

## TECH NOTE

# HOW-TO GUIDE Sending Ignition notifications to Telegram

This Technical Note contains all the information required to set up a connection between Ignition and Telegram to take advantage of the communication protocols available in Ignition to send notifications to an instant messenger application.

### Element8 Tech Note

TOC	
	<ol> <li>Introduction</li></ol>

### 1. Introduction

Ignition can be configured to send notifications to Telegram. This technical note describes the principal points to follow for a successful configuration.

Telegram is a freeware, cross-platform, cloud-based instant messaging software and application service. Telegram is one of the most secure IM services available as it uses a range of encryption technologies to secure messages and media. This application is available for free download on all app stores.

Ignition is an industrial application platform for collecting data, designing and deploying industrial applications throughout an enterprise. It empowers the creation of any kind of industrial application including SCADA, MES, IIoT, reporting, alarming and more. The trail is available for free download on inductiveautomation.com

To set up the connection between these two components, a Telegram bot has to be created. A bot is a software application that runs automated tasks over the Internet. In this case, a bot is a third-party application that runs inside Telegram. You can interact with them by sending them messages and requesting data. You can control them by using HTTPS requests to the bot API.

Telegram will provide us with a chat ID and API token which is essential for connection. A chat ID is a numeric value that identifies you. You can use a bot to find your personal chat ID. An API token is an access token that contains the security credentials for a login session and identifies the user, groups, privileges and applications.

## There are 4 main steps that you need to follow to send notifications from your Ignition project to Telegram:

- 1. Download and configure a bot in Telegram
- 2. Find the chat ID and API token in Telegram
- 3. Create a basic Ignition project / Launch an existing project
- 4. Set up the connection between Ignition and Telegram

These steps are described in greater detail in the following sections

### 2. Configuring a Telegram Bot

The first step is to create a new bot in Telegram. To do that we need to use BotFather; it is the main bot used in Telegram to create new bots and change settings for existing ones. Then the API token and chat ID can be

1. Firstly, create a new chat in Telegram. Search for "BotFather" and click on the result to start a new chat.



2. You will be greeted with this message. When you click on the "Start Bot" button, you will receive the following commands in conjunction with creating a new bot and bot settings.



3. Use the /newbot command to create a new bot. The BotFather will ask you for a name and username, then generate the authorization token for your new bot. This is your API token.



4. Secondly, you need to allow your bot to be added to groups. In your current chat with BotFather, type '/mybots' to access your newly created bot. Click on the 'Bot Settings' option. Click on the 'Allow Groups?' option.



5. This option enables the option for your bot to be added to a group chat. Make sure that it is enabled before mobbing onto the next step.



6. Lastly, a group chat ID is needed. A group needs to be created, and the bot needs to be added to that group.



The bot needs to be added to the group. Type in the handle name of the bot and select your bot once it appears in the search results. Enter a name for the group and click on "Create".

Back New Group Next kgritionNotifications WertyuioPickinskotfications WertyuioPickinskotfications WertyuioPickinskotfications kgritionNotifi	8:58 ≁		al 🕆 🗖	08:58	4	÷ ا
philonNotifications IgnitionNotifications Wertyuiop asdfghjkl Zxcvbnm 123 space dorm NotifGroup NotifGroup NotifGroup NotifGroup NotifGroup NotifGroup 123 space retur	ack	New Group 0/200000	Next	<b>K</b> Back	New Group	Crea
Image: Image	lgnitionNotif_bo	)t				
qwertyuiop asdfghjkl QzxCvbnmc3 123 space done 123 space retur	🗋 🕕 Ignit	tionNotifications		0	NotifGroup	
qwertyuiop asdfghjki crzxcvbnmc3 123 space done 123 space return						
qwertyuiop asdfghjkl ∂zxcvbnm. 123 space dome 123 space return					gnitionNotifications as no access to messages	
qwertyuiop asdfghjkl & zxcvbnm ⊗ 123 space done 123 space return						
qwertyuiop asdfghjkl & zxcvbnm 123 space done 123 space return						
qwertyuiop asdfghjkl & zxcvbnm 123 space done 123 space return						
qwertyuiop asdfghjkl &zxcvbnm 123 space done 123 space return						
qwertyuiop         asdfghjkl         Qxcvbnm         Qxcvbnm         123         space						
q       w       e       r       t       y       u       i       o         a       s       d       f       g       h       j       k       l         a       s       d       f       g       h       j       k       l         a       s       d       f       g       h       j       k       l         c       z       x       c       v       z       x       c       v       n       c         123       space       done       123       space       return						
asdfghjkl ☆ zxcvbnm 123 space done 123 space return	a w e	r t y u	i o p	qw	ertyu	i o
	a s d	fghj	k I	а	s d f g h	j k l
123 space done 123 space return	⇒ z x	c v b n	m 🗵	Ŷ	z x c v b	n m <
	123	space	done	123	space	return
	<u> </u>		.0,			

7. Open an internet browser, and type in the following URL:

https://api.telegram.org/bot{API TOKEN HERE}

Replace the {API TOKEN HERE} with your API token you received from BotFather, excluding the {} brackets. Your browser will return a list of JSON data. Look for the chat ID, which will be in the first two lines. A group chat ID will have a – as a prefix, that is how you know you have the correct ID.

This is your group chat ID.

8. Now that the API token and chat ID have been created, it can be utilized in an Ignition project.

### 3. Setup connection between Ignition and Telegram

To demonstrate how Ignition can send notifications to Telegram, a basic Ignition project with only a button component will be utilized.



The button component will act as the intermediary between an action in the Ignition project and the notification in Telegram. To enter the event script console of a component, double click on the component, and a screen similar to this will appear:

Component Scripting [Button]	]	$\frown$	>					
Event Handlers	< 🗌 Navigation 🛛 😻 Set Tag Value 🛢	SQL Update 🗉 Set Property 🗐 Script Editor						
action	> No Action							
actionPerformed								
focus	O Open / Swap							
key	Open 🔘	Pass Parameters						
mouse	🗹 and Center	Parameter Name Value	+					
mouseMotion	and Close This Window		音					
Custom Methods	Additional Instance							
-+ Add method	Additional Instance		e					
• Add memod	Swap							
	-							
	Window:							
	Main Window 🔻							
	F O Forward / Back							
	🔘 Go Back 🔘 Go Forward							
	Close							
	This Window	v 🔘 Other Window: Main Window 💌						
	c Action Qualifiers							
	Action Qualitiers							
	Security	Gonfirmation						
	× ×							
	Event Description	Event Object Properties						
	This event is fired when the 'action' of the	e component 💲						

The actionPerformed option on the left side panel will be automatically highlighted. This is the action event in which to code the connection. Click on the "Script Editor" tab and enter the following Python code:

```
url = "https://api.telegram.org/bot{0}/sendMessage?chat_id={1}&text={2}"
apiToken = "ENTER YOUR API TOKEN HERE BETWEEN DOUBLE QUOTATIONS"
chatID = "ENTER YOUR CHAT ID HERE BETWEEN DOUBLE QUOTATIONS"
chat = "Your message"
finalURL = "https://api.telegram.org/bot" + apiToken + "/sendMessage?chat_id=" + chatID
+ "&text=" chat
system.net.httpGet(finalURL, 10000, 60000)
```

Your code should look something like this:

<pre>Fivent Handlers  Fivent Handlers  CactionPerformed*  Set Tag Value SQL Update Set Property Cscript Editor  actionPerformed*  actionPe</pre>			
<pre>&gt; caction &gt;&gt;   url = "https://api.telegram.org/bot(0)/sendMessage?chat_id={1}&amp;text={2}" 2 apiToken = "apiTokenHere" 3 chatId = "chatIdHere 3 chatId = "chatIdHere 4 chat = "Hello World" 5 6 finalURL = "https://api.telegram.org/bot" + apiToken + "/sendMessage?chat_id=" + chatId + "&amp;text=" + chat 7 8 system.net.httpGet(finalURL, 10000, 60000) + Add method</pre>	👕 Event Handlers	Set Tag Value State Set Tag Value SQL Update Set Property Set Set Tag Value Set Property Set Tag Value Set Property Set Set Tag Value Set Value	
* Advanced Settings	Custom Maintens  Custom Methods  Add method	<pre>Navigation</pre>	
		Advanced Settings	

The url variable contains the API connection URL to the Telegram bot. This URL will be used in the finalURL variable where the apiToken, chatID and chat variables will be replacing {0}, {1} and {2} respectively.

Enter your API token in the double quotation marks next to the apiToken variable. Do the same with your chat Id next to the chatID variable.

For the chat variable, enter in double quotations your own message that you would like to send to your Telegram bot.

For the finalURL variable, replace all as shown above.

The GET request will be sent through the system.net.httpGet Python package. It needs the finalURL as the first parameter, the connect timeout as second parameter, and the read timeout as third parameter. Parameter descriptions to follow:

Parameters
String url
The URL to retrieve.
Integer connectTimeout
The timeout for connecting to the url. In millis. Default is 10,000.
Integer readTimeout
The read timeout for the get operation. In millis. Default is 60,000.

To send the notification to Telegram, click on the "Apply" and then the "OK" button. You will be receiving a notification via Telegram in a few seconds from clicking the button:

now

Once you open the notification, you will be directed to the group chat, where the bot received the message from Telegram and displays it as a message.

		al 🗟 🗖
Chats	NotifGroup 2 members	
	Uproad Mossagos	_
	Officau wessages	
IgnitionNoti	fications	
IgnitionNoti Hello Worl	fications d 09:30	
IgnitionNoti Hello Worl	fications d 09:30	00

### Error Handling

If you do run into some server request errors, here is a list of the potential problems that could be inside of your script:

#### 400 BAD\_REQUEST

The query contains errors. In the event that a request was created using a form and contains user generated data, the user should be notified that the data must be corrected before the query is repeated.

#### **Examples of Errors:**

- FIRSTNAME\_INVALID: The first name is invalid
- LASTNAME\_INVALID: The last name is invalid
- PHONE\_NUMBER\_INVALID: The phone number is invalid
- PHONE\_CODE\_HASH\_EMPTY: phone\_code\_hash is missing
- PHONE\_CODE\_EMPTY: phone\_code is missing
- PHONE\_CODE\_EXPIRED: The confirmation code has expired
- API\_ID\_INVALID: The api\_id/api\_hash combination is invalid
- PHONE\_NUMBER\_OCCUPIED: The phone number is already in use
- PHONE\_NUMBER\_UNOCCUPIED: The phone number is not yet being used
- USERS\_TOO\_FEW: Not enough users (to create a chat, for example)
- USERS\_TOO\_MUCH: The maximum number of users has been exceeded (to create a chat, for example)
- TYPE\_CONSTRUCTOR\_INVALID: The type constructor is invalid
- FILE\_PART\_INVALID: The file part number is invalid
- FILE\_PARTS\_INVALID: The number of file parts is invalid
- FILE\_PART\_X\_MISSING: Part X (where X is a number) of the file is missing from storage
- MD5\_CHECKSUM\_INVALID: The MD5 checksums do not match
- PHOTO\_INVALID\_DIMENSIONS: The photo dimensions are invalid
- FIELD\_NAME\_INVALID: The field with the name FIELD\_NAME is invalid
- FIELD\_NAME\_EMPTY: The field with the name FIELD\_NAME is missing
- MSG\_WAIT\_FAILED: A waiting call returned an error

#### **401 UNAUTHORIZED**

There was an unauthorized attempt to use functionality available only to authorized users.

#### **Examples of Errors:**

- AUTH\_KEY\_UNREGISTERED: The key is not registered in the system
- AUTH\_KEY\_INVALID: The key is invalid
- USER\_DEACTIVATED: The user has been deleted/deactivated
- SESSION\_REVOKED: The authorization has been invalidated, because of the user terminating all sessions
- SESSION\_EXPIRED: The authorization has expired
- AUTH\_KEY\_PERM\_EMPTY: The method is unavailble for temporary authorization key, not bound to permanent

#### **403 FORBIDDEN**

Privacy violation. For example, an attempt to write a message to someone who has blacklisted the current user.

#### 404 NOT\_FOUND

An attempt to invoke a non-existent object, such as a method.

#### 406 NOT\_ACCEPTABLE

Similar to 400 BAD\_REQUEST, but the app should not display any error messages to user in UI as a result of this response. The error message will be delivered via updateServiceNotification instead.

#### 420 FLOOD

The maximum allowed number of attempts to invoke the given method with the given input parameters has been exceeded. For example, in an attempt to request a large number of text messages (SMS) for the same phone number.

#### **Error Example:**

• FLOOD\_WAIT\_X: A wait of X seconds is required (where X is a number)

#### **500 INTERNAL**

An internal server error occurred while a request was being processed; for example, there was a disruption while accessing a database or file storage.

If a client receives a 500 error, or you believe this error should not have occurred, please collect as much information as possible about the query and error and send it to the developers.

#### **Other Error Codes**

If a server returns an error with a code other than the ones listed above, it may be considered the same as a 500 error and treated as an internal server error.

### 4. Alarm Event Trigger

The alarming feature in Ignition provides the functionality to set up and control alarms. Alarms can be configured on Tags, which will be demonstrated in this section. Notifications can be delivered to users when an alarm is triggered.

Alarm Notification Pipelines are drag and drop logic diagrams that you can configure to control how any of your projects' alarm notifications are sent to end users.

We will demonstrate how to create an alarm for a tag, create an alarm pipeline and connect the pipeline to the alarm in the example to follow.

1. First, a tag needs to exist on which the alarm can be configured. In this example an OPC tag is used. The tag Name, OPC Server and OPC Item Path is set up. A basic alarm has been added to trigger the alarm when the tag value is below or above a certain setpoint.

pe	rties				Details					
	12				Documentation	Diagnos	stics			
3	Basic Properties			^						
1	Name		OF	PC Alarm						
1	Fag Group		D	efault 🔻						
1	Enabled	true		Ŧ						
-	/alue									
-	/alue Source	OPC		*						
	DRC Server	Integ	Jenitian ODC UA C	-						
1	OPC Item Path	ns=1:s=[Gen9	im1 Meta:Random/Rai	erver *						
1	Numeric Properties	15-1,5-[0015	ing_wearandom/ra	idom y						
1	Deadband Mode	Abso	lute	-						
1	Deadband			0.0001						
	Scale Mode	Off		*						
	Engineering Units			*						
1	Engineering Low Limit			0.0						
	Engineering High Limit			100.0						
	Engineering Limit Mode	No_C	lamp	-						
	Format String		ŧ	<sup>t</sup> ,##0.##						
	Meta Data Properties			_						
	Documentation									
	Security									
1	Read Permissions			Public /						
i	Read Only	false	с.	w and the second						
1	Write Permissions			Public /						
	Scripting									
•	Fag Event Scripts		0 event s	cripts 🧷						
	Alarms									
1	Alarms		1 configured ala	rm(s) 🧷						
1	Alarm Eval Enabled	true		Ŧ						
1	History	1 272		~						
	later for blad									
							ок	Apply	Cancel	
		Tag Browser			a	×				
		+-0.0	default							
		~	ans	U	DT Definitions					
			Tag	0	Value					
		-								

The tag browser displays the tag together with a bell icon to represent the alarm function on the tag.

2. An On-call Roster and Alarm Notification Profile needs to exist on the Ignition Gateway before an Alarm Pipeline can be created. Here we have a basic admin On-call Roster and Email notification profile.

		true	Email Notification	Running	delete
Create new Alarm N	lotification Profile				
Test Pipelines and	Notification Profiles				
			-		
9 On-Call Rosters are	ordered lists of users that should be not	ified when an alarm occurs.	. These are used when configuring a	notification block inside an alarminį	3 pipeline.
P On-Call Rosters are	ordered lists of users that should be not Description	ified when an alarm occurs.	. These are used when configuring a Users	notification block inside an alarmin	g pipeline,
On-Call Rosters are Name Roster	ordered lists of users that should be not Description Test Roster	ified when an alarm occurs. # 1	. These are used when configuring a Users	notification block inside an alarmin	g pipeline. More 💌 edit

3. Back on the Ignition Designer, create an Alarm Notification Pipeline. In the Project Browser, right click on the Alarm Notification Pipeline option, and click on 'New Alarm Pipeline'.

- Filter			8	
- \Lambda Alarm Notifica	ition F	Pipelines		
👗 Sequential Fu	Ω	New Alarm Pipeli	ne	
Scripting	Î4	New Folder		
Transaction G	÷.	Paste		Ctrl+V
💽 Vision	<u>, m</u>	Export		
— 🛃 Client Even	ts			
<ul> <li>Windows</li> </ul>				
🖵 🗖 Main Wi	ndow	Þ		
- Root	Conta	iner		
□ĸ Bu	tton	P.		
Contraction Templates				
🗟 Named Querie	es			

4. A pipeline window with multiple drag and drop options will open up. Drag and drop a 'Notification' block and a 'Script' block onto the window. Connect the Start's arrow to the Notification block and connect the Notification block to the Script block.

🖌 Telegram - Perspective - Ignition Designer		- 🗆 X
File Edit View Project Tools Help		
	1	
roject Browser 🗇 _ 🗙	Pipeline Blocks	ē _ ×
Q- Filter		
Alarm Notification Pipelines	Notification Delay Splitter Switch Expression Set Property Jump Script	
Sequential Function Charts	1999	
Scripting  Perspective	START	
ag Browser		
- Q 💭 default 🔹 I -		
Tags UDT Definitions		
Tag Value	2	
	-	
peline Block Editor $\overrightarrow{a} = X$	1	
opout Condition		
Acknowledged		
Cleared 2		
Shelved	- -	
ipeline Enabled		
2 Enabled		
	C mm	>
	Pipeline 1 New Pipeline X	1
BlockDesignableContainer "New Pipelin	e" (0,0) [1549x340] Alt-drag to move.	484 / 4096 mb

5. In the Pipeline Block Editor under the Notification tab, choose your notification profile from the dropdown. Under the Contacts tab, choose your roster from the dropdown.

cation     Contacts     Consolidation       ne>     Image: Consolidation     Notification       ne>     Image: Consolidation       Image: Consolidation     Image: Consolidation	fication     Contacts     Consolidation       ine>     Image: Consolidation     Notification     Consolidation       ne>     Image: Consolidation     Consolidation     Image: Consolidation       ne>     Image: Consolidation     Consolidation     Image: Consolidation       ne>     Image: Consolidation     Image: Consolidation     Image: Con	tification Block —		- Notification Block
ne>  C Roster Type: Direct  On-Call Roster	nne>	otification Con	tacts Consolidation	Notification Contacts Consolidation
IN Notifications	ine> On-Call Roster il Notifications Roster	lone>	- C	Roster Type: Direct 🔹
il Notifications	nil Notifications	ne>		On-Call Roster
Hoster B		ail Notifications		Roster 🔻 📿

6. The pipeline diagram should look something like this:



7. Double click on the Script block and the Script editor will pop up. Here is where we add the code to send notifications to Telegram. You can adjust the code to send notifications to the users specified in the roster you created after sending them emails.



You can also configure your chat variable to send a specific message to the user, or any alarm event property. Remember to add your personalized apiToken and chatld to the respective variables. Save your script and project.

8. Now to connect your notification pipeline to your tag. Double click on the tag, open your tag Alarm options, and scroll down to the Notification option. Click on the dropdown next to 'Active Pipeline' and choose your pipeline. Click on 'Commit'.

Alarms Ignition Alarm - Outside Setpoints - Ou	Ignition Alarm - Outside Setpo	ints	
	+= +2		
	Ack Notes Required	false	* @
	Shelving Allowed	true	▼ GĐ
	<ul> <li>Alarm Mode Settings</li> </ul>		
	Mode	Outside Setpoints	*
	Low Setpoint		10 👄
	Low Inclusive	true	▼ ⊕
	High Setpoint		90 GD
	High Inclusive	true	▼ GĐ
	Any Change	false	▼ GĐ
	<ul> <li>Deadbands and Time Delays</li> </ul>		
	Deadband		0 👄
	Deadband Mode	Absolute	*
	Active delay (seconds)		0 👄
	Clear delay (seconds)		0 GĐ
	Notification		
	Ack Pipeline		▼ c∋
	Active Pipeline		
	Clear Pipeline	<none></none>	2
	Phone Call Settings	Telegram/Pipeline 1	
	Custom Message	relegiuminipenne i	GÐ
	<ul> <li>Email Notification Properties</li> </ul>		
	Custom Message		Ð
	Custom Subject		GÐ
	<ul> <li>SMS Notification Properties</li> </ul>		
	Custom Message		Θ,
	Active Pipeline	ass active events generated by this alarm	

9. Once your tag alarm has been triggered, you will receive a notification from Telegram with your specified message:



### 5. Summary

To send notifications from your Ignition project to Telegram, here is a summary of what you need to do:

- 1. Download and configure a bot in Telegram
- 2. Find the chat ID and API token in Telegram
- 3. Create a basic Ignition project / Launch an existing project
- 4. Set up the connection between Ignition and Telegram

For additional resourcing, please direct yourself to the Ignition user manual: https://docs.inductiveautomation.com/display/D0C80/Welcome



+27(0)115958458 information@element8.co.za

element8.co.za

#### in Element8. Know your Limitless. The Pivot. Block C. Montecasino Boulevard. Fourways. Gauteng. 2055. South Africa. Authorised Distributor of Industry Leading Software: Ignition, Canary & Flow Software J