge\\_ZGB20.c4i](#)

## Locks

- [august\\_lock\\_cin.c4z](#)
- [yale\\_lock\\_cin.c4z](#)
- [brisant\\_ultion\\_door\\_lock.c4z](#)
- [chowmain\\_elk\\_m1\\_lock.c4z](#)
- [lock\\_kwikset\\_zigbee\\_v2.c4z](#)
- [yale-control4-3.c4z](#)
- [Lock\\_Zigbee\\_Baldwin\\_SmartLock.c4i](#)
- [DriverWorks\\_BlackDecker\\_Lock.c4i](#)
- [lock\\_yale\\_zwave.c4z](#)

- [lock\\_weiser\\_zwave.c4z](#)
- [lock\\_baldwin\\_zwave.c4z](#)
- [lock\\_kwikset\\_zwave.c4z](#)
- [lock\\_schlage\\_zwave.c4z](#)
- [Lock\\_ZigBee\\_Yale.c4i](#)
- [Lock\\_Zigbee\\_Kwikset\\_SmartLock.c4i](#)
- [Lock\\_ZigBee\\_Yalev2.c4z](#)

## **C&S (Dammit) Water Sensors**

- [waterfreezesensor\\_zigbee\\_candselectronics\\_wwa.c4i](#)
- [watervalve\\_zigbee\\_candselectronics\\_wv01.c4i](#)

## **Nyce**

- [DriverWorks\\_HA\\_Profile\\_MSensor.c4i](#)
- [DriverWorks\\_HA\\_Profile\\_KeyFob4B.c4i](#)
- [HingeSensor\\_zb\\_nyce\\_3010.c4z](#)
- [GarageSensor\\_zb\\_nyce\\_3014.c4i](#)
- [GarageSensor\\_zb\\_nyce\\_3014.c4z](#)
- [DriverWorks\\_HA\\_Profile\\_KeyFob.c4i](#)
- [MotionSensor\\_zb\\_nyce\\_3043.c4i](#)
- [MotionSensor\\_zb\\_nyce\\_3041.c4i](#)
- [MotionSensor\\_zb\\_nyce\\_3045.c4i](#)
- [DoorWindow\\_zb\\_nyce\\_3011.c4i](#)
- [DriverWorks\\_HA\\_Profile\\_CSwitch.c4i](#)

## **Shades/Blinds**

- [blind\\_qmotion\\_qadvanced\\_roller\\_shade.c4z](#)
- [hunterdouglas\\_powerview\\_shade\\_cin.c4z](#)
- [hunterdouglas\\_powerview\\_shade\\_1\\_cin.c4z](#)
- [hunterdouglas\\_powerview\\_shade\\_2\\_cin.c4z](#)
- [hunterdouglas\\_powerview\\_shade\\_3\\_cin.c4z](#)
- [hunterdouglas\\_powerview\\_gen3\\_shade\\_1.c4z](#)
- [hunterdouglas\\_powerview\\_gen3\\_shade\\_2.c4z](#)

- [hunterdouglas\\_powerview\\_gen3\\_shade\\_3.c4z](#)

## Nest

- [protect\\_nest27.c4i](#)

## Axxess

- [axxess\\_contactsensor.c4i](#)
- [axxess\\_motionsensor\\_MS-C4ZB-11.c4i](#)
- [axxess\\_doorbell\\_DB-C4ZB-11.c4i](#)
- [axxess\\_motionsensor.c4i](#)
- [axxess\\_motionsensor.c4z](#)
- [axxess\\_motionsensor\\_v3.c4i](#)

## Sinope

- [water\\_valve\\_eb\\_zigbee\\_Sinope.c4z](#)
- [water\\_sensor\\_zigbee\\_Sinope.c4z](#)

## Shelly

- [shelly\\_flood.c4z](#)

## Additional Agent Notes for Low Battery Monitoring

- Some devices only trigger a Composer Event. When these events occur, the failed scan count for this driver will be set to 5 (**Alert**)
- New monitored drivers are added automatically to the project. Dealers can adjust the monitored drivers via the **Device List** property.
- For custom battery devices or other battery type alerts that the driver is not monitoring, dealers can use the Composer Command **Simulate Battery Alert** and handle trigger in programming.
- Customers using the Axxess Motion Sensors will need to send a Zigbee packet to the motions in order for them to update battery levels correctly. The procedure to do this is as follows:
  - On each motion in the project do the following:
  - In the Actions tab of the Motion Sensor driver click on the Enable Lua button and enter the password sesame123

- Then go to the Lua tab in the driver and in the top window enter:
- `SendZigbeePacket(TXCMDLIST["Set_Hardware"], "0200")`
- Press Execute (top right of lua tab)
- Go back to the Actions and hit the Get Hardware Config button.

## Default Notifications

- Default Message: **Low Battery ([DRIVER\_NAME] - [FILENAME]) [LOW\_LEVEL][SUFFIX]**
- Email Integrator: **Yes**

## Additional Injectable Variables

- DRIVER\_NAME: Driver Name of device that triggered event
- SUFFIX: Level suffix attribute based on the current low level value. Blank if not used.
  - Example *Low Level of 9%*: Low Level of [LOW\_LEVEL][SUFFIX]. The Suffix would be %
- LOW\_LEVEL: The current level value of the device reporting a low battery.
- TYPE: The kind of driver type (Sensor, Remote, Lock, Shade etc...) of the driver that triggered the event.
- FILENAME: The driver filename of the driver that triggered the event.
- ROOM\_NAME: The Room Name of the driver that triggered the event.

---

## Agent Actions (Composer Commands)

- *Action - Set LED Backlight Color* Sets the backlight color on selected Control4 Keypads and Lighting devices.
  - Device(s): Selects which Control4 devices you want to adjust
  - `switch_gen3.c4i`
  - `combo_dimmer.c4i`
  - `adaptive_phase_dimmer.c4i`
  - `forward_phase_dimmer.c4i`
  - `configurable_keypad.c4i`
  - `fan_speed_controller.c4i`



- Color: The color to set the backlight to.
- *Action - Set LED Ambient Light Profile* Sets the ambient light profile on selected Control4 Keypads and Lighting devices.
  - Device(s): Selects which Control4 devices you want to adjust
    - switch\_gen3.c4i
    - combo\_dimmer.c4i
    - adaptive\_phase\_dimmer.c4i
    - forward\_phase\_dimmer.c4i
    - configurable\_keypad.c4i
    - fan\_speed\_controller.c4i
  - Ambient Profile: Selects the LED Light Profile for the device. Each profile is based on the standard 4 Ambient Profiles available on the Control4 Lighting Device Configuration Page.
    - 1 [Dim, Dim - Bright, Bright - Bright, Off]
    - 2 [Off, Off - Bright, Bright - Bright, Off]
    - 3 [Dim, Dim - Bright, Bright - Bright, Bright]
    - 4 [Off, Off - Bright, Bright - Bright, Bright]
- *Action - Set LED Status Color* Sets the LED color for selected Control4 Keypads and Lighting devices.
  - Device(s): Selects which Control4 devices you want to adjust
    - switch\_gen3.c4i
    - combo\_dimmer.c4i
    - adaptive\_phase\_dimmer.c4i
    - forward\_phase\_dimmer.c4i
    - configurable\_keypad.c4i
    - fan\_speed\_controller.c4i
  - Button: Selects which buttons on the devices to adjust the LED Status Color.
    - Dimmer devices do not support Button 1 - 6
    - Keypad devices do not support Button Top, Bottom and Toggle
  - LED: Selects which LED to adjust
  - Current LED Value: Sets the current active or ON value of all selected device/buttons.

- On LED Value: Sets the On LED value of all selected device/buttons.
- Off LED Value: Sets the Off LED value of all selected device/buttons.
- All LED Values: Sets the Current, On and Off Values of all selected device/buttons.
- Color: The color to LED color to.
- *Action - Turn off Room(s)* Turns Off selected room Audio/Video, Lights and Shades.
  - Room(s): Selects which rooms you want to control.
  - All Rooms: If **Yes** then all rooms will be used. This setting overrides *Room(s)*.
  - Turn Off AV: If **Yes** then the *ROOM\_OFF* command will be sent to all selected rooms.
  - Turn Off Lights: If **Yes** then all lights in the selected rooms will be sent *OFF*
  - Close Shades: If **Yes** then all shades in the selected rooms will be send *CLOSE*
- *Action - Start Flash Light(s)* Will Start Flashing (Toggle On/Off) selected lights for the specified duration.
  - Note: Only One Flash Lights program can be running at a time.
  - Note: Lights are sent a *TOGGLE* command for each light. Lights will not be in sync and popcorn effect may happen on many lights.
  - Note: After the *Flash Lights* finishes, the lights will be set to the previous levels.
  - Light(s): Selects which lights to flash.
  - All Lights in the project are compatible
  - Interval (seconds): The amount of time between *TOGGLE* commands.
  - Run Time (minutes): The total amount of time the agent will flash the lights.
  - Set to **0** to have the lights flash until the *Stop* action is received.
- *Action - Stop Flash Light(s)* Will Stop any Flash Lights program running.
- *Action - Override Notifications* Will temporarily override the notification ability for a specific notification method.
  - Length (Hours): The amount of time to override the selected notification method

- Notification Methods: Select which method to override
- No Change: Will no do anything.
- Disable: Will block/mute notifications using this notification method.
- Enable: Will unblock/allow notifications for this notification method.
- *Action - Clear Override Notifications* Will reset any override notifications back to original settings.
- *Send Custom - Alert Template* Will send an alert message to the selected notification methods using default values.
  - Pre-Built Message: A list of standard alert messages for simple notifications.
  - Message: The custom message to be sent to the selected notification methods.
  - Supports *Variable Injection*
  - Snapshot: Selects the camera snapshots to be included with the alert.
  - Push and Navigator Popup can only display a single camera snapshot. If multiples are chosen the first snapshot is used.
  - Notification Methods: Choose which notification methods should be used for this command.
  - Push Alert, Email Owner, Email Integrator, Email Manager, Email Other, Voice Alert, Navigator Popup

---

## Variable Injection

- Variable Injection Customize the alert message by using variables found throughout the project.

---

## Support

If you require additional driver assistance or are having some issues please visit our help desk: <https://help.drivercentral.io>

## Showroom and Free Trial

All Cindev drivers for Control4 come with a 7-day trial. Simply add the DriverCentral licensing driver, *Cloud Driver*, along with this driver and you can

fully test the driver before purchasing. Including Cindev drivers in your drivercentral showroom project will give you unlimited use without purchase.

## Warranty & Disclaimer

[www.cindev.com/terms-and-conditions](http://www.cindev.com/terms-and-conditions)

---

## Developer Information

brought to you by:  
Cinegration Development, LLC



[www.cindev.com](http://www.cindev.com)  
[www.drivercentral.io/cindev/](http://www.drivercentral.io/cindev/)

We are always looking to improve our drivers.  
Please send your suggestions to: [info@cindev.com](mailto:info@cindev.com)