



# eagle.io

## **eagle.io documentation** *2023*

**Bentley Systems, Incorporated**

**May 30, 2023**



---

## Contents

---

<b>1</b>	<b>Getting Started</b>	<b>1</b>
1.1	Introducing eagle.io ( <a href="https://eagle.io">https://eagle.io</a> )	2
1.2	Tutorials	10
<b>2</b>	<b>Using eagle.io</b>	<b>13</b>
2.1	Workspaces Tree	13
2.2	Content Views	20
2.3	Nodes and Configuration	49
2.4	User Preferences	166
2.5	Alarms	170
2.6	Notifications	179
2.7	Operate and Control	182
2.8	Historic Data	188
2.9	Sharing and Security	220
2.10	Processing and Logic	223
2.11	Reports	234
2.12	Account Settings	264
2.13	Messages	286
2.14	Device Configuration	289
2.15	Troubleshooting	431
<b>3</b>	<b>HTTP API</b>	<b>433</b>
3.1	Quick Start	433
3.2	Best Practices	440
3.3	API Resources	442
<b>4</b>	<b>Reference</b>	<b>521</b>
4.1	Configuration	521
4.2	Expressions	523
4.3	Historic Data	569
4.4	Time	581
<b>5</b>	<b>Frequently Asked Questions</b>	<b>587</b>

5.1	General . . . . .	587
5.2	Data Loggers . . . . .	588
5.3	Files . . . . .	593
5.4	Cameras . . . . .	594
<b>6</b>	<b>Glossary</b>	<b>595</b>
	<b>Index</b>	<b>597</b>





# CHAPTER 1

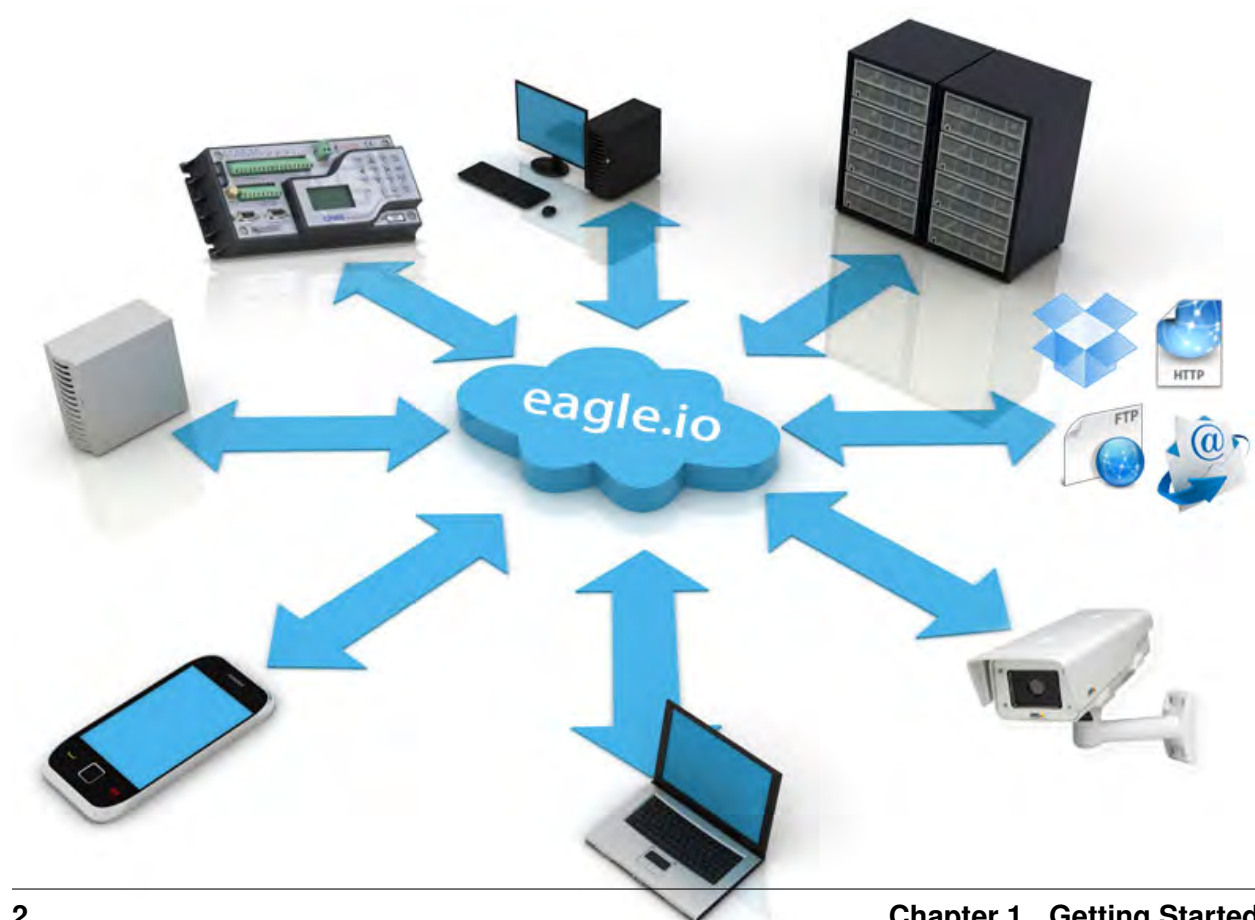
---

## Getting Started

---

### 1.1 Introducing eagle.io (<https://eagle.io>)

#### 1.1.1 Connect, configure and control your monitoring fleet from the cloud



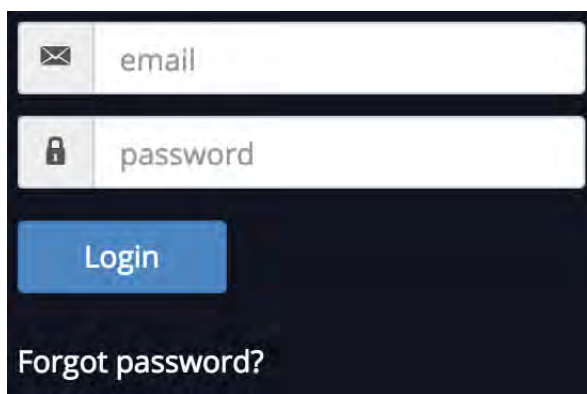
Introducing [eagle.io](https://eagle.io) (<https://eagle.io>), the secure, reliable, easy to use cloud-based data monitoring service. Whether you're working with simple data files or managing a fleet of *data loggers*, eagle.io has you covered.

With real-time communications, collaboration and instant updates built into the core eagle.io has everything you've always wanted and more. The lightning fast interface has been designed to work across all modern phones and computing devices with no software to install and no page reloads.

### 1.1.2 Log in to eagle.io

You will need a user profile to log in. This requires you to [sign up](https://eagle.io/trial) (<https://eagle.io/trial>) for an account or receive a *Workspace Sharing* (page 220) invitation from a registered user.

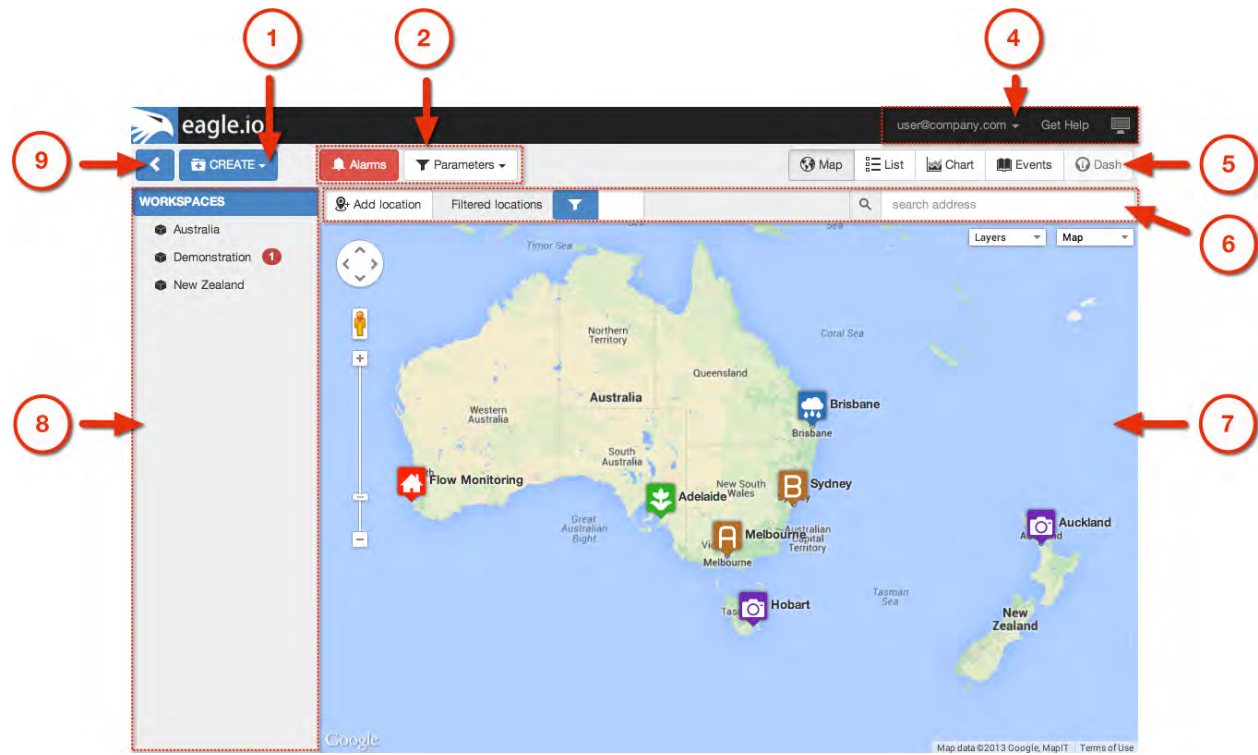
*Login form*

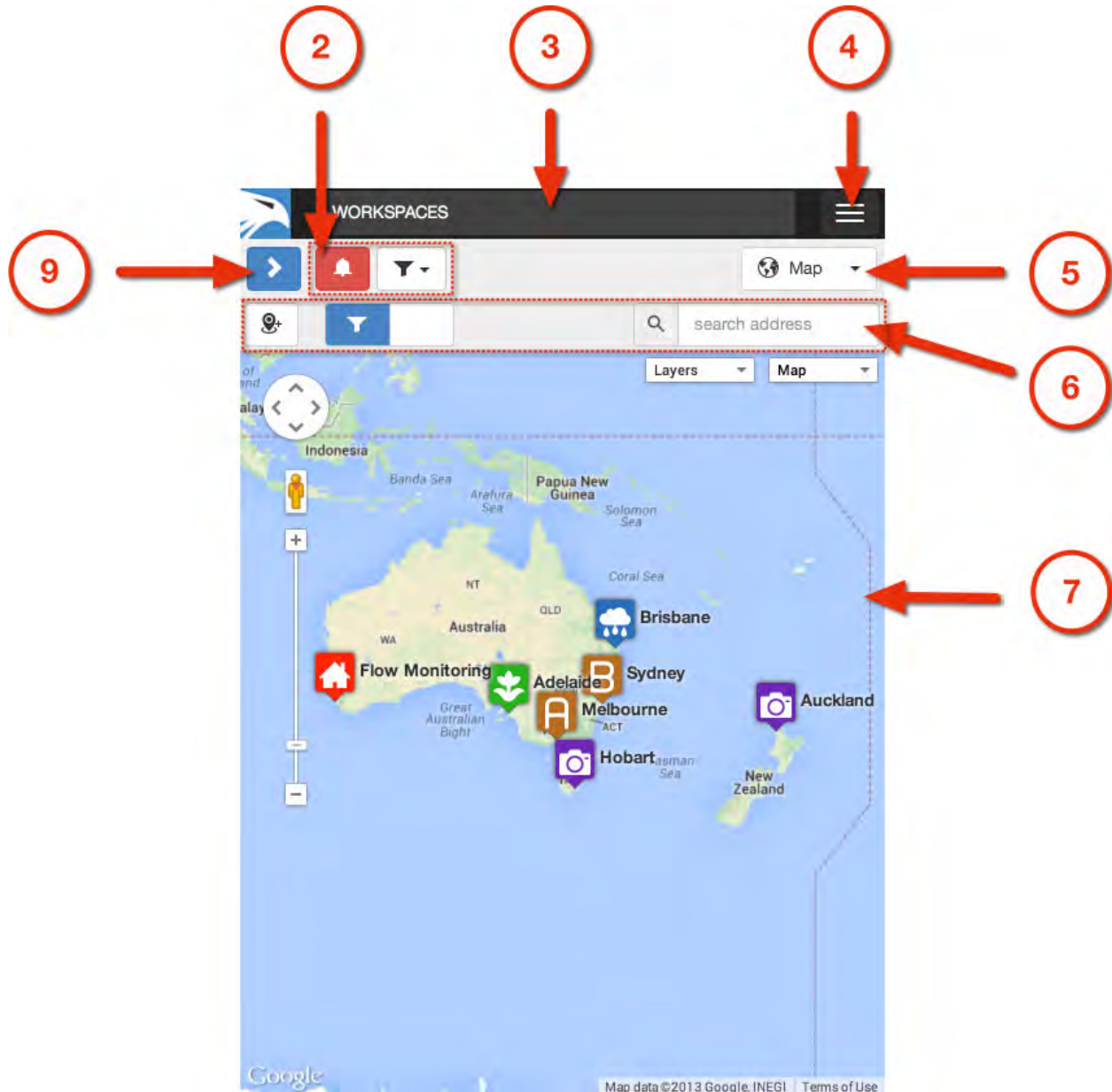
A screenshot of the eagle.io login interface. It features a dark background with two white input fields. The first field is labeled 'email' with an envelope icon on the left. The second field is labeled 'password' with a lock icon on the left. Below these fields is a blue 'Login' button. At the bottom left, there is a link that says 'Forgot password?'.

Enter your *email* address & *password* to login. If you have forgotten your password, simply click the *forgot password* link to reset it.

### 1.1.3 Layout and Navigation

The user interface is divided into sections which group similar functionality and content. The layout of these sections will vary depending on the screen size of your device including its orientation (portrait or landscape). The figures below highlight the Main features of the interface as displayed on different devices.

*Desktop layout**Phone and Tablet layout*



### Main features

1. Create Menu
2. Filters
3. Status Bar
4. User Menu, Messages, Help and Fullscreen mode
5. View Selector
6. Toolbar
7. Content Area
8. Workspaces Tree

## 9. Workspaces Tree show/hide button

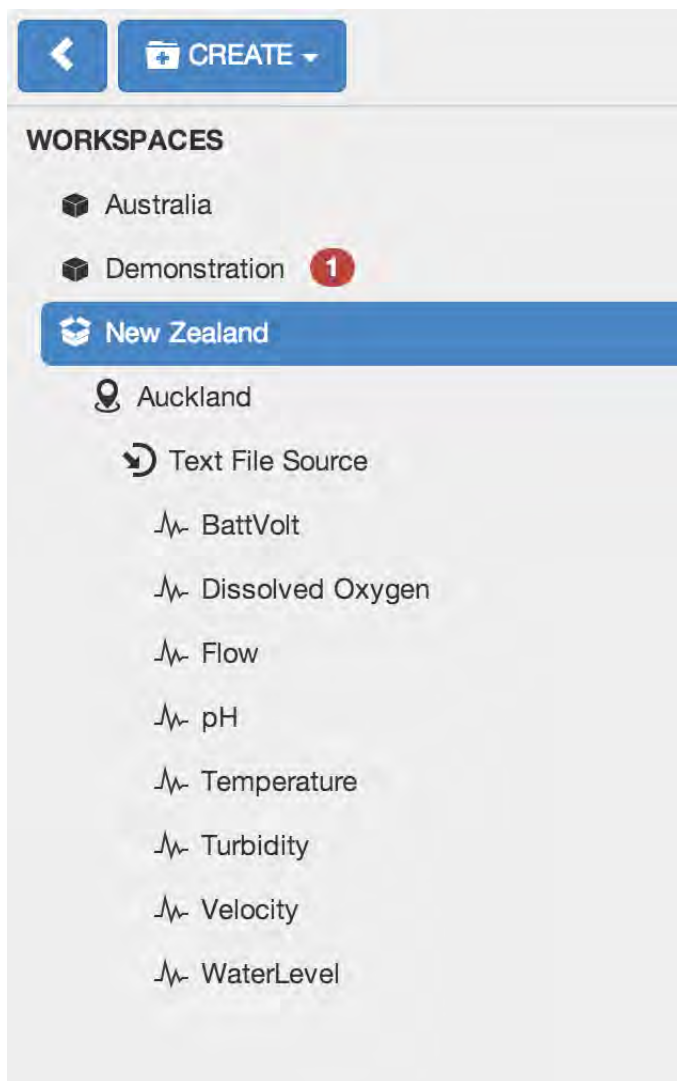
### Selecting Nodes in the Workspaces Tree

The Workspaces Tree displays all the Workspaces and associated content you have access to. Each item in this menu is called a Node. The highlighted item is referred to as the Selected Node.

Most Nodes are hierarchical and may also contain Nodes themselves. The *Workspaces* label is the best example of this as its referred to as the Base Node and contains all the Workspace Nodes.

If you have access to more than one Workspace you may notice some of the Workspaces are *GRAYED* out when you first login. Select the Workspace to load its assets.

*Example Workspaces Tree showing hierarchy*



Change the Selected Node by clicking or touching on another Node. You will notice that container Nodes will automatically expand and collapse to show and hide their contained Nodes on selection.

Selecting a Node focuses your View. Only the selected Node and the Nodes it contains will be available in the Content Area.

You can easily show all content by selecting the Base Node.

---

**Note:** The Workspaces Tree will be hidden by default when displayed on small screens. You can see the Selected Node displayed in the Status Bar. Double-click or Double-tap the status bar to quickly reveal the Workspaces Tree. Alternatively you can use the Workspaces Tree show/hide button.

---

*Most common types of Nodes:*



**Workspace** Workspaces are special container Nodes that can be shared with other users. All other Nodes are created within the Workspace. Workspaces can only be created by the Account Owner or Administrators.



**Folder** Folders (*previously Groups*) are simple nodes that provide structure to your Workspaces. You may choose to group related content into folders by geography or related purpose. Folders can be created inside Workspaces, Locations and other Folders.



**Location** Locations are special containers used to associate data with a geographical location. Locations can be created inside Workspaces and Groups.



**Chart** Create various types of charts with customized functionality and appearance including historic charts with multiple y-axes and wind rose charts with configurable ranges.



**Dashboard** Dashboards allow users to create customized views of their data in visually appealing layouts using animated gauges, charts, lists, maps and other graphical controls.



**Data Source** Data Sources automatically acquire or receive timeseries data using a variety of different transport options. Connect to a [data logger](#) or collect data from files. Data Sources can be created inside Locations only.



**Parameter** Parameters are managed by and contained within Data Sources. Parameters store the individual timeseries acquired by its Data Source. There are a few different types of parameters but essentially they support Monitoring or Controlling of values.

Review the full list of [Node types](#) (page 49).

## Changing the View

The Selected Node (including any Nodes it contains) can be displayed in a variety of Views. Use the View Selector to change the View. You will notice the Content Area and Toolbar change.

### Content Views



**Map** Map provides a fully featured interactive interface for displaying location based data.



**List** Intuitive categorised lists provide quick access to the data for your most important Nodes: Locations, Sources, Parameters, Attachments.



**Chart** Use the Chart View for reviewing your timeseries data in highly interactive charts. Choose to group data on charts by Location or Parameter type. You can also setup custom charts for full customization over functionality and appearance.



**Events** Every interaction (user and system generated) is recorded in the Events log. Use the Events View to quickly audit the system, a user or history of alarms.



**Dash** The Dash is for viewing the customized graphical dashboards and also provides an overview of all dashboard nodes available based on the Selected Node in the Workspaces Tree.

### User Menu

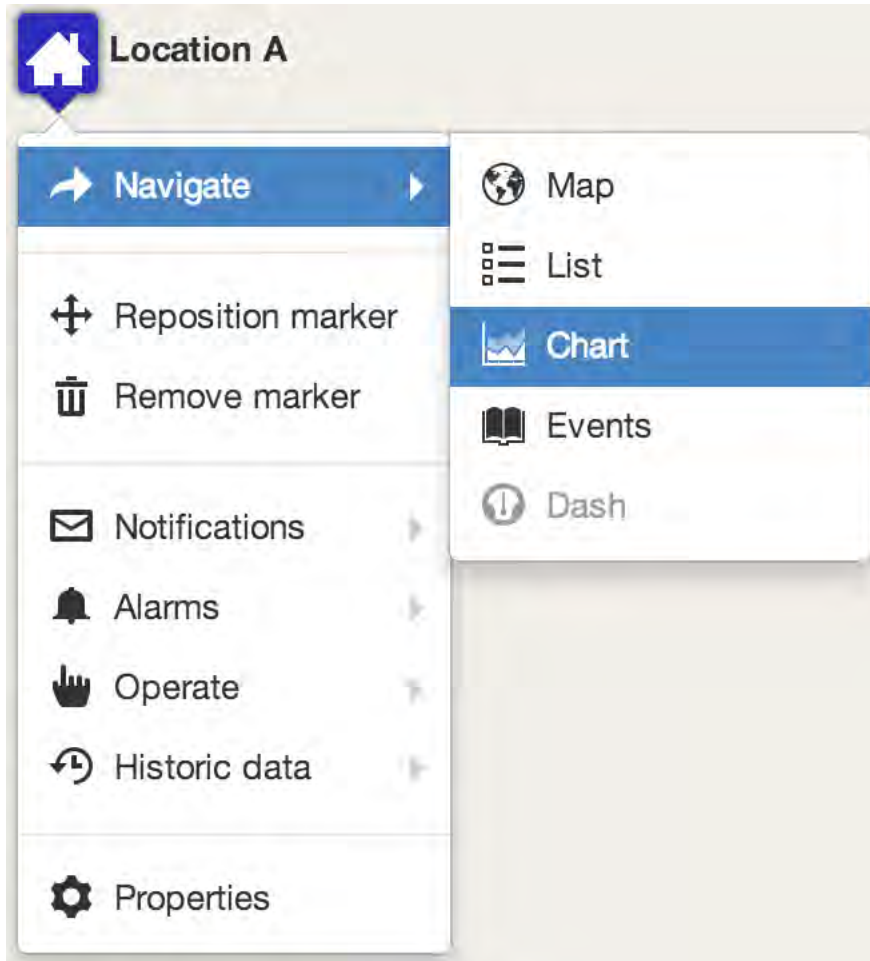
The User Menu provides quick access to the User Profile, Account Settings, Log out and Help.

### Context Menus

Context menus are built into the interface and provide quick access to navigation, operations and configuration for their associated elements. Right-click or long-press on a Node in the Workspaces Tree or an item in the Content Area (ie. Map marker, List item, etc.) to display it's associated context menu.

*Example Context menu shown for Map marker*






---

**Note:** Context menus display only the items relevant to the associated Node or content the menu is displayed for. Context menu items are also dependant on User permissions. If there are no available options then no context menu will be displayed.

---

### Click, Touch, Tap, What?

The user interface has been designed to work seamlessly across all modern computing devices including desktop computers, tablets and phones which means it has built-in support for Touch devices.

*Throughout this documentation you will notice the following terms:*

**Click, Touch, Tap** Use a mouse to select an item or touch the screen (touch device only) momentarily where the item is displayed.

**Double-click, Double-tap** Click the item twice in quick succession with a mouse, or Tap the item twice with a touch device.

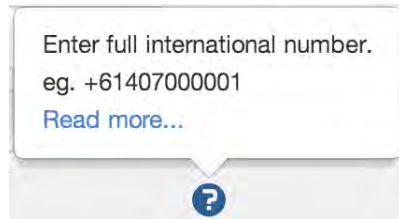
**Right-click, Long-press** Using a dual button mouse, use the alternate button to Click. When using a single button mouse or touch device you need to click (or touch) and hold without releasing for a few seconds.

### 1.1.4 Inline help



Help is available throughout the interface - wherever you see the help icon. You can hover, click or touch the help icon to show the help popup - many of which have quick links back to the relevant section in this documentation.

*Inline help example*



### 1.1.5 Offline use

An internet connection is required. If the connection to eagle.io has failed a disconnected notice will be displayed. Normal operation will resume once the connection has been re-established.

**Disconnected.** Trying to contact server...

---

**Note:** You may notice the app refresh after the connection is re-established after a long timeout or when updates have been released.

---

## 1.2 Tutorials

### 1.2.1 Basics of Navigation

Watch this tutorial at: <https://www.youtube.com/watch?v=EiRfV6kzhSw>

### 1.2.2 Create a Text File Source

Watch this tutorial at: <https://www.youtube.com/watch?v=1TeD9AneNI8>

### 1.2.3 Create a Dashboard

Watch this tutorial at: <https://www.youtube.com/watch?v=2r6DazwRd6M>

### 1.2.4 Chart Customization

Watch this tutorial at: <https://www.youtube.com/watch?v=Q39nSTJfcKM>



Introductions to all the key parts of [eagle.io](https://eagle.io) (<https://eagle.io>) you'll need to know:

## 2.1 Workspaces Tree

The Workspaces Tree is the place to go for organising, configuring and managing your Nodes. This sections provides an overview of the common Workspaces Tree functions. Depending on the Selected Node type and your security permissions, not all options may be available.

---

**Note:** Most operations require you to display the Context Menu. Right-click or long-press on a Node in the Workspace Menu to display its Context Menu.

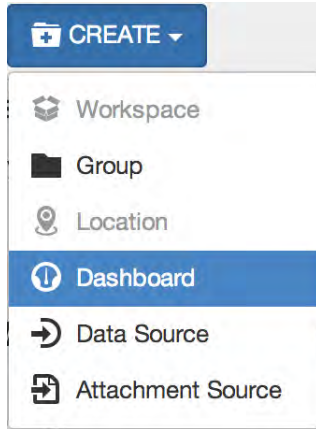
---

### 2.1.1 Create

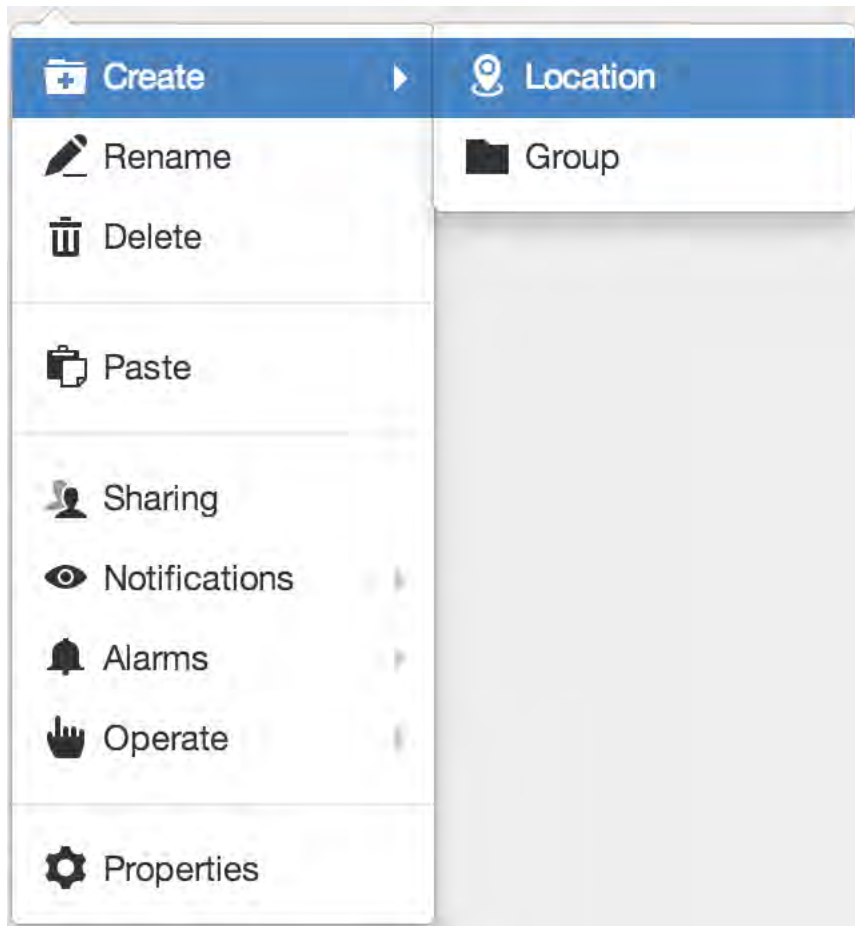


New Nodes can be created using the *Create Menu* or from the *Context Menu*.

*Create Menu*



The Create Menu displays a list of the types of Nodes that can be created inside the currently Selected Node. Items that are disabled can not be created inside the Selected Node - either a restriction on the type of Node or you do not have the required security permissions.

*Create using Context Menu*

The Context Menu has a Create submenu which offers the same options as the Create Menu.

### 2.1.2 Rename



Rename a Node from the context menu or press the *F2* key after selecting a Node.



The Node will be selected and you can type in a new name and press enter/return or click anywhere outside the Node to apply.

---

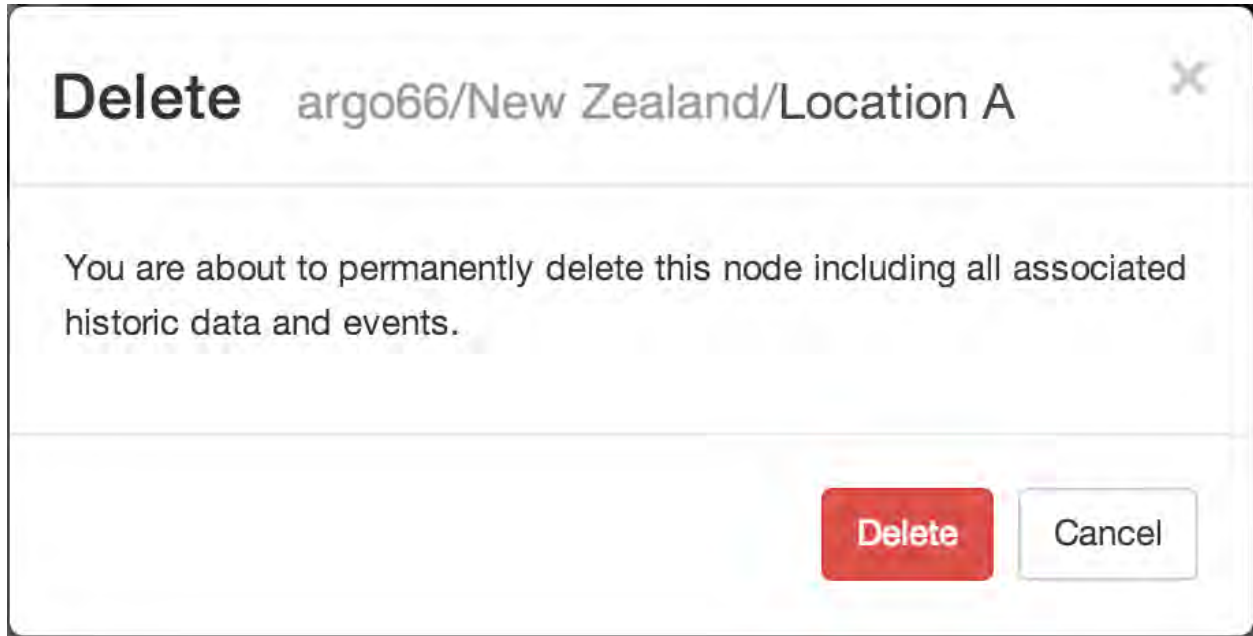
**Note:** Nodes must be uniquely named within the current container Node and must not exceed 120 characters in length. Restricted characters include forward slash '/', underscore '\_', semicolon ';' and curly braces '{' and '}'.

---

### 2.1.3 Delete



Permanently delete a Node using the Context menu or press the *delete* key after selecting the Node. A confirmation message will be displayed.



---

**Note:** Deleting a Node will also delete any of the Nodes it contains including any associated historic data, events, attachments and configuration. This operation is NOT reversible.

---

### 2.1.4 Copy and Paste

Copy (*Ctrl+C*) and Paste (*Ctrl+V*) is the quickest and easiest way to duplicate a Node including all Nodes it contains. You must have configuration permission for the Node you intend to copy and the destination Node where you intend to paste.



Copy a Node



Paste a Node anywhere in the Workspaces Tree that accepts the copied Node type.

---

**Note:** The copy operation is done at the time of pasting. The original Node being copied must exist or the paste operation will be aborted.

---



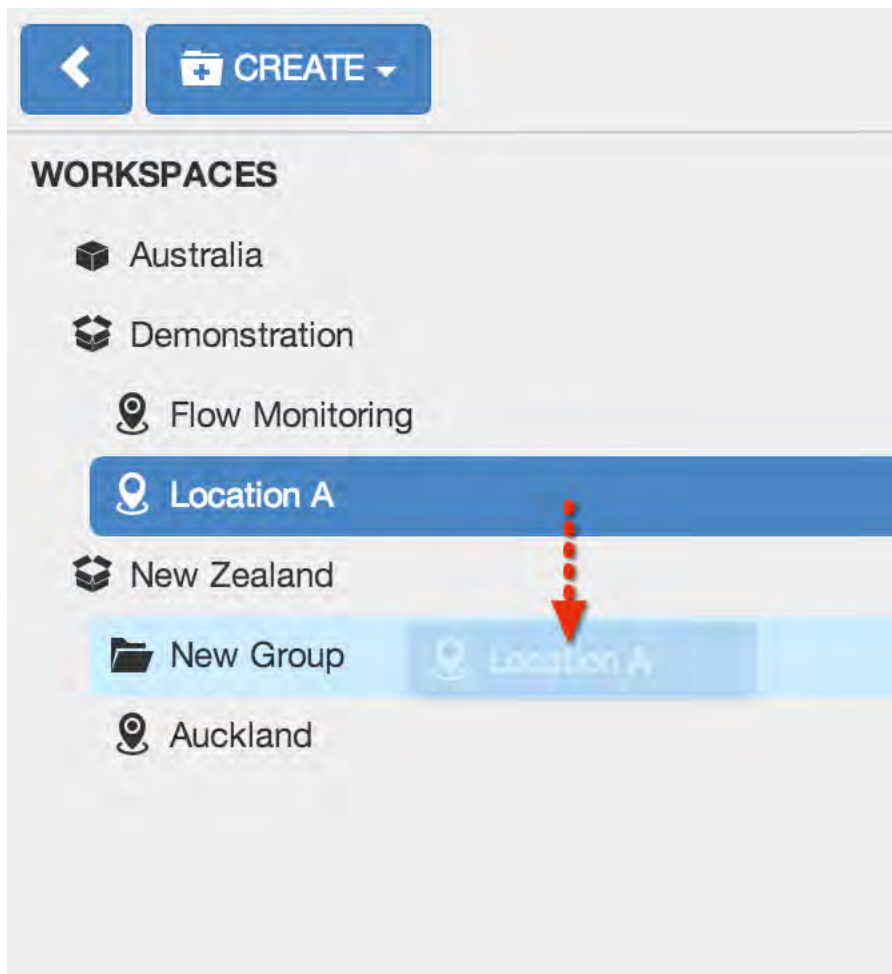
### 2.1.5 Move



Move a Node (including all Nodes it contains) within a Workspace or between Workspaces. You can also move Workspaces between accounts that you have permission to access.

Select *Move* from the Context Menu to open the *Move dialog* and select a destination node for the move operation.

Alternatively *Click and Drag* a Node in the Workspaces Tree to the destination Node and release the mouse button when you see the destination node highlighted. Nodes will automatically expand when hovered.

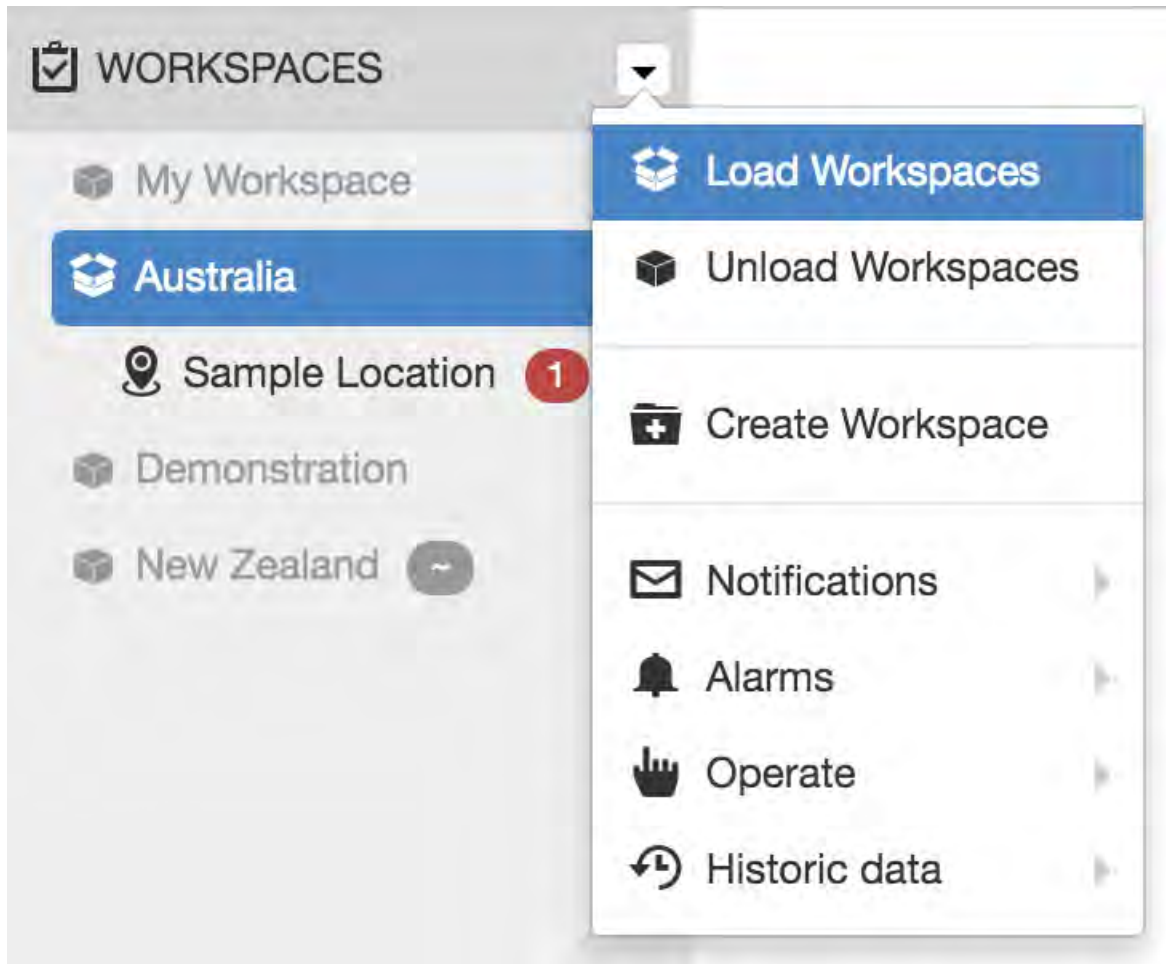


### 2.1.6 Workspace Loading On-demand

If you have access to more than one Workspace you may notice some of the Workspaces are *GRAYED* out in the Workspaces Tree when you first login. Click on the Workspace or select *Load Workspace* from the

Workspace context menu to load its assets.

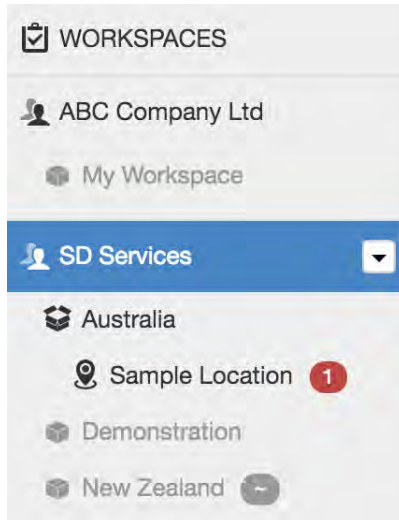
You can load all Workspaces by selecting *Load Workspaces* from the top-level *WORKSPACES* context menu.



### 2.1.7 Workspace Grouping by Account

If you have access to more than one account you will notice Workspaces are automatically grouped under the associated account in the Workspaces Tree.

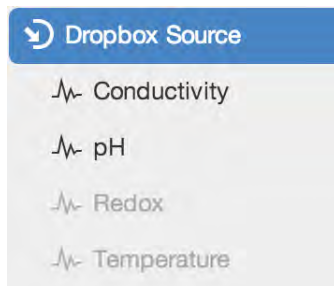
Selecting an account will display content related to all *Loaded Workspaces* you have access to for that account.



### 2.1.8 Disabled Nodes

You may notice *GREY* Nodes within a Workspace.

Sources being configured for the first time will be shown as disabled until they have been fully configured. Parameters that had been created and then unchecked (or unavailable) for collection will also be shown as disabled in the Workspaces Tree, List and Map Views.



## 2.2 Content Views



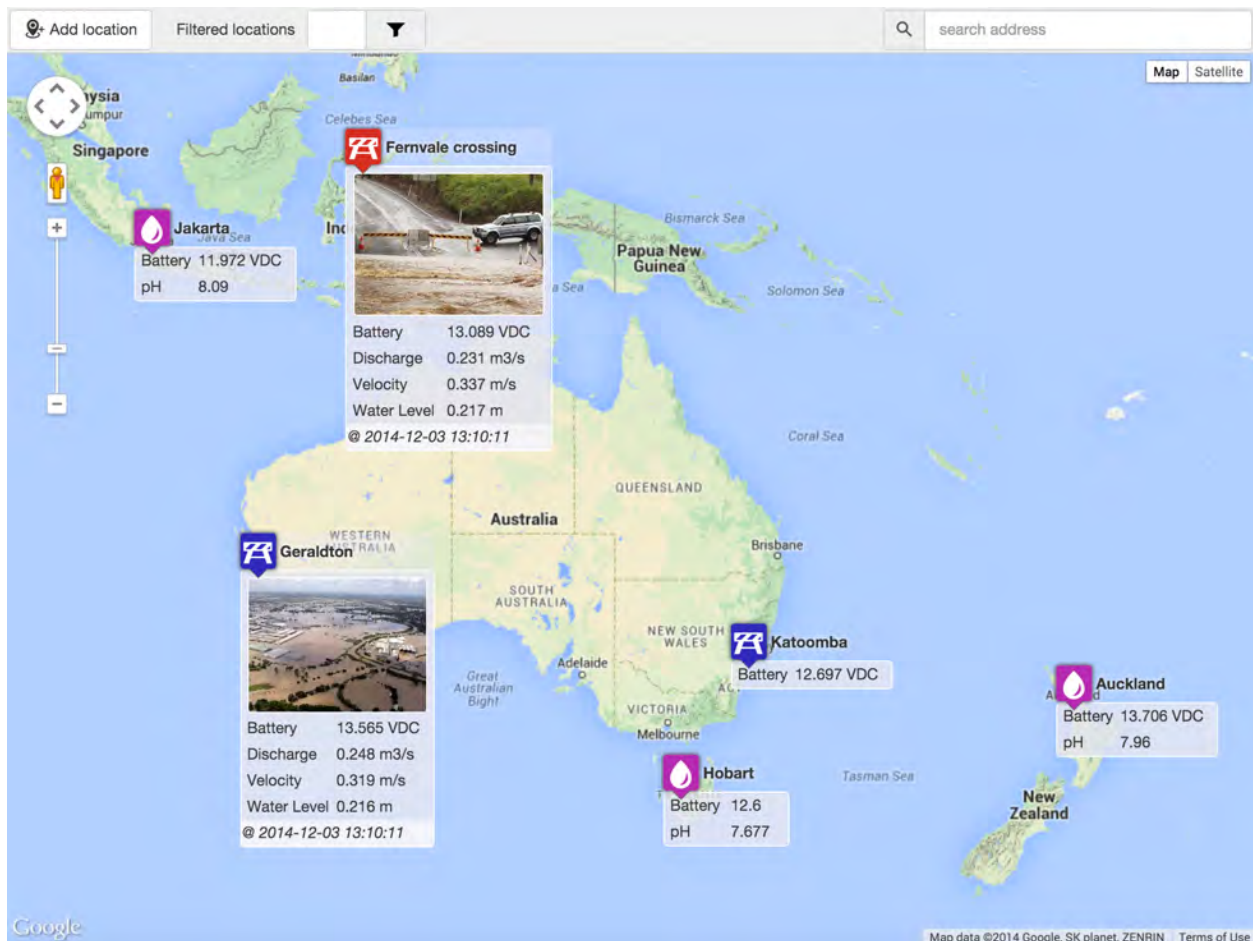
Overview and Guide to using the available Content Views:

### 2.2.1 Map



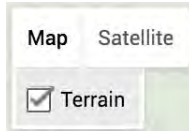
Map provides a fully featured interactive interface for displaying location based data.

**Note:** The view displays content based on the Selected Node in the Workspaces Tree. Use filters to quickly target specific parameters or review alarms.



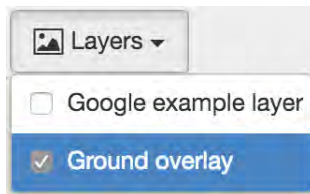
### Map styles

Use the style controls in the top-left corner of the map to change the map appearance. Toggle between Map, Satellite and Terrain styles.

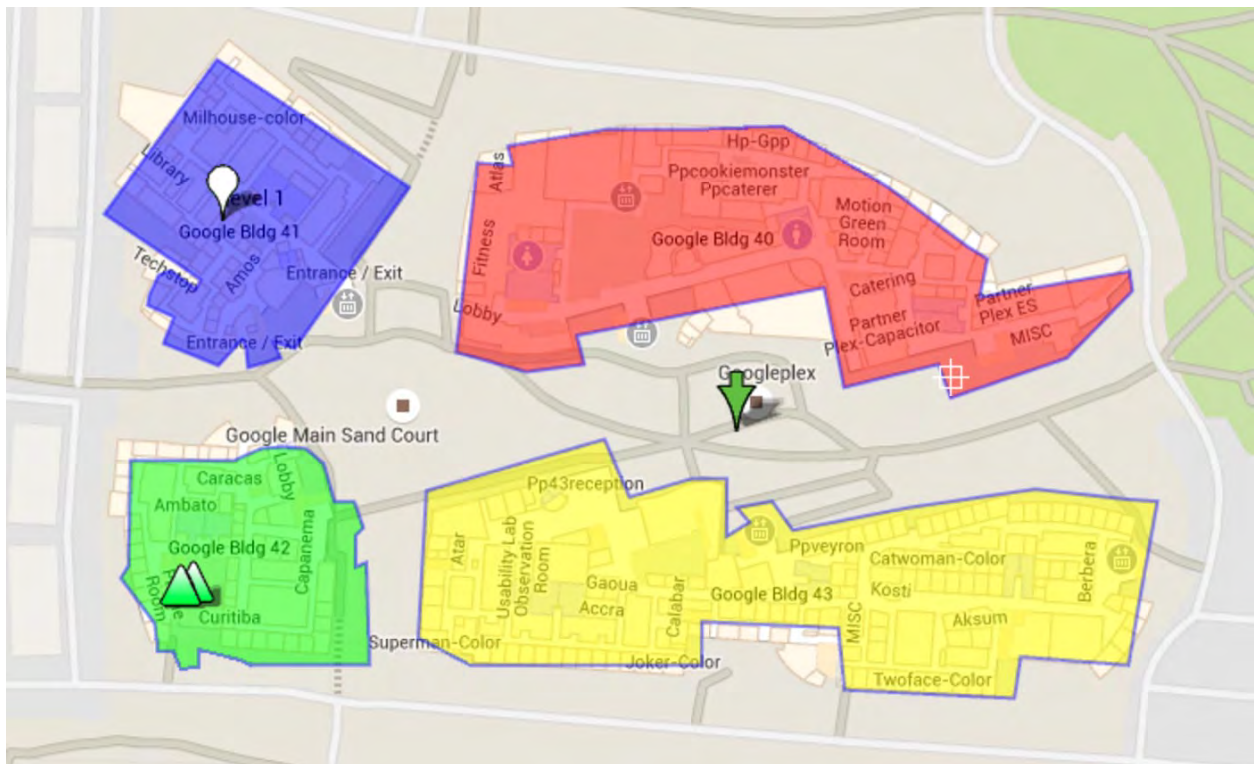


## Map layers

Use the Map layers toolbar menu to quickly show or hide KML (and KMZ) files that have been associated with Workspace(s). The menu will display layers based on your current selection in the *Workspaces Tree* (page 13). Refer to *Workspaces configuration* (page 52) for help with adding or removing map layers.



Example KML layer displayed on map:



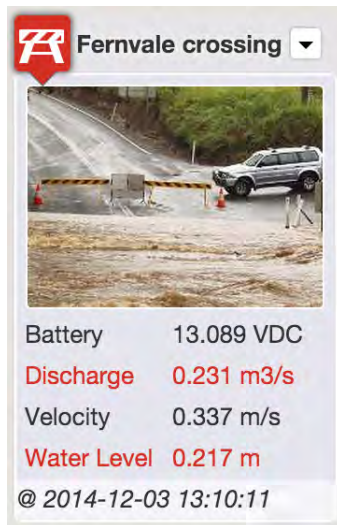
**Note:** A maximum of 10 unique layers can be shown on the map at any one time. Additional layers can be shown only after another layer has been hidden. Remote layers are automatically refreshed every 5 minutes.

## Zoom and Pan

Use the standard Google Map controls to Zoom/Pan or enter Street View. You can also use your mouse wheel or pinch gestures to zoom. The appearance and layout of the map controls many vary depending on the type of device you are using.

## Click and Hover

Hover over a Location icon with a mouse to quickly show its parameters. Click or Touch the Location to pin the parameters list open or closed.



## Filters

Use filters to quickly show locations of interest.

**Alarms Filter** will show only the Locations within the Selected Node that are currently in alarm or acknowledged state (including any contained Nodes).

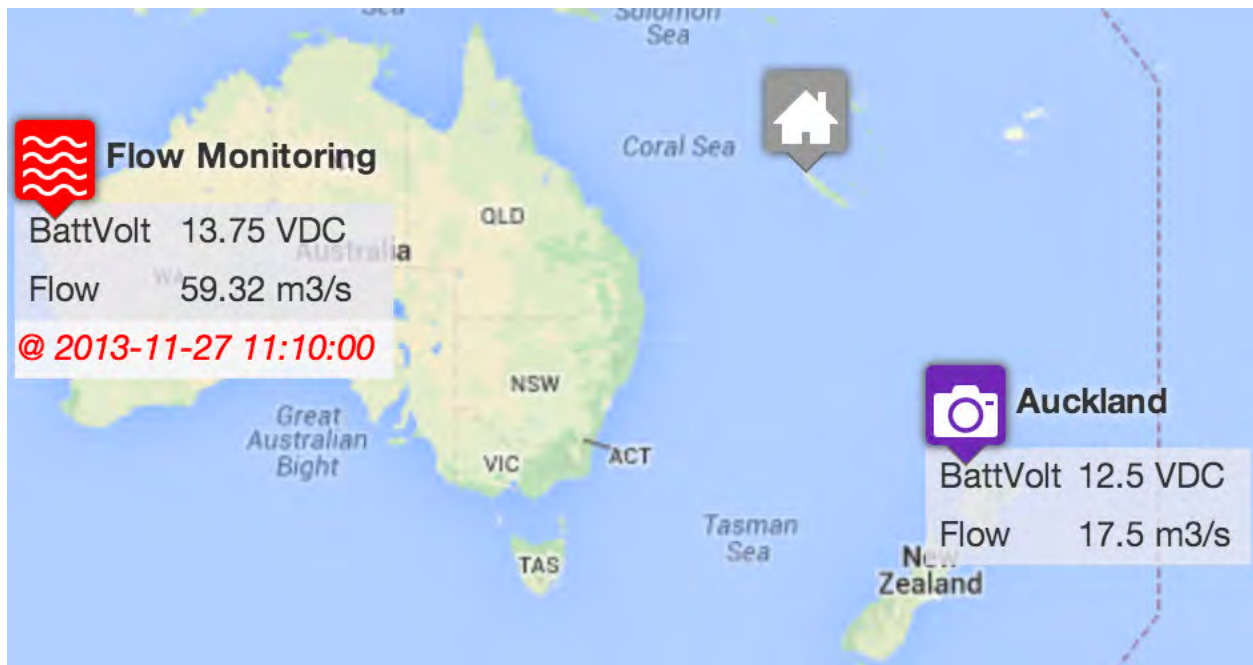
**Parameter Filters** will show only the Locations within the Selected Node that contain parameters matching the filter name(s). Both Alarms and Parameter Filters can be combined.



The show/hide filtered locations button found in the toolbar can be used to toggle the display of filtered locations. When active you will notice any filters locations are displayed with just a *GREY* marker and no label. You can still hover or click these filtered locations to display the parameters list.





*Map with filters applied*

When parameter filters are applied, the corresponding parameters for each applicable Location are automatically displayed. Hover or Click the locations to display all parameters.

Locations containing alarms will have their marker displayed in *RED*. If all alarms within a location have been acknowledged the marker will be displayed in *ORANGE*.

### Reserved marker colors

These reserved marker colors temporarily overwrite the normal marker color:

*Active Alarm*



*Acknowledged Alarm*



*Filtered*

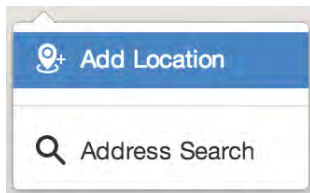
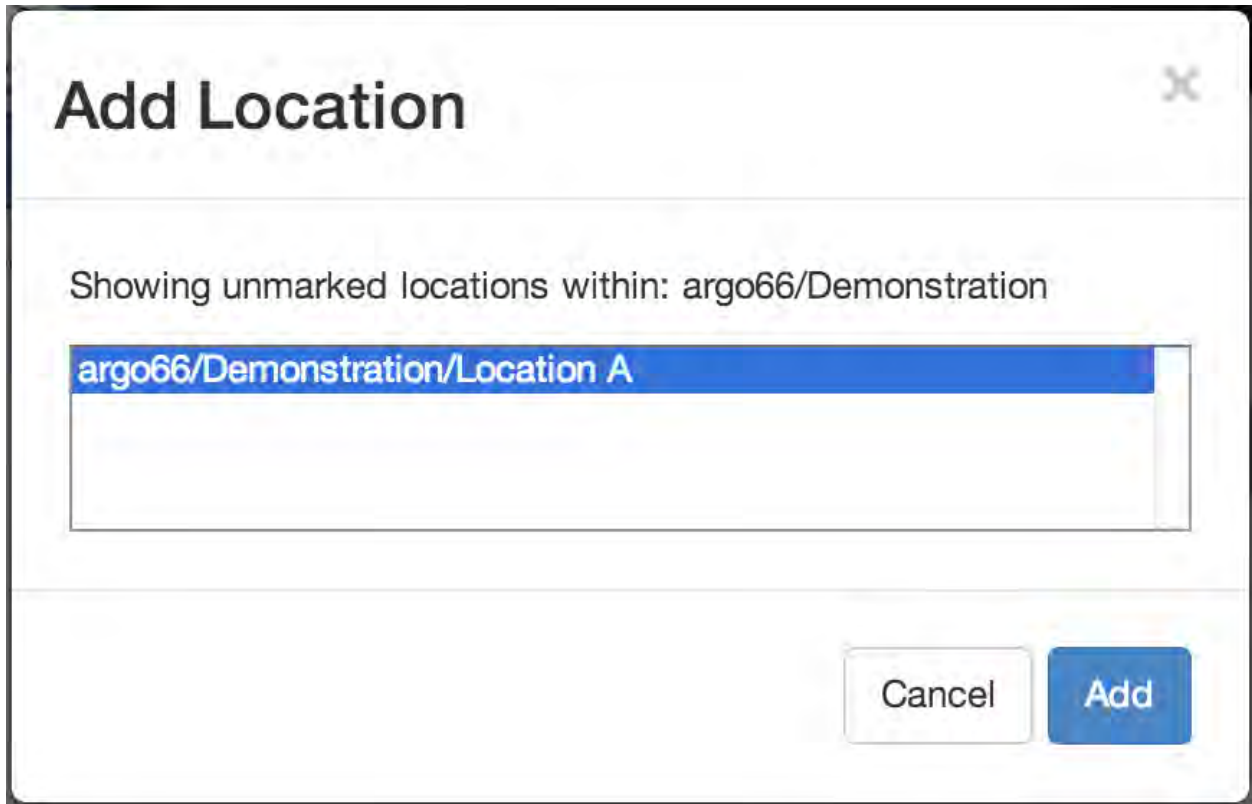


## Adding a new Location marker

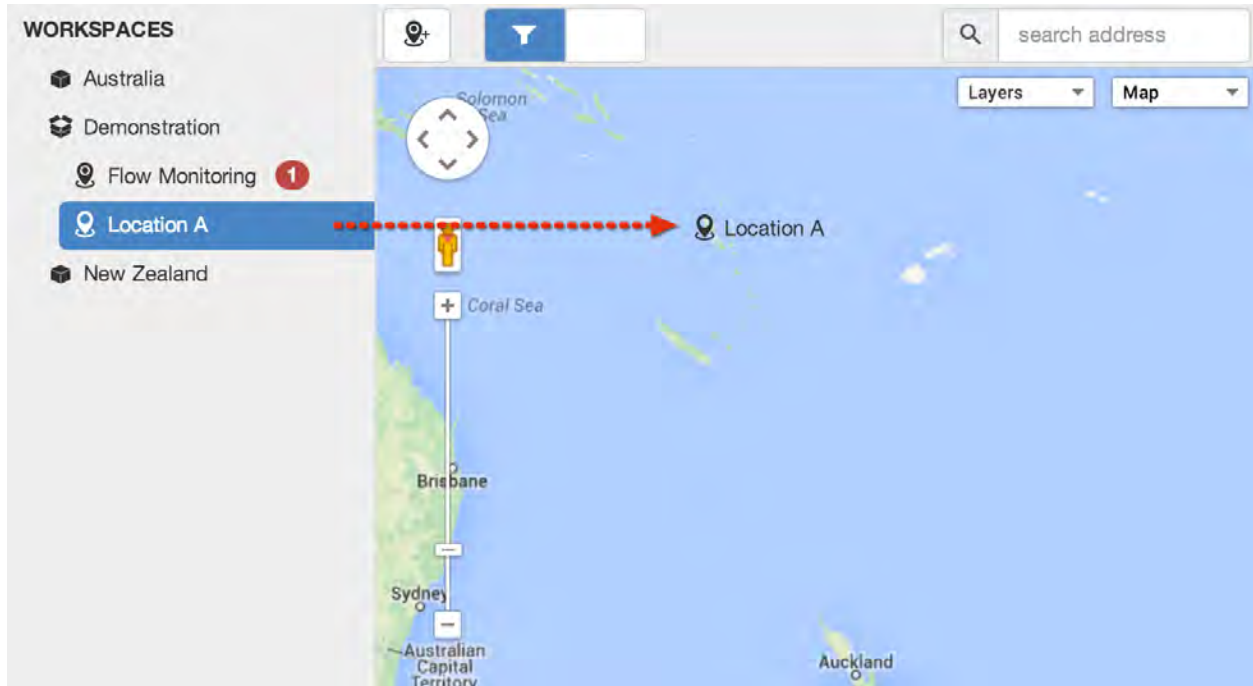


Use the *Add Location* button in the toolbar or from the map context menu with a right-click or long-press on the map. The Add Location dialog will be displayed showing all locations within the Selected Node that have not been assigned coordinates.



*Map context menu**Add Location dialog*

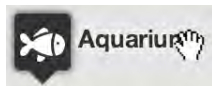
*Drag and Drop* Locations directly from the Workspaces Tree (not supported on touch devices).



## Reposition marker



Select *Reposition marker* from the map marker context menu to enter reposition mode. The marker will be shown with a bouncing effect while in reposition mode. Drag and drop the marker and pan/zoom the map until the marker is in the correct position. click or touch on the map background to apply the new coordinates.



## Remove marker



Select *Remove marker* from the map marker context menu to remove the marker from the map. This operation resets the location coordinates only. It will not delete the Location Node or any historic data.

---

**Note:** The Location must use Fixed Coordinates and you must have *configure* permission for the Workspace to Add, Reposition or Remove the marker. Locations with Dynamic Coordinates will automatically be added, repositioned and removed as the Source receives updates.

---

## 2.2.2 List



Intuitive categorised lists provide quick access to the data for your most important Nodes: Locations, Sources, Parameters and Attachments.

**Note:** The view displays content based on the Selected Node in the Workspaces Tree. Use filters to quickly target specific parameters or review alarms.

Sort rows in the list by clicking on column headers.

List rows are temporarily highlighted immediately following an update. Rows will be shown with a *RED* background to indicate active alarms on the Node or one of its contained Nodes. Rows with acknowledged alarms will be shown with an *ORANGE* background. Rows will be shown in *GREY* if the corresponding node configuration has not been finalised (Sources) or the parameter is not linked to the Data Source for automatic collection (Parameters).

### Active Alarm

<input type="checkbox"/>	argo66/Demonstration/Aquarium	Dropbox Source	pH	7.82	LOW	2014-01-09 11:57:49			
--------------------------	-------------------------------	----------------	----	------	-----	---------------------	--	--	--

### Acknowledged Alarm

<input type="checkbox"/>	argo66/Demonstration/Aquarium	Dropbox Source	pH	7.82	LOW	2014-01-09 12:05:49			
--------------------------	-------------------------------	----------------	----	------	-----	---------------------	--	--	--

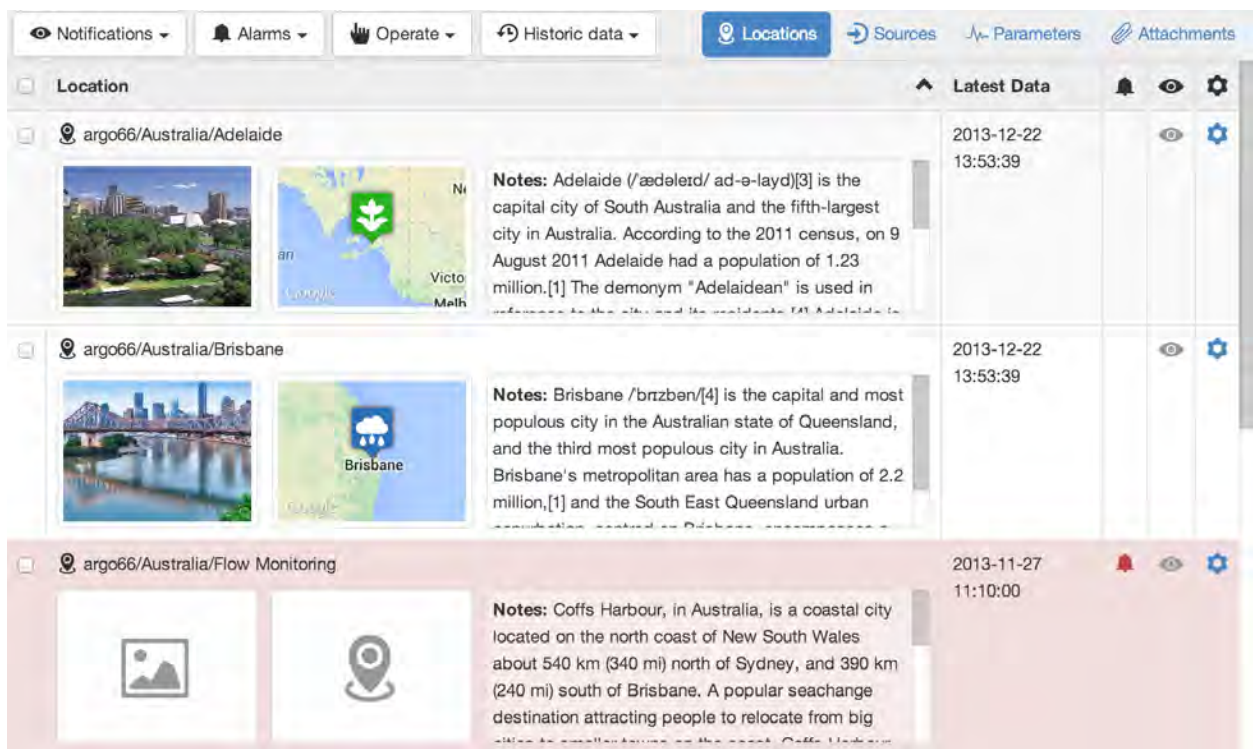
*Disabled*



## Subviews

### Locations

Locations Subview displays an overview of Location Nodes.



**Location** Location name with path, including a preview image, map thumbnail and any configured location metadata.

**Latest Data** Timestamp of the most recent data from all Sources contained within the Location.

**Alarms** An alarm icon will be shown if there are any active or acknowledged alarms on or within the Location. Click or touch the alarm icon to display the Alarm Panel.

**Subscribe** Click or touch the subscribe icon to subscribe to the Nodes you would like to receive notifications for. You must have *subscribe notifications* permission for the Workspace. If the Location contains Sources, the Subscription Panel will be displayed so you can toggle your subscription preferences individually or all together.

**Properties** Show the Node properties dialog.

## Sources

Sources Subview displays a list of all Data and Attachment Sources.

Location	Source	Status	Latest Data	Collected
argo66/Australia/Adelaide	Text File Source	SCHEDULED	2013-12-22 13:53:39	6
argo66/Australia/Brisbane	Text File Source	SCHEDULED	2013-12-22 13:53:39	6
argo66/Australia/Flow Monitoring	CR200 Scout	FAILED	2013-11-27 11:10:00	0
argo66/Australia/Hobart	Text File Source	UPDATING	2013-12-22 13:48:37	11
argo66/Australia/Melbourne	Text File Source	SCHEDULED	2013-12-22 13:53:39	6
argo66/Australia/Sydney	Text File Source	SCHEDULED	2013-12-22 13:16:21	59
argo66/New Zealand/Auckland	Text File Source	SCHEDULED	2013-12-22 13:48:37	11

**Location** Location name with path.

**Source** Name of Source including relative path within Location

**Status** Current status of the Source:

**DISABLED** The Source has been disabled by a user and will not be updated.

**IDLE** The Source is enabled for communication but has no collection scheduled.

**SCHEDULED** The Source is enabled for communication and has collection scheduled.

**QUEUED** The Source is queued for communication.

**CONNECTING** Attempting to connect to the Source.

**CONTROLLING** Control operations are being performed on the connected Source.

**DOWNLOADING** Data is currently being downloaded from the connected Source.

**RETRYING** Acquisition, Control or Configure operations have failed so a retry attempt has been scheduled.

**FAILED** Acquisition, Control or Configure operations have failed (after retry attempts exceeded).

**PROCESSING** Data is being processed after being downloaded from the connected Source.

**CONFIGURING** Sending configuration to the Source.

**REMOTE** Remote access is enabled and a client is connected. Acquisition, Control and Configure operations will not be available during this time.

**Latest Data** Timestamp of the most recent data from the Source.

**Collected** Displays number of records collected for a Data Source during last collection or number of files processed for an Attachment Source.

**Alarms** An alarm icon will be shown if there are any active or acknowledged alarms on or within the Source. Click or touch the alarm icon to display the Alarm Panel.

**Subscribe** Click or touch the subscribe icon to subscribe to the Nodes you would like to receive notifications for. You must have *subscribe notifications* permission for the Workspace. If the Source contains parameters, a subscription panel will be displayed so you can toggle your subscription preferences individually or all together.

**Properties** Show the Node properties dialog.

## Parameters

Parameters Subview displays a list of all parameters.

<div> <div>Notifications</div> <div>Alarms</div> <div>Operate</div> <div>Historic data</div> <div>Locations</div> <div>Sources</div> <div>Parameters</div> <div>Attachments</div> </div>							
Location	Source	Parameter	Value	State	Latest Data		
argo66/Demonstration/Aquarium	Dropbox Source	Δ- Conductivity	42.2 mS	NORMAL	2014-01-12 17:27:52		
argo66/Demonstration/Aquarium	Dropbox Source	Δ- pH	7.8	LOW	2014-01-12 17:27:52		
argo66/Demonstration/Aquarium	Dropbox Source	Δ- Redox	-374 mV	LOW	2014-01-12 17:27:52		
argo66/Demonstration/Aquarium	Dropbox Source	Δ- Temperature	28.6 °C	NORMAL	2014-01-12 17:27:52		

**Location** Location name with path.

**Source** Name of Source including relative path within Location.

**Parameter** Name of parameter.

**Value** Value of the parameter at the time indicated by the Latest Data column.

**State** State associated with current parameter value. An indicator will be displayed if the parameter has non-default states configured.

**Latest Data** Timestamp of the most recent data for the parameter.

**Alarms** An alarm icon will be shown if there are any active or acknowledged alarms for the parameter. Click or touch the alarm icon to display the Alarm Panel.

**Subscribe** Click or touch the subscribe icon to subscribe to the Nodes you would like to receive notifications for. You must have *subscribe notifications* permission for the Workspace.

**Properties** Show the Node properties dialog.

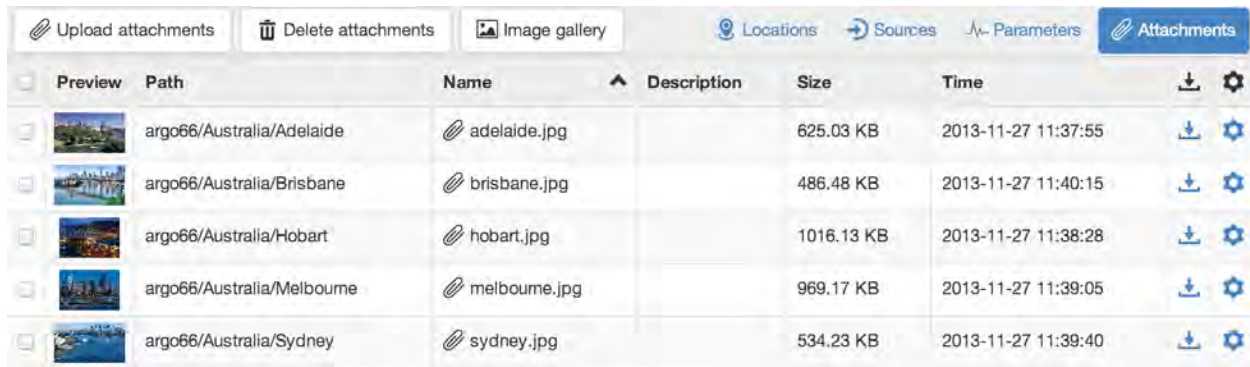


Control parameters can be controlled inline in the parameters list. Click the control icon and set the new value or cancel. Requires *operate* permission for the Workspace. See [Operate and Control](#) (page 182) for full details.



## Attachments

Attachments subview displays a list of all file *attachments* (page 105).



Preview	Path	Name	Description	Size	Time		
	argo66/Australia/Adelaide	adelaide.jpg		625.03 KB	2013-11-27 11:37:55		
	argo66/Australia/Brisbane	brisbane.jpg		486.48 KB	2013-11-27 11:40:15		
	argo66/Australia/Hobart	hobart.jpg		1016.13 KB	2013-11-27 11:38:28		
	argo66/Australia/Melbourne	melbourne.jpg		969.17 KB	2013-11-27 11:39:05		
	argo66/Australia/Sydney	sydney.jpg		534.23 KB	2013-11-27 11:39:40		

**Preview** Thumbnail image or file icon. Images can be clicked or touched to be shown in the media gallery.

**Container** Path of the container Node where the attachment is located.

**Name** Name of the attachment.

**Description** User defined description of attachment.

**Size** Size of the attachment.

**Time** Time of the attachment upload.

**Download** Download the attachment.

**Properties** Show the Node properties dialog.

---

**Note:** You must have the *read attachments* permission for the Workspace to see attachments in the list.

---

## Toolbar Controls

Toolbar controls should be used in combination with the row selection checkboxes to perform the selected operation on multiple Nodes at once. The same options are also available via the context menu for each row item.



## Notifications



Subscribe or Unsubscribe from receiving *notifications* (page 179) for the checked Nodes including all contained Nodes.

## Alarms



Acknowledge alarms with or without a comment, or Clear the alarms for the checked Nodes. Refer the section on *Alarms* (page 170) for more info.

## Operate



Acquire or Control the checked nodes. See *Operate and Control* (page 182) for more info.

## Historic Data



View data as a Table of values or graphically on a Chart. You can also export the historic data for the checked Nodes. See *Historic Data* (page 188) for full details.

## Delete



Permanently delete the selected Nodes. This operation is not reversible!

## Upload files



Upload files as attachments of the selected Workspace Node. You can also drag and drop new attachments directly onto any of the content views. Requires configure permission. Ensure the selected Workspace Node accepts attachments.

## Media Gallery



Displays images and supported video attachments in the gallery view. If no attachments are checked then all attachments will be available in the gallery.



### 2.2.3 Chart



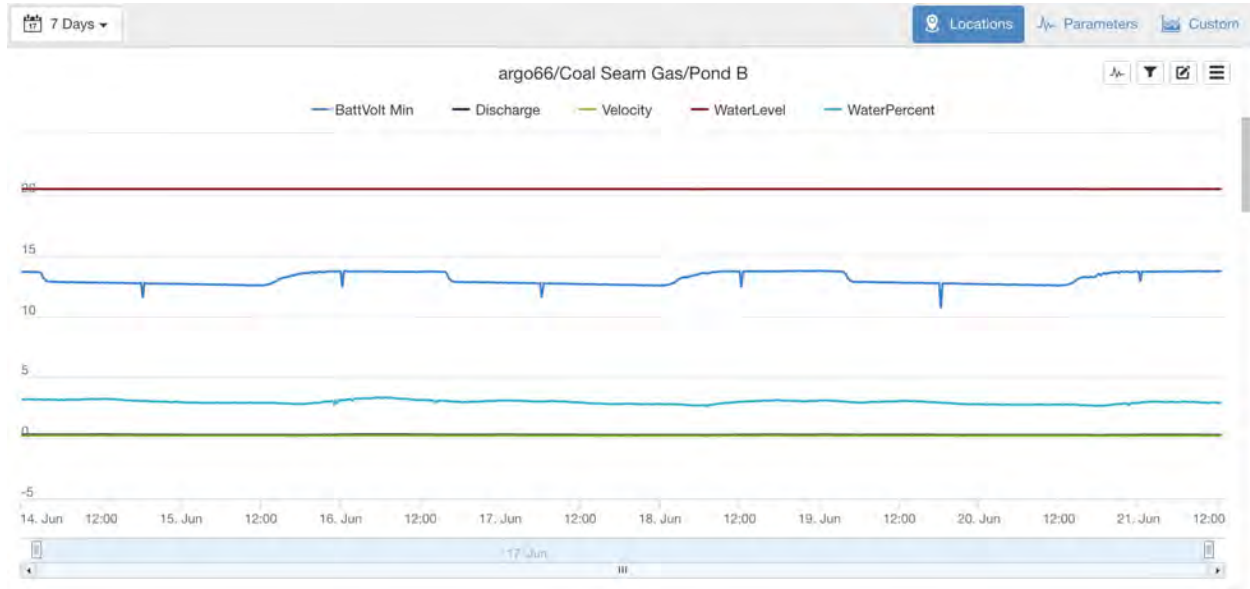
Use the Chart View for reviewing your timeseries data as highly interactive charts. Choose to group data on charts by Location or Parameter type. Charts are displayed using the settings configured for each individual parameter. See [Parameter configuration](#) (page 147) for full details.

**Note:** The view displays content based on the Selected Node in the Workspaces Tree. Use filters to quickly target specific parameters or review alarms.

### Subviews

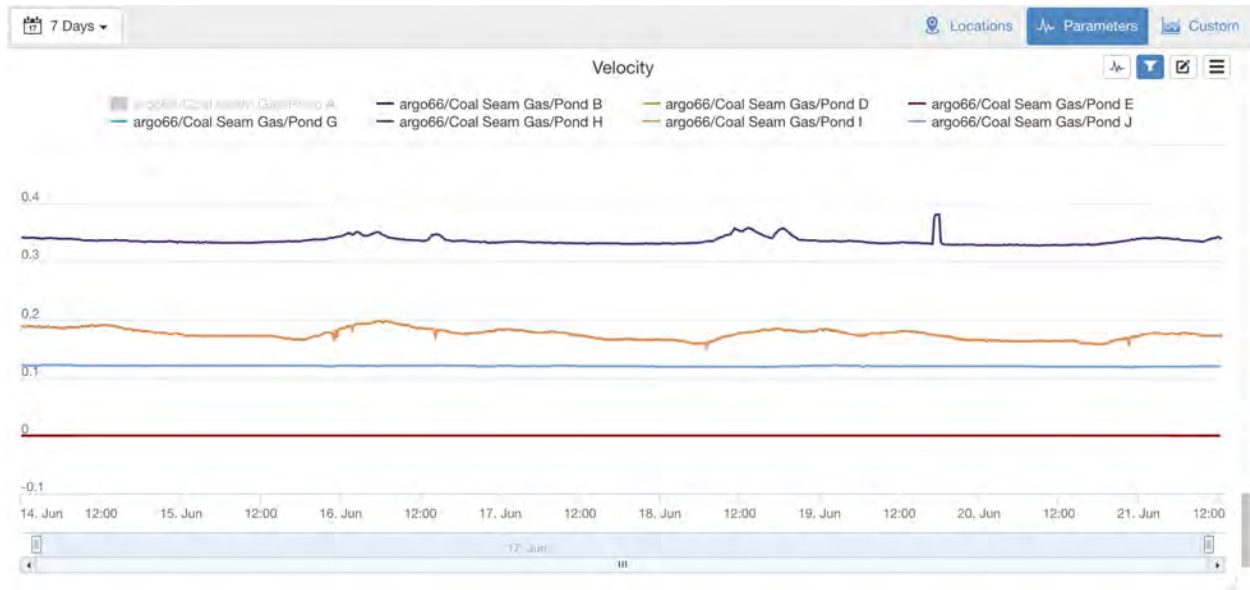
#### Locations

Locations Subview displays a chart per Location. Any Parameters contained within the Location will be displayed on the same chart. If a Location does not contain any Parameters (or the Parameters have been filtered), no chart will be displayed.



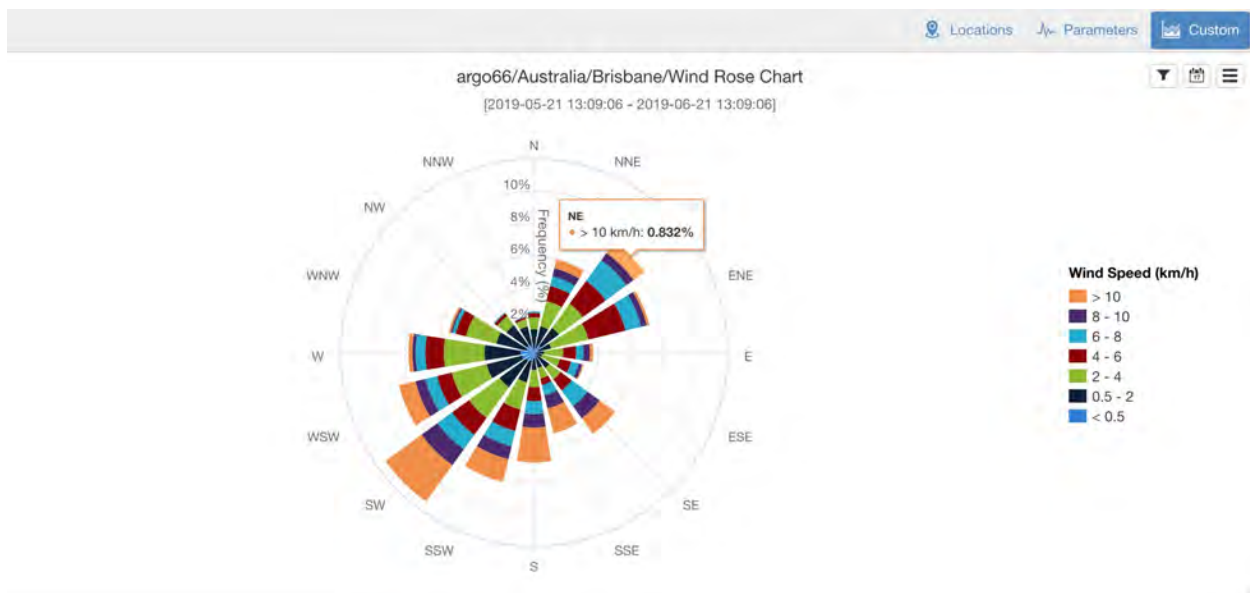
#### Parameters

Parameters Subview displays a chart per Parameter (based on Parameter name). Useful for comparing Parameters of the same type across multiple Locations. Filters can be applied to reduce the number of charts and/or Parameters being displayed.



## Custom

Custom Subview displays a list of pre-configured *custom charts* (page 66).



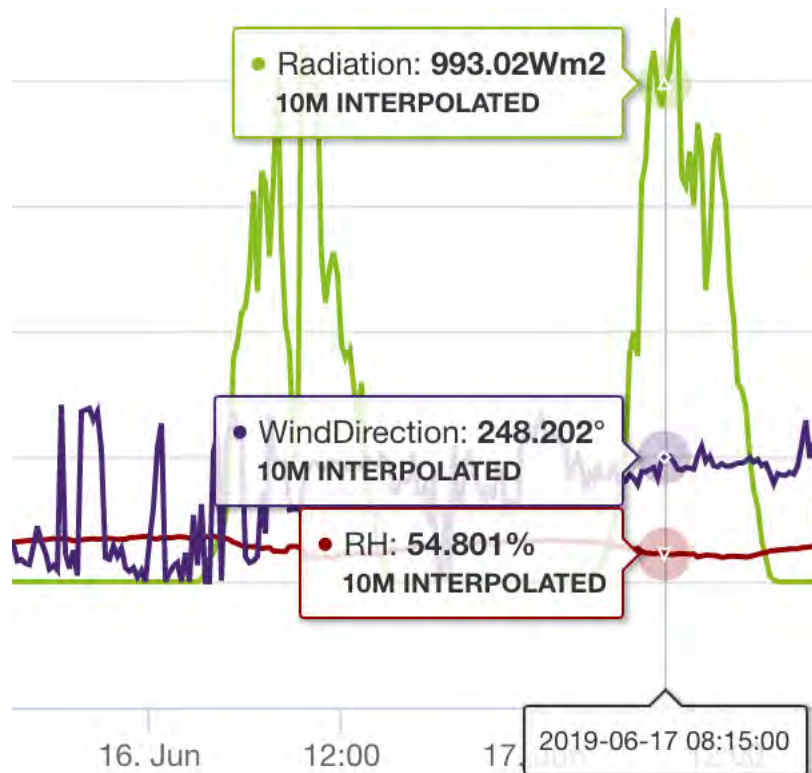
## Working with charts

### Parameter Values

Hover over a chart with the mouse cursor or touch part of the display area of a chart to display the value of the Parameters corresponding with the time on the x-axis.

Parameter values may be shown with aggregation applied - indicated with a duration and aggregate name. Aggregation is based on the date range of the chart data and the Parameter configuration settings. See [Parameter configuration](#) (page 147) for full details.

If you have assigned [quality codes](#) (page 577) to the data, the associated quality code and color is displayed on hover.



### Adjust Height

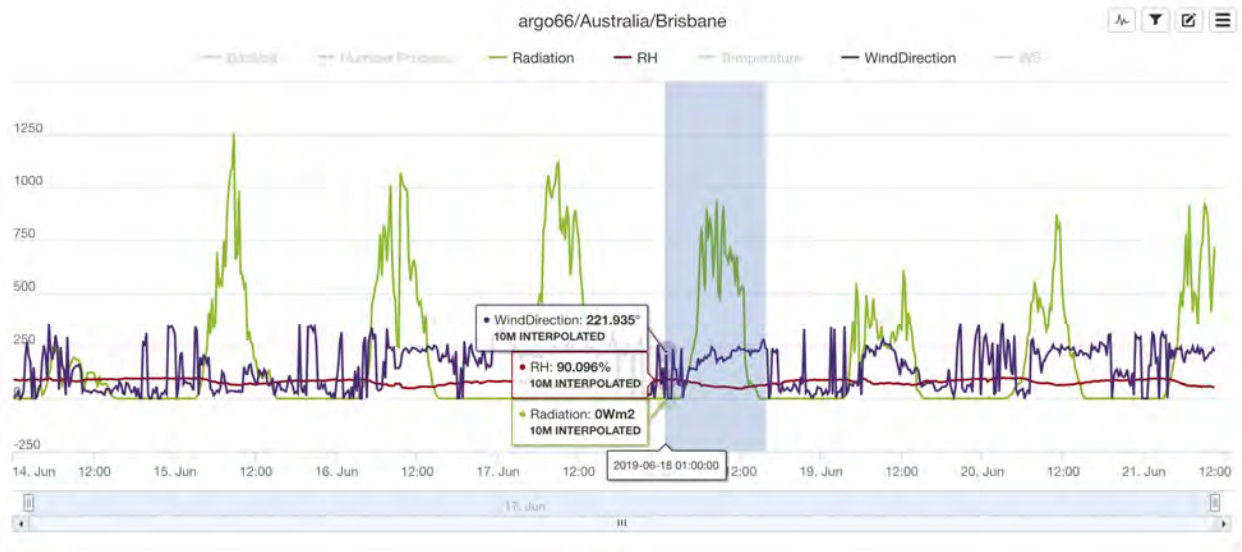
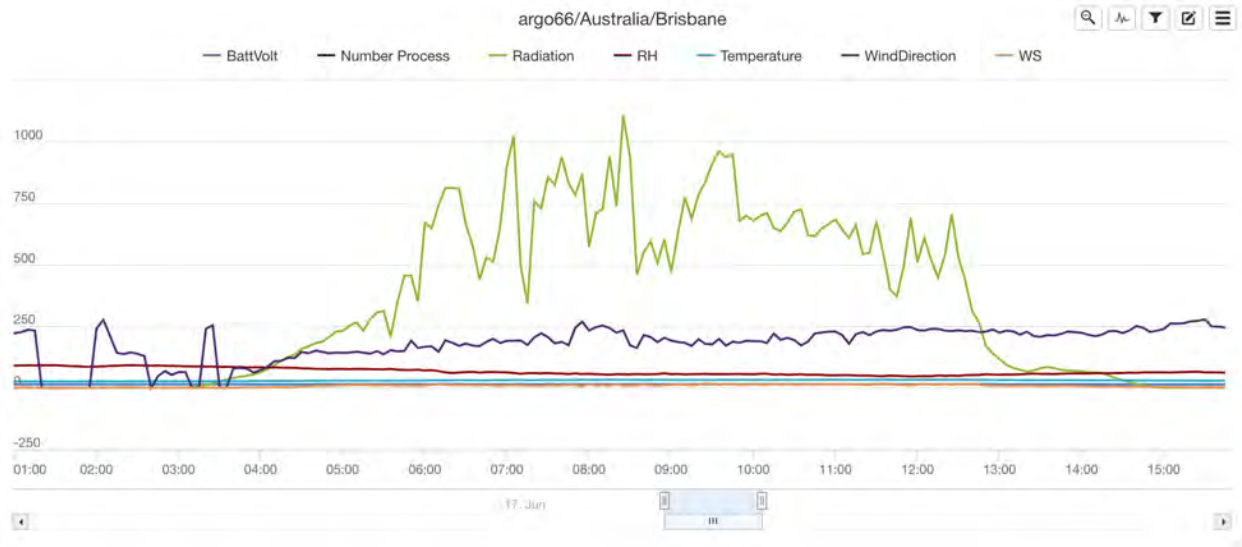
Use the grab handle in the bottom right corner of the chart to increase or decrease its height. This option is currently not available on touch devices.

## Zoom



Click and Drag (or pinch on touch devices) over the main chart area to zoom. As you zoom into the chart, higher resolution data is loaded for the selected period (depending on the parameter configuration settings).

Use the navigator located directly under the main chart area to pan the chart or zoom in/out. After zooming in on a section of the chart the *zoom out* icon is displayed in the chart toolbar to reset the zoom range.

*Chart area selected for zoom**Chart after zoom*



## Toggle Parameter display



Click the parameter icon to quickly toggle the visibility of all parameters. Click parameters in the chart legend to toggle the visibility of individual parameters.



## Change date range



Date range selector is used to quickly adjust the date range being shown on all charts. You can always zoom in on individual charts, but you cannot zoom out past the range set by this selector. Use Custom Range to select precise start/end dates.



Jun 14, 2019 - Jul 24, 2019 ▼

Today
2019-06-14
2019-07-24

24 Hours
3 Days
7 Days
1 Month
3 Months
1 Year
Custom

Apply
Cancel

Jun 2019
Jul 2019

Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1	30	1	2	3	4	5	6
2	3	4	5	6	7	8	7	8	9	10	11	12	13
9	10	11	12	13	14	15	14	15	16	17	18	19	20
16	17	18	19	20	21	22	21	22	23	24	25	26	27
23	24	25	26	27	28	29	28	29	30	31	1	2	3
30	1	2	3	4	5	6	4	5	6	7	8	9	10

## Filter data on quality



Filter the dataset by selecting *quality codes* (page 577) associated with the data points that you want to include. While editing data, *apply* filters to automatically select associated data points and temporarily hide filtered data points to prevent modification.

**Filter / select quality**

☒ **Select all**

☒ **Uncertain**

☒ ★ Unknown

☒ ★ Uncertain - Subnormal [149]

☒ **Bad**

☒ ★ Bad - No Data [155]

☒ ★ Bad - No Bounds [156]

☒ ★ Bad - Data Unavailable [158]

☒ **Good**

☒ ☆ Unset

☒ ★ Good - No Data [165]

☒ ★ Good - Verified [192]

Apply

Reset

### Print or save chart as image



The dropdown menu allows you to print or save the chart as an image.

## Edit chart data



Use this feature to toggle into data editing mode and display the data editing toolbar. Please refer to the [data editing](#) (page 205) documentation for further details.



## 2.2.4 Events



Every interaction (user and system generated) is recorded in the Events log. Use the Events View to quickly audit the system, a user or history of alarms.

**Note:** The view displays content based on the Selected Node in the Workspaces Tree. Use filters to quickly target specific parameters or review alarms.

The events list is live updating and always displayed in reverse chronological order. New events are automatically added to the top of the list. Scroll down the list to view older events.

Type	Time	Source	Message	User
	2013-12-22 15:39:23	argo66/New Zealand/Auckland/New Group	Move node 'argo66/New Zealand/New Group' to 'argo66/New Zealand/Auckland/New Group'	user@company.com
	2013-12-22 15:28:50	argo66/New Zealand/milford-sound.jpg	Created node 'milford-sound.jpg'	user@company.com
	2013-12-22 14:28:44	argo66/New Zealand/Brisbane/Text File Source/Temperature	Rename node 'Temp' to 'Temperature'	user@company.com

## Searching the Event Log

## Jump to Date

Use the Jump to date control to jump to a specific date. After scrolling down, you can scroll back up to show more recent events.



## Detail Level

Select the Detail level of the event log. Debug is the lowest level and will show all events, select a higher detail level to filter out the less important events. For example Info includes Warning and Error events (but not Debug events). The available detail levels include:



Debug



Info



Warning



Error







---

**Note:** Debug level events are automatically purged from the event log after 30 days.

---

## Event Type

Filter events by Event type. Use this drop down to quickly find events related to a certain type of action. For example you can quickly display all Notifications that have been sent for the Selected Node. The available event types include:

	Event Node Type	Event Trigger Action
 Acquisition	Source	<ul style="list-style-type: none"> <li>• User operate action</li> <li>• Scheduled collection</li> <li>• Device connection</li> <li>• File received</li> <li>• API</li> </ul>
 Configuration	All	<ul style="list-style-type: none"> <li>• User configuration change</li> <li>• User or Group subscription change</li> <li>• API</li> </ul>
 State Change	Parameter	<ul style="list-style-type: none"> <li>• Data acquisition</li> </ul>
 Notification	Source, Parameter, Report	<ul style="list-style-type: none"> <li>• Alarm raised/cleared</li> <li>• Report generated on schedule</li> </ul>
 Control	Control Parameter	<ul style="list-style-type: none"> <li>• User operate action</li> <li>• API</li> </ul>
 Security	All	<ul style="list-style-type: none"> <li>• User security change</li> <li>• User login/logout</li> <li>• API</li> </ul>

## Node Type

By default events will be displayed for the Selected Node and all Nodes it contains. You can use the Node type drop down to filter on a specific *type of node* (page 49). For example select Attachment to display all events related to attachments within the Selected Node.



All



Account



Workspace



Folder



Location



Chart



Dashboard



Source



Parameter



Attachment

## 2.2.5 Dash

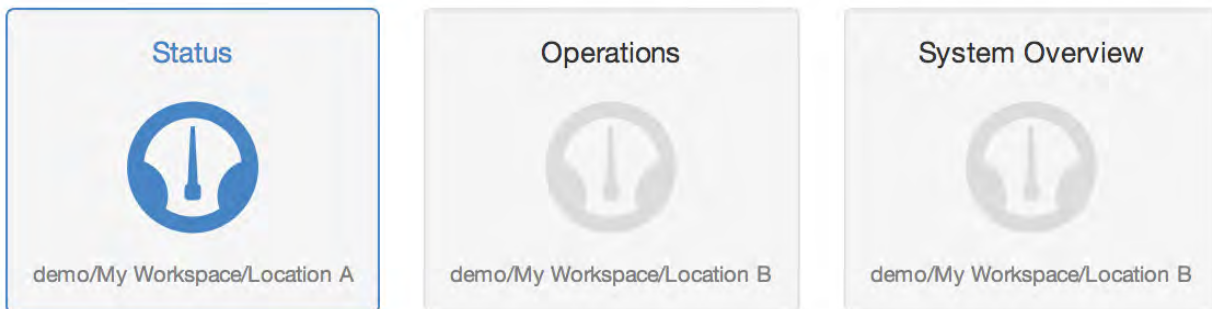


The Dash is for selecting and viewing the customized graphical *dashboards* (page 96).

### Overview



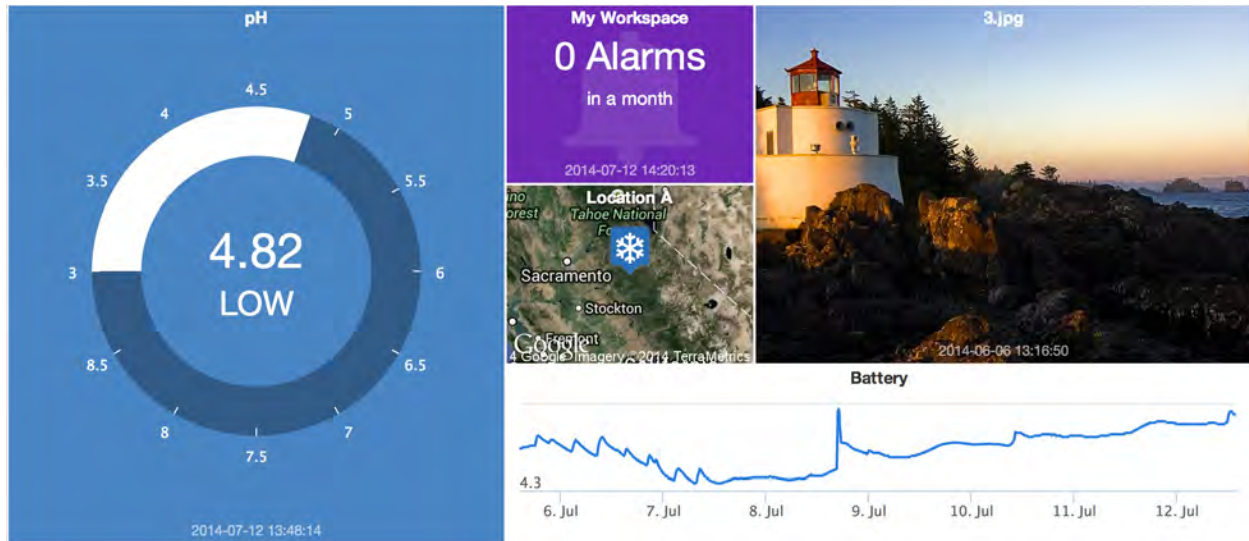
The dashboards overview is displayed when a container Node is selected in the Workspaces Tree. All dashboard Nodes within the Selected Node are shown. Click on a dashboard to select it for display. When viewing an individual Dashboard you can click the Overview button to get back to this view.



### Dashboard view

Dashboards contain user configurable tiles which can be linked to Nodes to display your important data in real-time with live updates. Tile layout is automatically adjusted on mobile devices.

Refer to *Dashboard configuration* (page 96) for Creating and working with Dashboards.





## 2.3 Nodes and Configuration

Nodes are the container objects that store all data and configuration for your projects. The following types of Nodes are available:



**Workspaces** *Workspaces* (page 52) are special container Nodes that can be shared with other users. All other Nodes are created within the Workspace. Workspaces can only be created by the Account Owner or Administrators.



**Folder** Folders are simple nodes that provide structure to your Workspaces. You may choose to group related content into folders by geography or related purpose. Folders can be created inside Workspaces, Locations and other Folders.



**Location** *Locations* (page 56) are special containers used to associate data with a geographical location. Locations can be created inside Workspaces and Groups.



**Template** *Templates* (page 60) are special container Nodes that are used to define consistent structure and configuration. They can be referenced as instances in other Workspaces to speed up automated configuration for large deployments.



**Chart** Create various types of *charts* (page 66) with customized functionality and appearance including historic charts with multiple y-axes and wind rose charts with configurable ranges.



**Dashboard** *Dashboards* (page 96) allow users to create customized views of their data in visually appealing layouts using animated gauges, charts, lists, maps and other graphical controls.



**Attachment** *Attachments* (page 105) include images, documents and other files stored in eagle.io. New attachments can be uploaded manually by a user or acquired via an Attachment Source. Attachments can be added inside Workspaces, Locations, Groups and Attachment Sources.



**Attachment Source** *Attachment Sources* (page 109) have the ability to automatically acquire or receive Attachments. Attachment Sources can be created inside Locations only.



**Data Source** *Data Sources* (page 134) automatically acquire or receive timeseries data using a variety of different transport options. Connect to a *data logger* or collect data from files. Data Sources can be created inside Locations only.



**Processor** *Processors* (page 224) are a type of Data Source containing a process that generates multiple output values, which are stored as new Parameters. Processors can be created inside Locations only.



**Report** *Reports* (page 161) generate PDFs and can be scheduled for delivery to subscribed users and groups via email.



**Number Parameter** Number *Parameters* (page 147) store historic time series data in numeric format. By default they display the current or historic value(s). Parameters are managed by and are always contained in Data Sources.



**Number Parameter (State variation)** Number *Parameters* (page 147) can also be set to display the current or historic state rather than value. This is useful for Boolean or Discrete state data.



**Number Control Parameter** Number Control *Parameters* (page 147) are similar to Number Parameters however the value can be controlled by the user. Historic data represents the user control operations.



**Number Control Parameter (State variation)** Number Control *Parameters* (page 147) can also be set to display the current or historic state(s)



**Number Process Parameter** Number *Process Parameters* (page 223) are similar to Number Parameters however the value is generated by a process.



**Number Process Parameter (State variation)** Number *Process Parameters* (page 223) can also be set to display the current or historic state rather than value. This is useful for Boolean or Discrete state data.



**Rating Parameter** *Rating Parameters* (page 156) are similar to Number Parameters however the value is generated by a rating.



**Rating Parameter (State variation)** *Rating Parameters* (page 156) can also be set to display the current or historic state rather than value. This is useful for Boolean or Discrete state data.



**Text Parameter** Text *Parameters* (page 147) store historic time series data in text format. They are useful when trying to capture events from log files or text-based data files.



**Text Control Parameter** Text Control *Parameters* (page 147) are similar to Text Parameters however the value can be controlled by the user. Historic data represents the user control operations. Please note: Text Control Parameters for Campbell Scientific data loggers currently do not support Control operations.



**Text Process Parameter** Text *Process Parameters* (page 223) are similar to Text Parameters how-

ever the value is generated by a process.



**Time Parameter** Time *Parameters* (page 147) store historic time series data as timestamps. Note: Time Parameters can not be displayed on charts.



**Time Control Parameter** Time Control *Parameters* (page 147) are similar to Time Parameters however the value can be controlled by the user. Historic data represents the user control operations. Note: Time Parameters can not be displayed on charts.



**Time Process Parameter** Time *Process Parameters* (page 223) are similar to Time Parameters however the value is generated by a process.

### 2.3.1 Workspace



Workspaces are special container Nodes that can be shared with other users. All other Nodes are created within the Workspace. Workspaces can only be created by the Account Owner or Administrators.

#### Properties



Configure properties via the Workspaces Tree context menu or List View properties icon. Requires *configure* permission.

#### Map layers

Use the *Map layers* section to add or remove KML (or KMZ) files to be *displayed on the map* (page 21) when this workspace is selected in the *Workspaces Tree* (page 13). Click the *Edit* button to show and modify the configuration for each layer. Use the *Add layer* button to add a new layer to the list. The following configuration options are available:

**Type** **Local KML** allows you to upload a file from your computer.

**Remote KML** should be used when the file is located on a publically accessible web server. Remote KML files will be automatically reloaded every 5 minutes.

**Select file / File URL** Select a file to upload or enter the full HTTP(S) URL to the file.


**Label** The label is displayed in the layers menu on the map toolbar.

Layers can be removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the layer item).

Drag and drop layers in the list (or use the dropdown menu) to change their display order on the map.

Example URL: `https://volcano.si.edu/ge/GVPWorldVolcanoes.kml`

Drag and drop to reorder (max 10 layers)


**Weather Forecast**  
Weather.Bundle.kmz

Hide

Type

Local KML

?

Select file

Browse


Weather.Bundle.kmz

?

Label

Weather Forecast

?


**NOAA Stream Heights**  
http://water.weather.gov/ahps/worldfiles/ahps\_national\_obs.kmz

Hide

Type

Remote KML

?

File URL

http://water.weather.gov/ahps/worldfiles/ahps\_national\_obs.kmz

?

Label

NOAA Stream Heights

?

Add layer

**Note:** Restrictions apply to map layers:

- Maximum fetched file size (raw KML compressed KMZ): 3MB
- Maximum uncompressed KML file size: 10MB
- Maximum number of Network Links: 10
- Maximum number of total document-wide features: 1,000
- Remote layers will be automatically refreshed every 5 minutes

Refer to the [google developer documentation](https://developers.google.com/maps/documentation/javascript/kmllayer?csw=) (https://developers.google.com/maps/documentation/javascript/kmllayer?csw=) for a complete list of restrictions and supported KML elements.

## Security

The Security section allows a user with *security* permission to share the Workspace with other users individually or as a group.

---

**Note:** Workspace permissions do not expire by default. Temporary permissions can be granted by choosing an optional expiry time (this only applies to non-administrative users and groups).

---

Users can be individually added to the workspace by entering the email address of the new users (separate multiple email addresses with a comma), choosing an optional expiry time, assigning a security *role* (page 270) and clicking *Add*. You can remove or modify security roles for existing Workspace users. You can also modify the expiry time for existing Workspace users. Click *Save* when complete.

Name	Email	Role
John Citizen <small>Last login 16 hours ago</small>	john@company.com	Owner
Simon Kelly <small>Last login a year ago</small>	s.kelly@company.com	Administrator
Peter Tennor <small>Save required</small>	peter@company.com <small>Access expires 2019-11-14 10:43:30</small>	View

View Add

You can also *Add* a pre-configured *group* (page 269) of users to the Workspace, choose an optional expiry time, and assign a security *role* (page 270) which will be applied to all users in that group. The *Add groups* dropdown displays a list of all groups that have been configured in the account. To add a new group or assign users to the group refer to the section on *group security* (page 269).

Name	Role
Inspection Team <small>Save required</small>	1 user Access expires 2019-11-14 10:45:33 Operate
Maintenance <small>Save required</small>	1 user View

**Add groups** No groups available View Add

An email notification will be sent to users when they are added or removed from the Workspace or when their security role is changed. New users will need to follow the instructions in the email to configure their user profile and log in. Existing users who are currently logged in and viewing the interface will notice the new settings take affect immediately.

You can restrict user/group access to any Node in the Workspace by opening the corresponding *Node Security* (page 58) dialog.

---

**Note:** Administrators automatically receive unrestricted access to all Workspaces in the account.

---

## 2.3.2 Location



Locations are special containers used to associate data with a geographical location. Locations can be created inside Workspaces and Groups.



### Properties



Configure properties via the Workspaces Tree context menu or List View properties icon. Requires *configure* permission.

### General



**Coordinates** Locations require coordinates for them to be displayed in Map View. The coordinates drop down displays a list of Sources (within the Location) that can be used to dynamically update the Location latitude and longitude. Select *Fixed Location* for manual positioning of the location on the map.

 Fixed Location 

**Map Marker** Select the color and style of the marker displayed on the map. Click the icon to reveal a drop down menu.

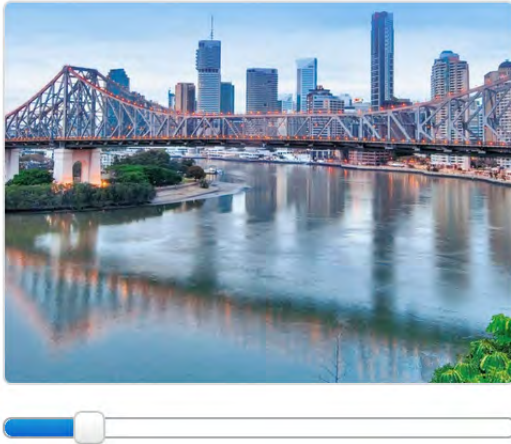


**Image** Locations can have an image associated with them. The image can be static (uploaded by user), or dynamically updated with an Attachment Source (within the Location). Select *Fixed Image* to upload an image or choose an existing image attachment (within the Location).

 Fixed Image 

**Image zoom/pan** Fixed images can be zoomed and panned as required. Use the zoom slider to quickly zoom in/out and click or touch the image and drag around the preview container to pan.







**Quality Alarm** See *Alarms and Notifications* (page 174).

## Metadata

Metadata provides a simple key/value data structure that can be used to store static information about the location. Values can be saved as Number or Text and are displayed in the Locations List View.

Label	Value	Type	
<input type="text" value="Address"/>	<input type="text" value="1600 Amphitheatre Parkway, Mounta"/>	<input type="text" value="Text"/>	
<input type="text" value="Service Tag"/>	<input type="text" value="44301"/>	<input type="text" value="Number"/>	
<b>Add New</b>			
<input type="text" value="Label"/>	<input type="text" value="Value"/>	<input type="text" value="Text"/>	<input type="button" value="Add"/>

## Historic

Click the *Delete* buttons and save to permanently remove all *historic coordinate data* or *events* for this Location node. Alternatively use the *Source Historic* (page 121) section to delete historic data or events for a Data Source and its Parameters.

**Historic data**Delete

This will permanently remove all historic coordinate data for this location.


**Events**Delete


This will permanently remove all events for this location node.

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

 Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

 Groups		
Name		Role
Inspection Team	 3 users	No Access

---

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

---

### 2.3.3 Template



Templates are special container Nodes that allow a hierarchy of configuration to be defined once and replicated to any number of *Instances* (page 62). This reduces configuration time and ensures consistency for large deployments, while allowing individual attributes to be customised.

## WORKSPACES

### Templates

#### Template 1

##### Dashboard

##### Historic Chart

##### Text File Source

### Workspace

#### Instance 1

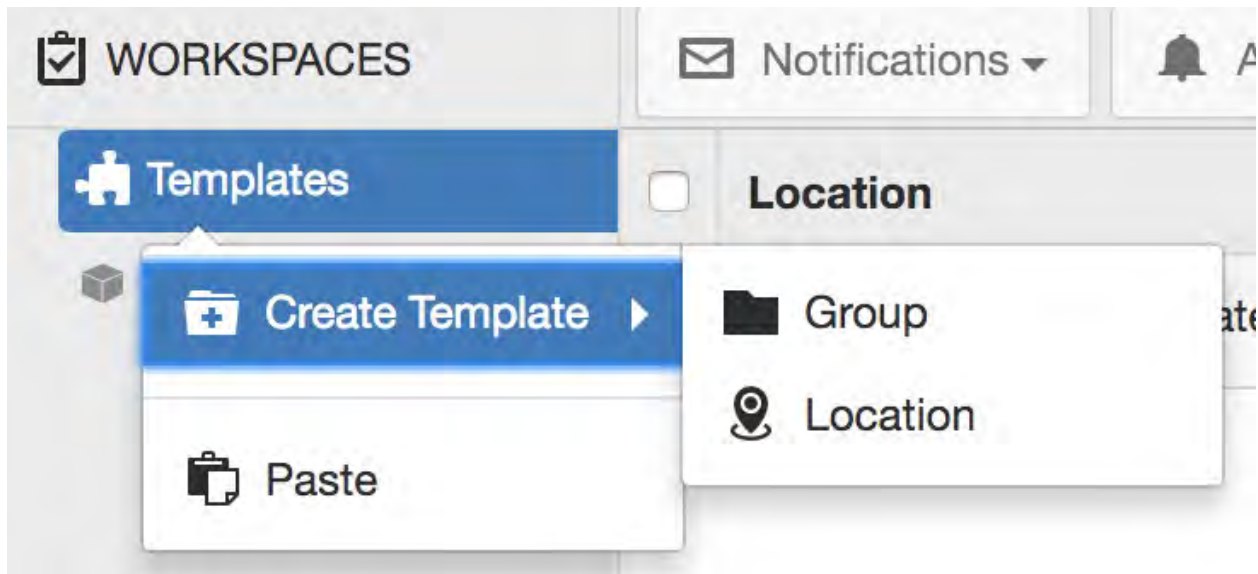
##### Dashboard

##### Historic Chart

##### Text File Source

#### Instance 2

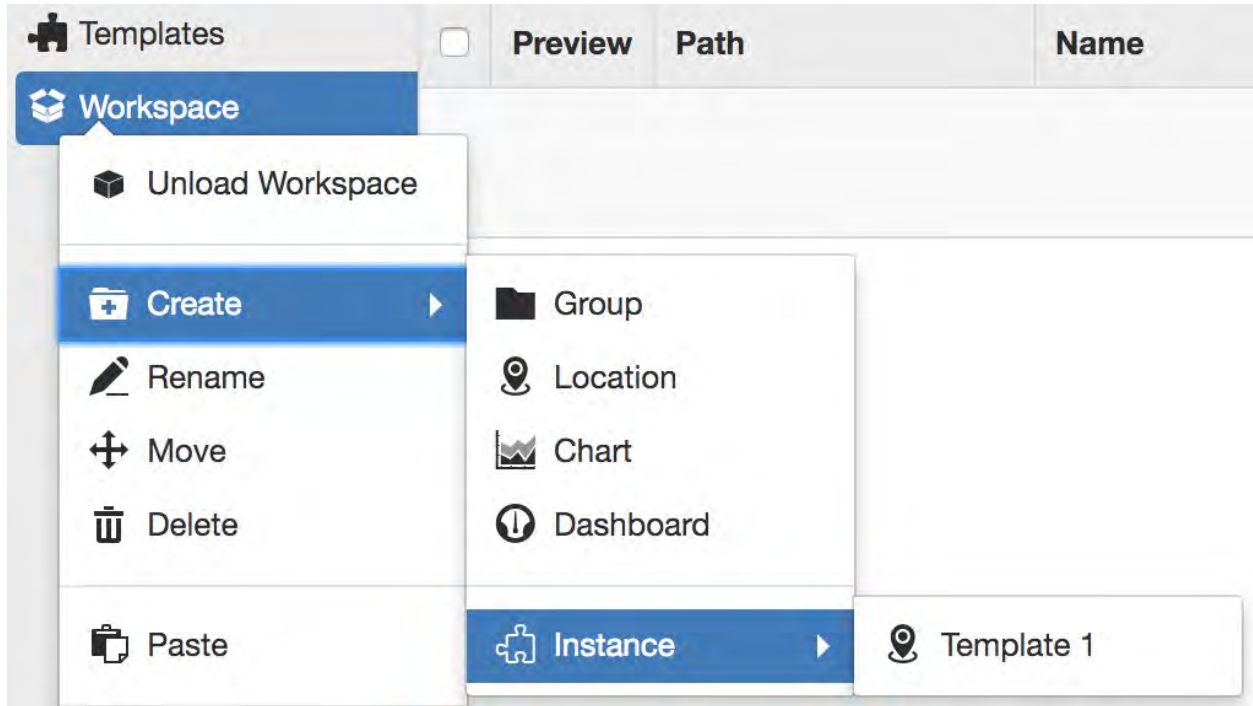
The *Templates* Node always appears at the top of the Workspaces Tree and is identified by the puzzle icon. A new Template begins as a Group or Location, and is then populated with any number of Nodes. Changes to an existing Template (or Node therein) will be immediately reflected in all Instances referring to that Template. These changes can include the addition or removal of Nodes, or any configuration change within the Template.



**Warning:** Removing a Template will also remove all Instances linked to the Template.

## Instances

Instances can be created in any of your Workspaces and must refer to one of the Templates you have already created. Each Instance may be renamed but all additional configuration items will be locked to the Template.

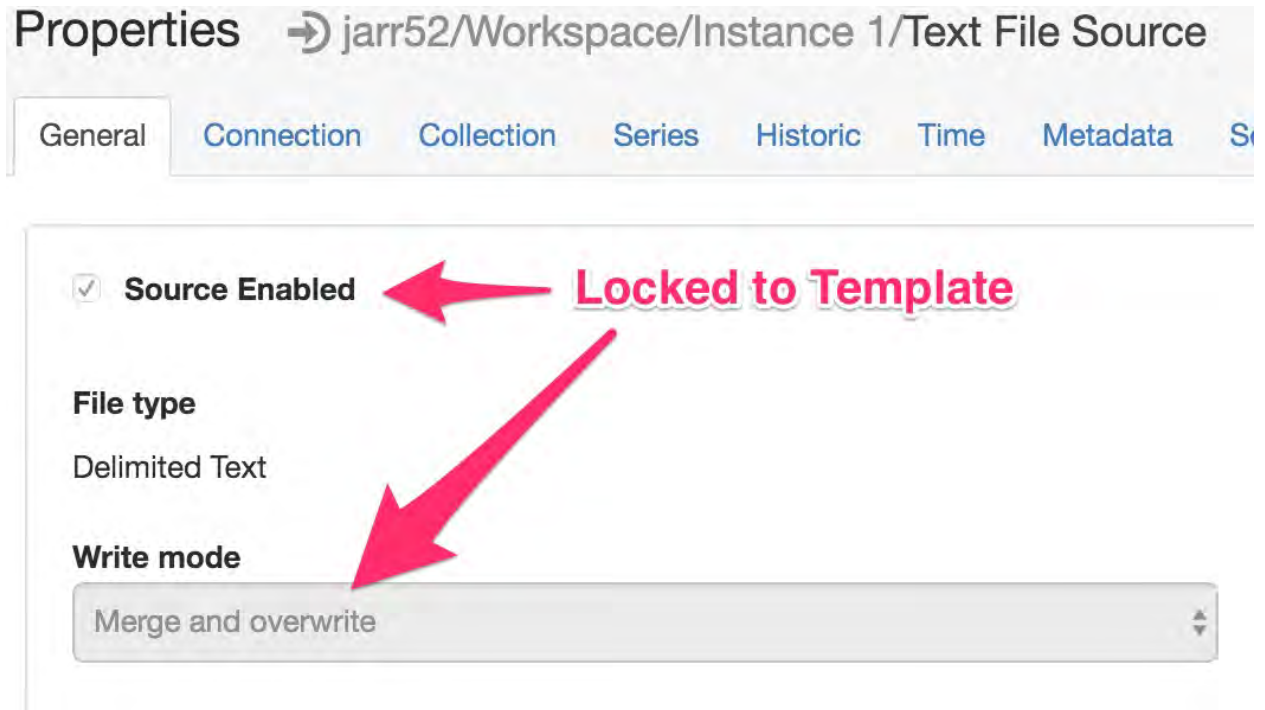


---

**Note:** An Instance can only be created directly under a Workspace or Folder.

---

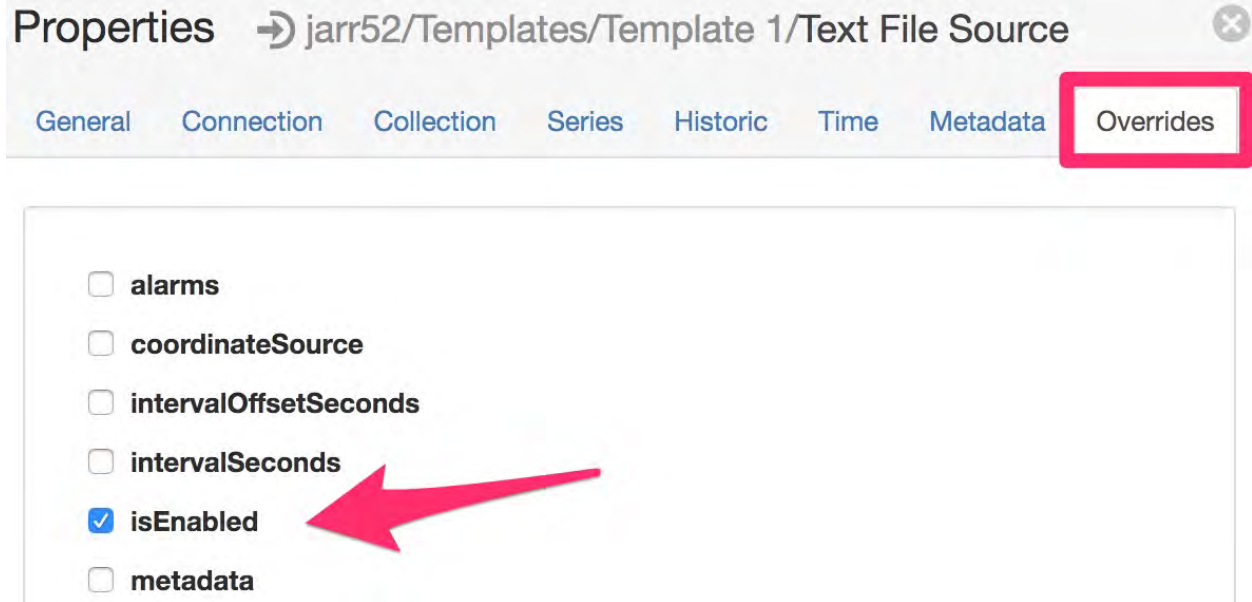
By default, Instance configuration attributes are locked to the Template and cannot be modified. In this way, locked attributes ensure consistency across all Instances that refer to the same Template.



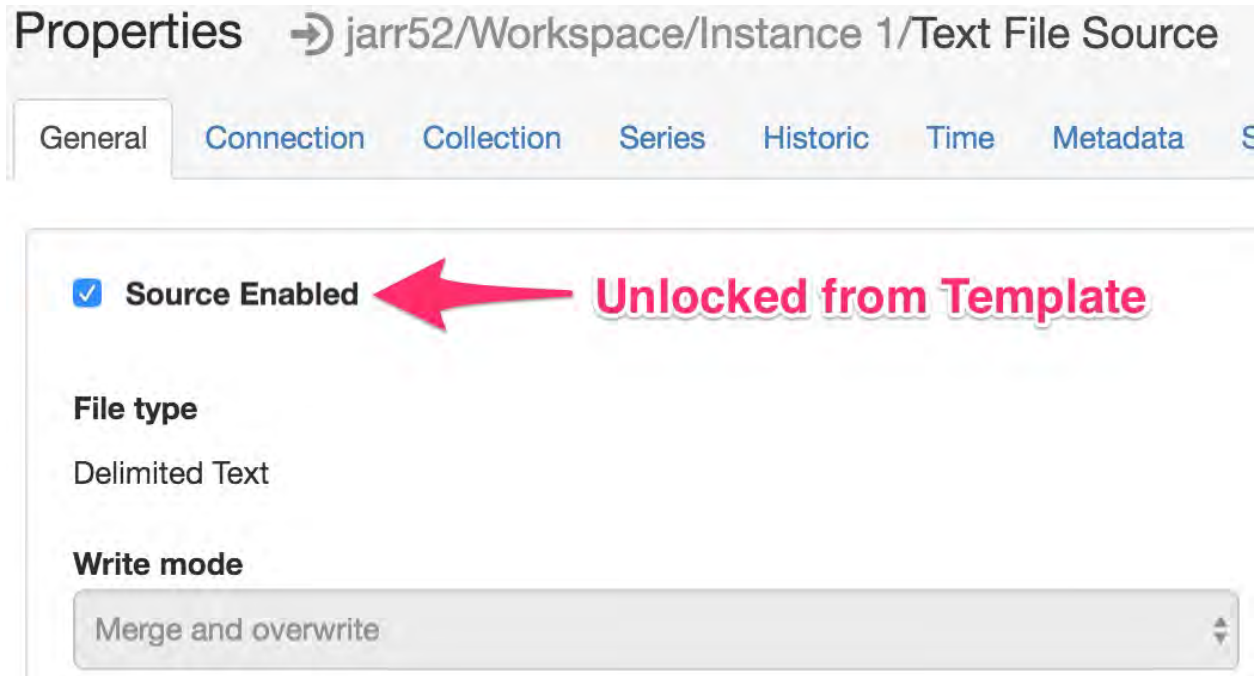
## Overrides

Overrides allow you to select attributes in a Template that may be individually customised in the corresponding instance Nodes. Each type of Template Node will expose a different list of available attributes. Any attributes that have not been selected as overrides in the Template will be shown as locked in the Instance configuration, indicating that they cannot be changed.





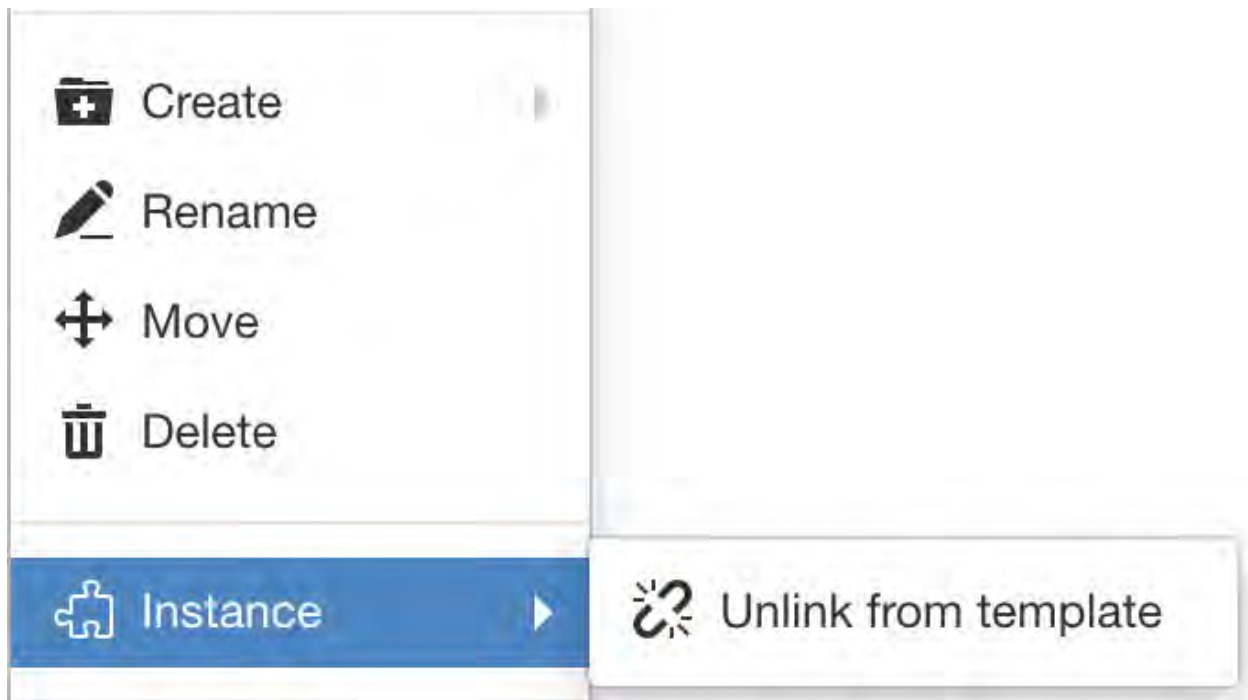
Once an override has been added in the Template configuration, the attribute will become modifiable in every Instance.



**Warning:** Removing an override in a Template will reset that attribute in the corresponding Instance Nodes back to the templated value.

## Instance operations

The Instance context-menu accessible via the Workspaces Tree contains instance operations.



### Unlink from Template

Unlinking an Instance will permanently disassociate it from the Template. The instance (and its descendants) will be converted to standard nodes. Configuration, events and historic data will remain intact.

## 2.3.4 Chart



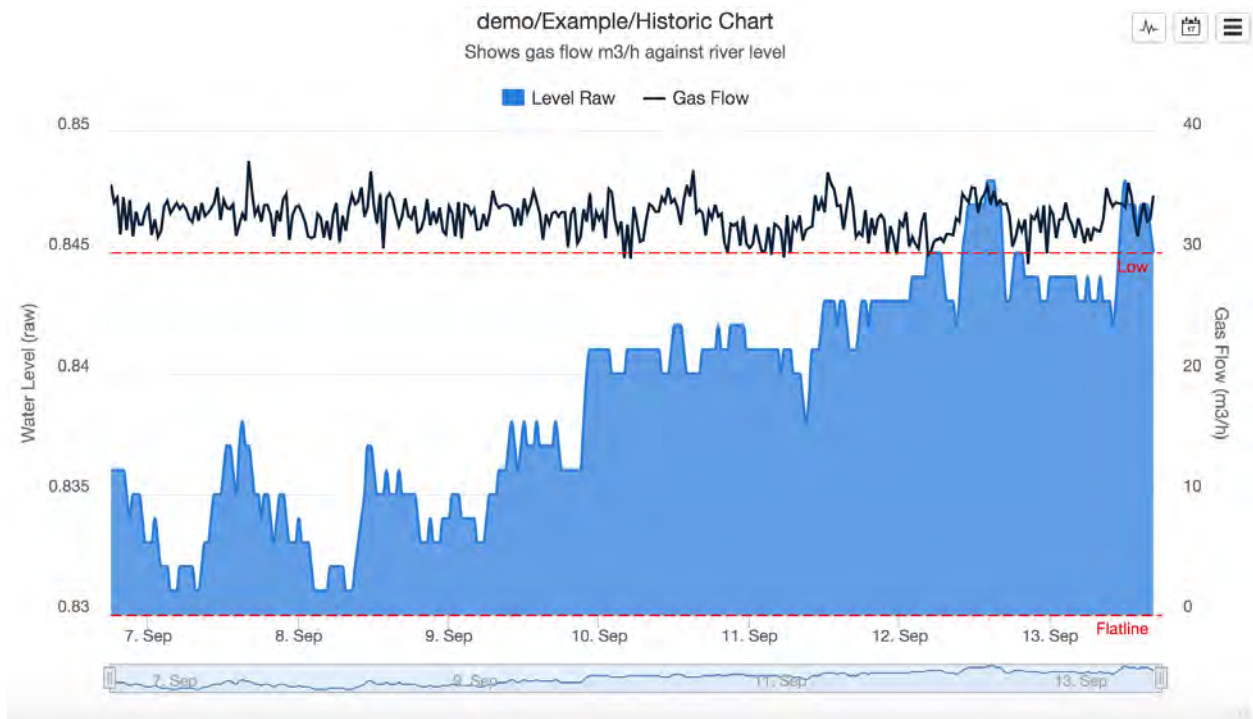
Create various types of charts with customized functionality and appearance including historic charts with multiple y-axes, displacement charts and wind rose charts with configurable ranges.

To view custom charts navigate to *Chart View* → *Custom*. You will see a list of all custom charts under the selected Node in the Workspaces Tree. You can also display any pre-configured charts as a *Custom chart* tile on a *Dashboard* (page 96).

Types of charts available:

## Historic

Historic charts allow for more customization over the default *Location* and *Parameter* charts available in [Chart View](#) (page 35). You can create multiple y-axes with fixed or automatic scaling, change the data aggregation and styles of individual parameters and preset the default time range.



You can customize the chart to visually indicate assigned *quality codes* (page 577) via associated quality colors. Refer to *Quality style* in *Parameter chart configuration* (page 152).



## Properties



Configure properties via the Workspaces Tree' context menu. Requires *configure* permission.

## General

The General section allows a default *Start range* and *End range* to be set. This range will be used when the chart is initially displayed. Choose a preset or *Custom OPC* for *relative time* (page 585). Select *Custom Fixed* for absolute time.

<b>Start range</b>	<div>Custom Fixed</div> <div>2016-01-01 00:00:00</div>	?
<b>End range</b>	<div>Now</div>	?

Optionally enter a description to be displayed on the chart below the title.

Description	<input type="text" value="optional description displayed on chart"/>	?
-------------	----------------------------------------------------------------------	---

## Axes

The Axes section allows you to customize the style and functionality of the X-axis and Y-axes of the chart.

### X Axis

Set the position of the x-axis (time) to display on the *Top* or *Bottom* of the chart.

Position	<input type="text" value="Bottom"/>	?
----------	-------------------------------------	---

### Y Axes

You can create unlimited y-axes to be displayed on the chart.

Click the *Edit* button to show and modify the configuration for each y-axes. Use the *Add Y axis* button to create a new axis. Axes can be removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the axis item).

Y-axes can be *linear* with *automatic* or *fixed* minimum and maximum ranges or displayed with *logarithmic* scaling.

Drag and drop axes in the list to change their display order on the chart.

**Y axes**Drag and drop to reorder

**Axis 1**

Axis will be shown as **1st** axis on **left** side of chart.  
Range will be **automatically** scaled based on series being displayed.

Hide

**Name**

?

☒ Show label on axis

**Position**

Left

?

**Range**

Linear - Automatic

?

☐ Invert axis values

**Axis 2**

Axis will be shown as **1st** axis on **right** side of chart.  
Range will be **logarithmically** scaled based on series being displayed.

Edit

Add Y axis

---

**Note:** By default, links are automatically assigned to the first y-axis. You need to *Edit* each link and specify a *Y axis*.

---

## Links

Links assign Nodes in the Workspaces Tree to the chart so the corresponding data can be displayed and updated live.


Click the *Edit* button to show and modify the configuration for each link. Use the *Add link* button to display the Workspaces Tree and select a new Node to link. Links can be removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the link item).

Charts can not have links to Nodes in different Workspace and currently do not support the same nodes being linked more than once to the same chart.

Drag and drop links in the list to change their display order on the chart.


Refer to [Parameter chart configuration](#) (page 152) for a description of all available settings.

Drag and drop to reorder

 **Temperature**


demo/Workspace/Sector 3/New Location/Text File Source/Temperature


Parameter will be shown as **line** series on **Axis 1** axis.




Hide


General   **Style**   Data   Customize


**Link**  change 

**Name**  

**Y axis**


Axis 1




 **Relative Humidity**

demo/Workspace/Sector 3/New Location/Text File Source/Relative Humidity

Parameter will be shown as **line** series on **Axis 2** axis.




Edit

Add link

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

 Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

 Groups		
Name		Role
Inspection Team	 3 users	No Access

---

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

---

## Public Access

Enabling public access allows anyone to view the node (attachment, chart, dash, report or historic data) using the generated *URL*.

You can also embed charts and dashboards into your own website with basic HTML. Select *Copy embed HTML* from the URL dropdown menu for a generic code snippet (you will need to paste this in to your website). You can also revoke and replace the automatically assigned URL by selecting the *Generate new URL* option in the menu.

Use the available options to customize the URL based on your display preferences. These options are transient and you can use multiple variations of the URL at the same time.



Public access

On

?

URL

<https://public.eagle.io/public/chart/nzjbx01ii4mcsn>

COPY

Format

HTML - Interactive

Header

Show logo

Timezone

Auto-detect

☐ Ignore Daylight Savings Time (fixed UTC offset)

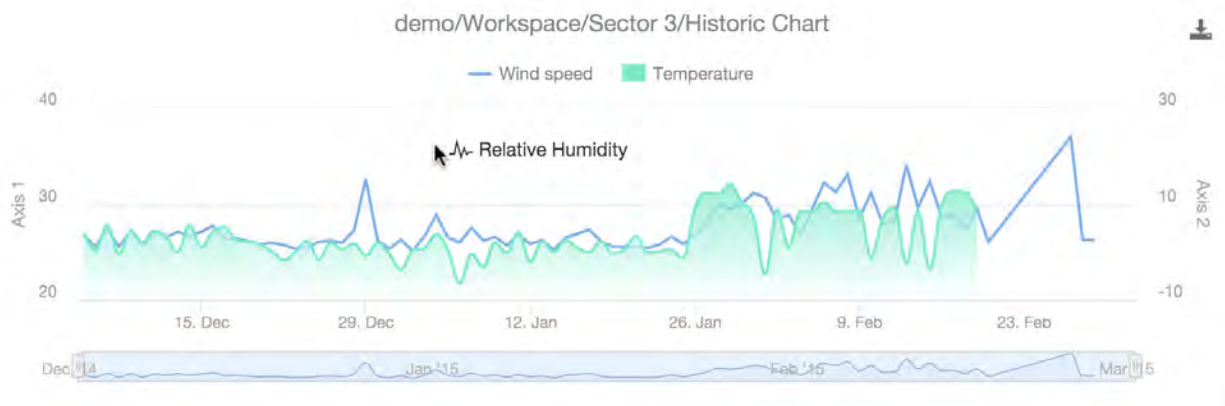
i

The link provides anonymous access to this chart

**Note:** This feature is not included in *Starter* accounts.

## Drag and Drop Links

Nodes can be dragged with a mouse directly from the Workspaces Tree and dropped onto applicable charts (indicated with drop-accept and drop-blocked mouse cursor icons).



## Copy/Paste Charts



Copy/Paste operations can be performed from the Workspaces Tree to quickly duplicate one or more charts. Links are automatically re-assigned to corresponding Nodes relative to the chart if they exist at the destination. ie. You can copy/paste complete Locations containing charts without manually re-assigning all Links.

## Move Charts



Moving a chart within a Workspace will keep existing links. Moving a chart between Workspaces will invalidate any Link where the linked node is not available in the destination Workspace.

## Link error

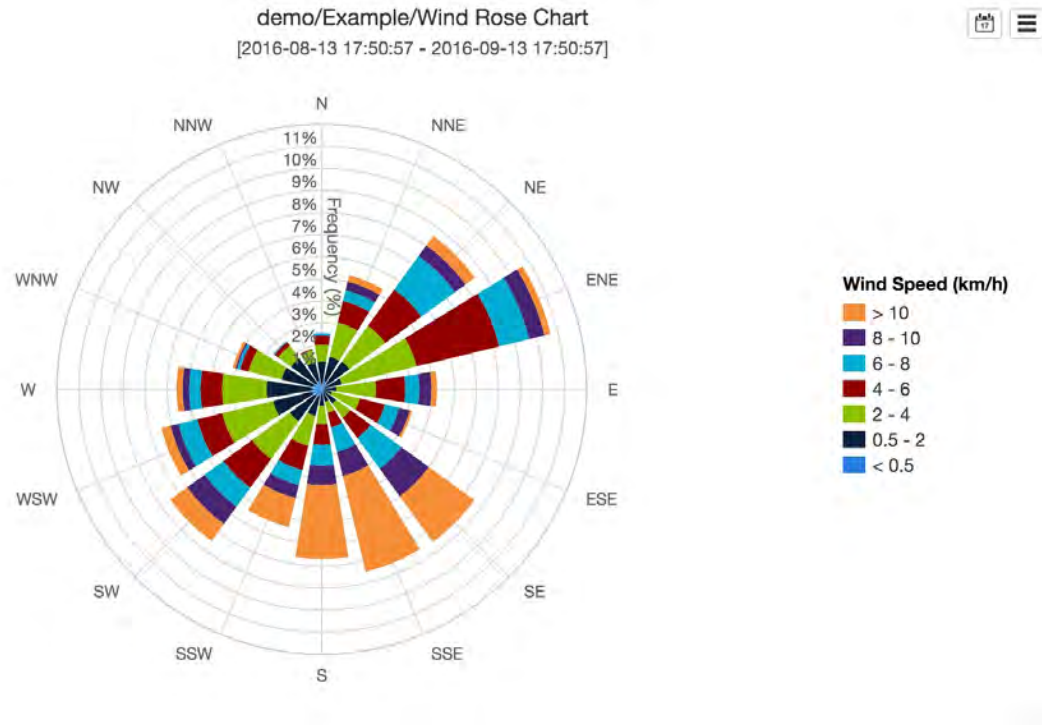


If a linked Node is removed or not accessible after a copy/paste or move operation then the chart will display the Link error icon to indicate a problem with one or more Links. Links can be removed or reconfigured from the Links section in the chart properties dialog.

## Wind Rose

Wind Rose charts provide a view of how wind speed and direction are typically distributed at a particular location.

Using a polar coordinate system of gridding, the frequency of winds over a time period is plotted by wind direction, with color bands showing wind speed ranges. The direction of the longest spoke shows the wind direction with the greatest frequency.



## Properties




Configure properties via the 'Workspaces Tree' context menu. Requires *configure* permission.

## General

The General section allows a default *Start range* and *End range* to be set. This range will be used when the chart is initially displayed. Choose a preset or *Custom OPC* for *relative time* (page 585). Select *Custom Fixed* for absolute time.

<b>Start range</b>	<div>Custom Fixed</div> <div>2016-01-01 00:00:00</div>	?
<b>End range</b>	<div>Now</div>	?

Optionally enter a description to be displayed on the chart below the title.

<b>Description</b>	<input type="text" value="optional description displayed on chart"/>	
--------------------	----------------------------------------------------------------------	-------------------------------------------------------------------------------------

## Axes

The Axes section allows you to customize the style and functionality of the chart including *Speed* and *Speed Ranges*.

## Speed

Set the *legend title* and *speed units* to be displayed on the chart and tooltips. *Speed range colors* is set to *Automatic* by default. If you would like to overwrite colors for speed ranges you should select *Fixed* and *Edit* the specific ranges that require changes.

<b>Legend title</b>	<input type="text" value="Wind speed"/>	
<b>Speed units</b>	<input type="text" value="m/s"/>	
<b>Speed range colors</b>	<input type="text" value="Fixed"/>	

## Speed ranges

Click the *Edit* button to show and modify the configuration for each speed range including its lowest threshold limit. Use the *Add range* button to create new ranges. Ranges can be removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the range item).

> 10

Range will include values greater than 10 m/s

Hide

Threshold

10

?

Color

#f28f43

?

8 - 10

Range will include values between 8 and 10 m/s

Edit

6 - 8

Range will include values between 6 and 8 m/s

Edit

4 - 6

Range will include values between 4 and 6 m/s

Edit

2 - 4

Range will include values between 2 and 4 m/s

Edit

0.5 - 2

Range will include values between 0.5 and 2 m/s

Edit

< 0.5

Range will include values less than 0.5 m/s

Edit

Add range


**Note:** Range items are automatically reordered based on threshold value.

## Links

Links assign Nodes in the Workspaces Tree to the chart so the corresponding data can be displayed and updated live. Wind Rose charts require both *speed* and *direction* parameters (direction should contain numeric values between 0-360).

Click the *Edit* button to show and modify the configuration for each link. Use the *change* button to display the Workspaces Tree and select a new Node to link.

Charts can not have links to Nodes in different Workspace and currently do not support the same nodes being linked more than once to the same chart.

 **Speed**  
demo/Example/Text File Source/Wind Speed  
Parameter will be used as wind rose **speed**.


Hide

Link

demo/Example/Text File Source/Wind Speed

change

?


 **Direction**  
demo/Example/Text File Source/Wind Direction  
Parameter will be used as wind rose **direction**.




Edit

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

 Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

 Groups		
Name		Role
Inspection Team	 3 users	No Access 

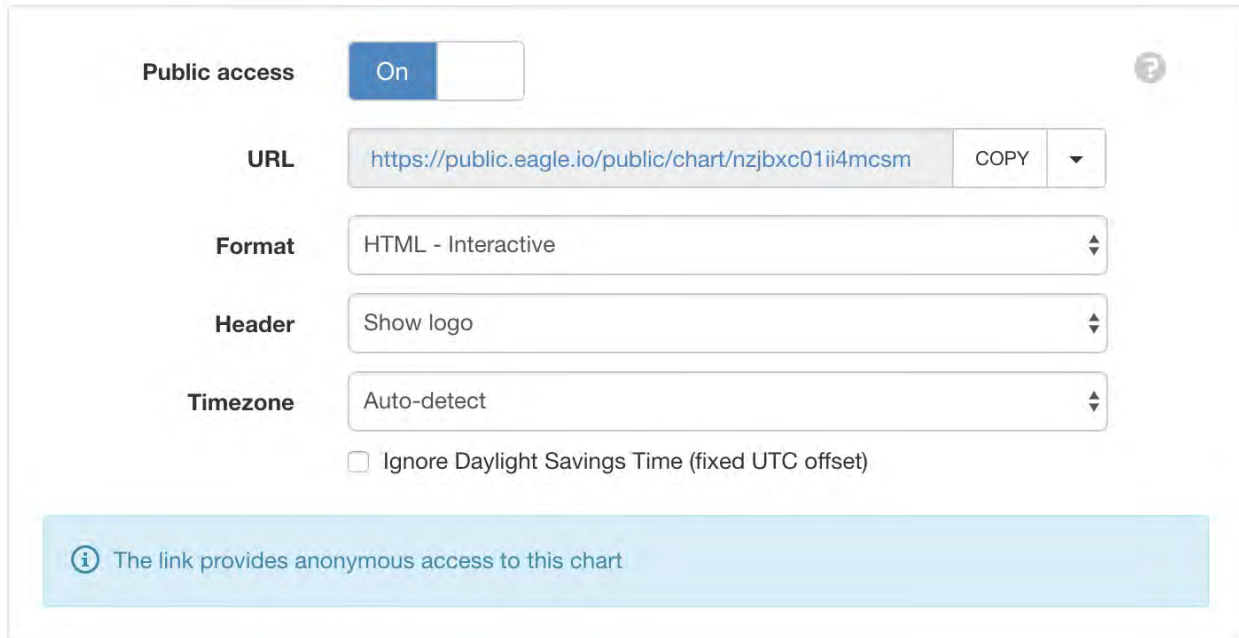
**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

## Public Access

Enabling public access allows anyone to view the node (attachment, chart, dash, report or historic data) using the generated *URL*.

You can also embed charts and dashboards into your own website with basic HTML. Select *Copy embed HTML* from the URL dropdown menu for a generic code snippet (you will need to paste this in to your website). You can also revoke and replace the automatically assigned URL by selecting the *Generate new URL* option in the menu.

Use the available options to customize the URL based on your display preferences. These options are transient and you can use multiple variations of the URL at the same time.



The screenshot shows a settings panel for public access. At the top, 'Public access' is set to 'On' with a toggle switch and a help icon. Below this, the 'URL' field contains the link 'https://public.eagle.io/public/chart/nzjbx01ii4mcsm' with a 'COPY' button and a dropdown arrow. The 'Format' dropdown is set to 'HTML - Interactive'. The 'Header' dropdown is set to 'Show logo'. The 'Timezone' dropdown is set to 'Auto-detect', with an unchecked checkbox for 'Ignore Daylight Savings Time (fixed UTC offset)'. A light blue information box at the bottom states: 'The link provides anonymous access to this chart'.

---

**Note:** This feature is not included in *Starter* accounts.

---

## Drag and Drop Links

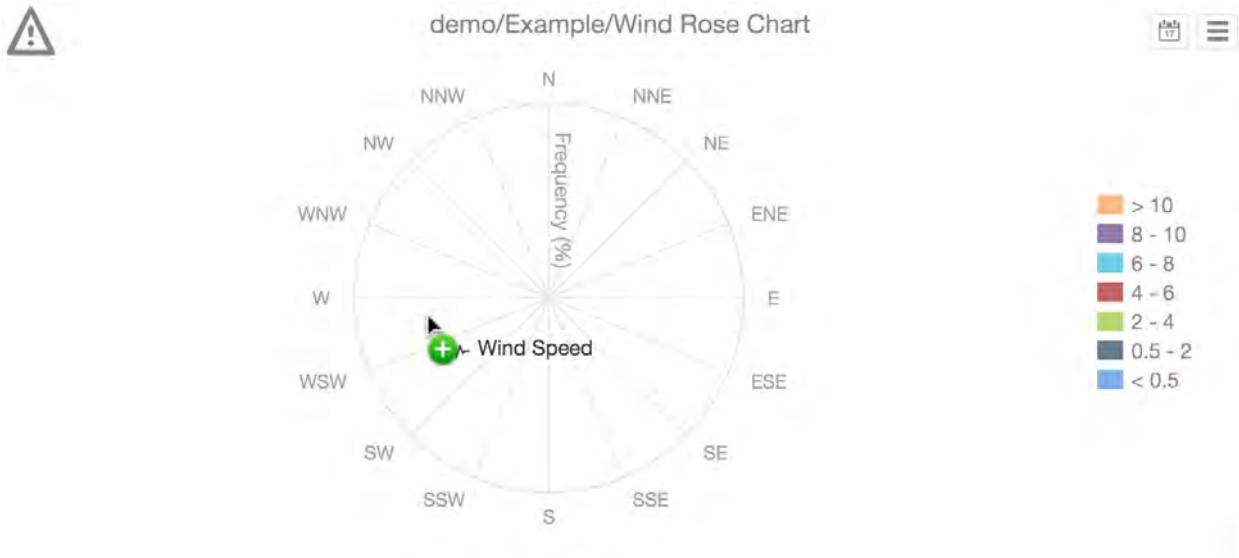
Nodes can be dragged with a mouse directly from the Workspaces Tree and dropped onto applicable charts (indicated with drop-accept and drop-blocked mouse cursor icons).

---

**Note:** Parameters will be automatically assigned as either *speed* or *direction* based on parameter names. However you should review the *links* from the properties dialog to ensure correct assignment.

---





## Copy/Paste Charts



Copy/Paste operations can be performed from the Workspaces Tree to quickly duplicate one or more charts. Links are automatically re-assigned to corresponding Nodes relative to the Chart if they exist at the destination. ie. You can copy/paste complete Locations containing charts without manually re-assigning all links.

## Move Charts



Moving a chart within a Workspace will keep existing links. Moving a chart between Workspaces will invalidate any Link where the linked node is not available in the destination Workspace.

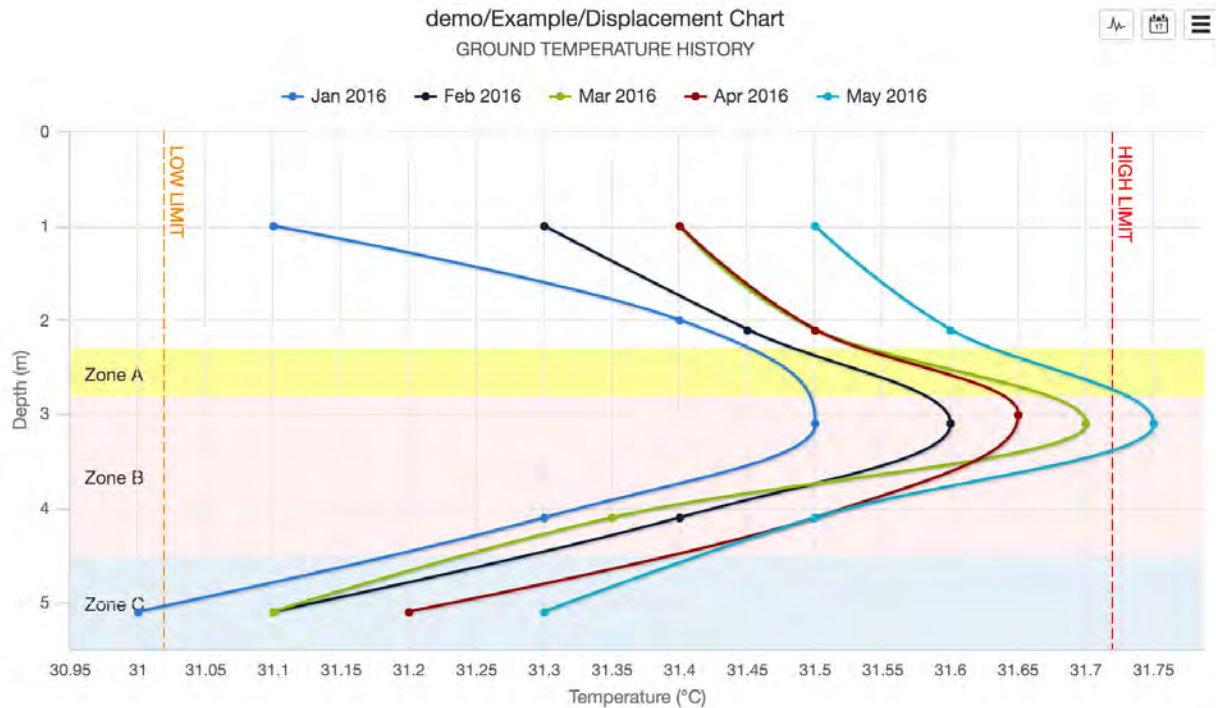
## Link error



If a linked Node is removed or not accessible after a copy/paste or move operation then the chart will display the Link error icon to indicate a problem with one or more Links. Links can be removed or reconfigured from the Links section in the chart properties dialog.

## Displacement

Displacement charts allow you to visualize *vertical* or *horizontal* measurement displacement over time. Typically, instruments that require displacement charts are on a string along a line, either down a borehole, along a structure or underneath embankments.



## Properties



Configure properties via the Workspaces Tree' context menu. Requires *configure* permission.

## General

The General section allows a default *Start range* and *End range* to be set. This range will be used when the chart is initially displayed. Choose a preset or *Custom OPC* for *relative time* (page 585). Select *Custom Fixed* for absolute time. The *None* option can be used to read forward or back through the time series data until the *Maximum series* limit is reached.

<b>Start range</b>	<div>Custom Fixed</div> <div>2016-01-01 00:00:00</div>	?
<b>End range</b>	<div>Now</div>	?

You can customize the chart with a number of other general settings:

<b>Profile orientation</b>	<div>Vertical - Depth</div>	?
<b>Maximum series</b>	<div>10</div>	?
<b>Series time format</b>	<div>MMM YYYY</div> <div>▼</div>	?

example: Nov 2016

<b>Measurement baseline</b>	<div>Absolute</div>	?
<b>Measurement process</b>	<div>Raw</div>	?
<b>Measurement offset</b>	<div>0</div>	?
<b>Measurement units</b>	<div>°C</div>	?
<b>Depth units</b>	<div>m</div>	?

**Profile orientation** *Vertical* displays depth on Y-Axis and measurement on X-Axis. *Horizontal* displays distance on X-Axis and measurement on Y-Axis.

**Maximum series** The maximum number of series to display on the chart (between 1 and 15).

**Series time format** Select a preset display format from the drop down list or specify a custom format using [time format tokens](#) (page 583).

**Measurement baseline** *Absolute*: Measurements are not modified.

*Relative to first value in range*: Measurements from first series in requested range are subtracted from all subsequent series.

*Relative to first recorded value:* Measurements from first series in parameter history are subtracted from all subsequent series.

*Relative to fixed time:* Measurements from series at a user-defined time are subtracted from all subsequent series. Select the time from the control below this option.


**Measurement process** Processing to apply to all measurement values. *Raw* will not apply any processing. *Sum from Start* and *Sum from End* will totalise measurement values.

**Measurement offset** Numeric value to add/subtract from each measurement value.

**Measurement units** Suffix to display with measurement values.

**Depth/Distance units** Suffix to display with the depth/distance values.





Optionally enter a description to be displayed on the chart below the title.

<b>Description</b>	<input type="text" value="optional description displayed on chart"/>	
--------------------	----------------------------------------------------------------------	-------------------------------------------------------------------------------------





## Axes

The Axes section allows you to customize the style and functionality of the chart axes.

**X axis**

<b>Name</b>	<input type="text" value="Temperature"/>	
	<input checked="" type="checkbox"/> Show label on axis	
<b>Position</b>	<input type="text" value="Bottom"/>	
<b>Range</b>	<input type="text" value="Linear - Automatic"/>	
<b>Tick interval</b>	<input type="text" value="0.5"/>	
	<input type="checkbox"/> Invert axis values	

**Y axis**

<b>Name</b>	<input type="text" value="Depth"/>	
	<input checked="" type="checkbox"/> Show label on axis	
<b>Position</b>	<input type="text" value="Left"/>	
<b>Range</b>	<input type="text" value="Linear - Fixed"/>	
<b>Minimum range</b>	<input type="text" value="0"/>	
<b>Maximum range</b>	<input type="text" value="5.5"/>	
<b>Tick interval</b>	<input type="text" value="interval between tick mark on axis. eg. 0.5 or leave empty for automa"/>	
	<input checked="" type="checkbox"/> Invert axis values	

**Thresholds**

Threshold *lines* and *bands* can be added to the chart and displayed either horizontally or vertically. The *threshold* value for a line will determine where the line is placed on the axis. The *threshold start* and *threshold end* values for a band will determine the band position on the axis.

Drag and drop thresholds in the list to change their display order on the chart. Thresholds can be removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the threshold item).

Lines and bands Drag and drop to reorder

Line 1

Threshold **line** will be shown **vertically** at **31.02** on the X axis.

Hide

Name

Line 1

?

☒ Show label

Type

Line

?

Threshold

31.02

?

Orientation

Vertical

?

Line style

Long Dash

?

Color

#f20ba3

?

Band 1

Threshold **band** will be shown **horizontally** between **2.3** and **2.8** on the Y axis.

Edit

Add threshold

## Style

Use the style section to change the appearance of the chart series. Refer to [parameter chart configuration](#) (page 152) for a description of all available settings.

Colors

☒ Automatic ☐ Fixed

Line type

Normal

Spline

Line width

2

Line style

Solid

Marker type

Circle

Marker size

3

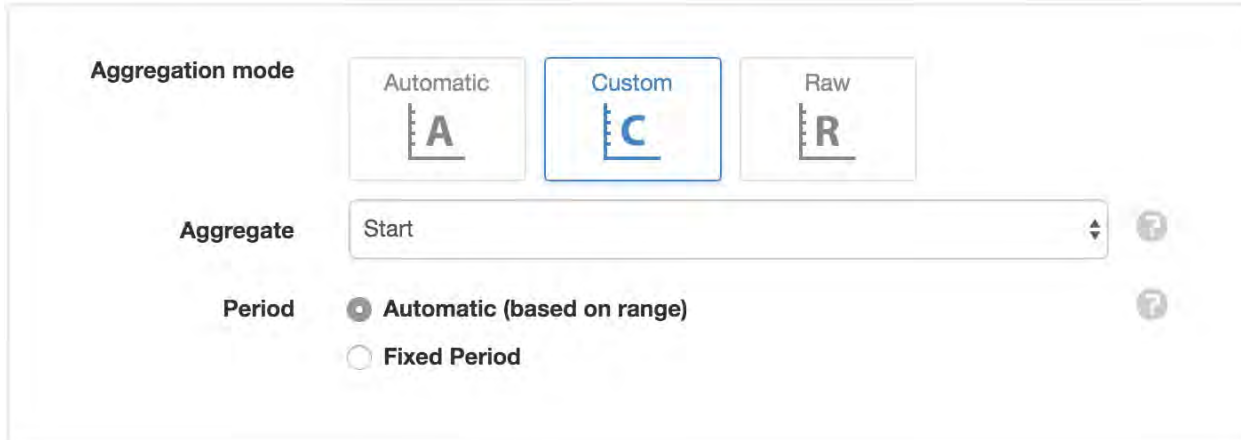
3d effects

☒ Drop shadow

Reset to defaults

## Data

Use the data section to set the aggregation mode for time series data to be displayed on the chart. The aggregation mode applies to the measurement data, while the aggregation period applies to the time ranges. Refer to [parameter chart configuration](#) (page 152) for a description of all available aggregation settings.



The screenshot shows the 'Aggregation mode' settings in eagle.io. It features three buttons: 'Automatic' (labeled 'A'), 'Custom' (labeled 'C' and highlighted with a blue border), and 'Raw' (labeled 'R'). Below these, the 'Aggregate' dropdown is set to 'Start'. The 'Period' section has two radio buttons: 'Automatic (based on range)' (selected) and 'Fixed Period'. Help icons (question marks) are visible next to the 'Aggregate' dropdown and the 'Automatic (based on range)' radio button.

**Automatic** Automatic allows the chart to determine the most suitable period based on the chart range.

**Custom** Custom requires the choice of aggregate type (which will be applied to the measurement data) and period interval (which will determine the timestamp of each series).

**Raw** Raw uses the timestamps and measurements of the records directly as the series represented on the chart. However, as the maximum number of series that can be displayed on a chart is 15, while a typical data file could contain hundreds or thousands of individual timestamped records, some form of aggregation is generally recommended.

## Profile & Links

Displacement charts require you to define a profile containing depths/distances with corresponding measurement parameters.

Choose from 3 *profile type* options:

### Parameter

Set the *profile type* to *parameter* for dynamic depth/distance values based on time series data stored in linked parameters. Optionally specify a numeric *offset* to add or subtract from each depth/distance value.



Depth profile type

Parameter

?

Depth offset

0

?

Depth parameters

Link 1

:s/Text File Source/depth 1

≡

Link 2

:s/Text File Source/depth 2

≡

Link 3

:s/Text File Source/depth 3

≡

Link 4

:s/Text File Source/depth 4

≡

Link 5

:s/Text File Source/depth 5

≡

Add links

Remove all

Measurement parameters

Link 1

ts/Text File Source/temp 1

≡

Link 2

ts/Text File Source/temp 2

≡

Link 3

ts/Text File Source/temp 3

≡

Link 4

ts/Text File Source/temp 4

≡

Link 5

ts/Text File Source/temp 5

≡

Add links

Remove all

### Fixed - Manual

Set the *profile type* to *fixed - manual* to manually define static depth/distance values.

2.3. Nodes and Configuration

89

**Depth profile type** Fixed - Manual ?

**Depths (m)**

Value 1	1.5	≡
Value 2	2	≡
Value 3	2.6	≡
Value 4	3.2	≡
Value 5	4.1	≡

Add depth Remove all

**Measurement parameters**

Link 1	ts/Text File Source/temp 1	≡
Link 2	ts/Text File Source/temp 2	≡
Link 3	ts/Text File Source/temp 3	≡
Link 4	ts/Text File Source/temp 4	≡
Link 5	ts/Text File Source/temp 5	≡

Add links Remove all

### Fixed - Incremental

Set the *profile type* to *fixed - incremental* to automatically generate depth/distance values using *start* and *increment* values.

**Depth profile type**

Fixed - Incremental

?

**Depth start**

0

?

**Depth increment**

5

?

**Depths (m)**

Value 1

0

Value 2

5

Value 3

10

Value 4

15

Value 5

20

**Measurement parameters**

Link 1

ts/Text File Source/temp 1

Link 2

ts/Text File Source/temp 2

Link 3

ts/Text File Source/temp 3

Link 4

ts/Text File Source/temp 4

Link 5

ts/Text File Source/temp 5

Add links

Remove all

Drag and drop items in each list to re-order. Links and Values can be updated or removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the item).

Charts can not have links to Nodes in different Workspace and currently do not support the same nodes being linked more than once to the same chart.

---

**Note:** You must ensure each depth/distance has a corresponding measurement in the correct order. eg. 'Measurement Link 1' is associated with 'Depth Link 1', etc.

---

Use the *Add links* button to link multiple parameters at once. The *Nodes* list will display all parameters under the current *Path*. Click the *Change* button next to the path to select an alternate (or higher) node in the current Workspace.

You can filter the list by typing part of a parameter name in the *Node name* search field. Parameters that have already been linked will be shown as disabled in the node list.

Use the checkboxes to select the parameters you would like to link and click *Add links*.

**Add measurement links**

Path: `nick10/Dev/Sample/` Change

Node name: `temp` Reset

☐ Match using regular expression Matching 5 / 10 nodes

☒ **Nodes**

- ☒ `nick10/Dev/Sample/Text File Source/temp 1`
- ☒ `nick10/Dev/Sample/Text File Source/temp 2`
- ☒ `nick10/Dev/Sample/Text File Source/temp 3`
- ☒ `nick10/Dev/Sample/Text File Source/temp 4`
- ☒ `nick10/Dev/Sample/Text File Source/temp 5`

Select measurement parameters to link

Add links Cancel

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

Groups		
Name		Role
Inspection Team	 3 users	No Access

---

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

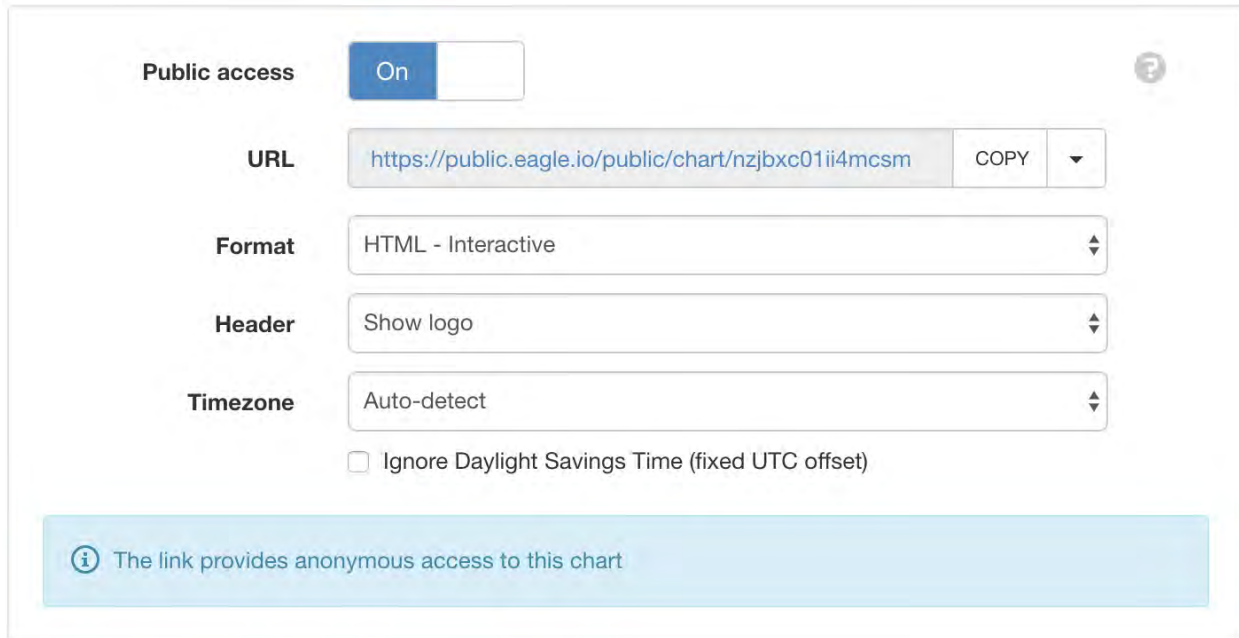
---

## Public Access

Enabling public access allows anyone to view the node (attachment, chart, dash, report or historic data) using the generated *URL*.

You can also embed charts and dashboards into your own website with basic HTML. Select *Copy embed HTML* from the URL dropdown menu for a generic code snippet (you will need to paste this in to your website). You can also revoke and replace the automatically assigned URL by selecting the *Generate new URL* option in the menu.

Use the available options to customize the URL based on your display preferences. These options are transient and you can use multiple variations of the URL at the same time.



The screenshot shows a settings panel for public access. At the top, 'Public access' is set to 'On' with a toggle switch. To the right is a help icon. Below this, the 'URL' field contains the link 'https://public.eagle.io/public/chart/nzjbx01ii4mcsn' and a 'COPY' button. The 'Format' dropdown is set to 'HTML - Interactive'. The 'Header' dropdown is set to 'Show logo'. The 'Timezone' dropdown is set to 'Auto-detect', with an unchecked checkbox for 'Ignore Daylight Savings Time (fixed UTC offset)'. A light blue information box at the bottom states: 'The link provides anonymous access to this chart'.

---

**Note:** This feature is not included in *Starter* accounts.

---

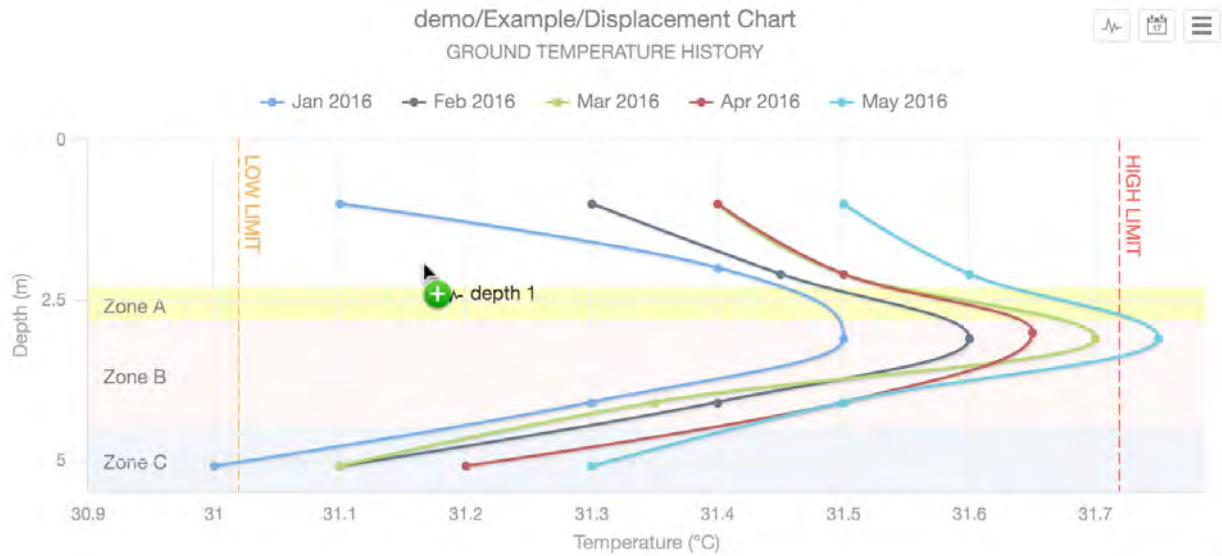
## Drag and Drop Links

Nodes can be dragged with a mouse directly from the Workspaces Tree and dropped onto applicable charts (indicated with drop-accept and drop-blocked mouse cursor icons).

---

**Note:** Parameters will be automatically assigned as either *depths/distances* or *measurements* based on parameter names. You should review the *profile & links* from the properties dialog to ensure correct assignment and ordering.

---



## Copy/Paste Charts



Copy/Paste operations can be performed from the Workspaces Tree to quickly duplicate one or more charts. Links are automatically re-assigned to corresponding Nodes relative to the Chart if they exist at the destination. ie. You can copy/paste complete Locations containing charts without manually re-assigning all links.

## Move Charts



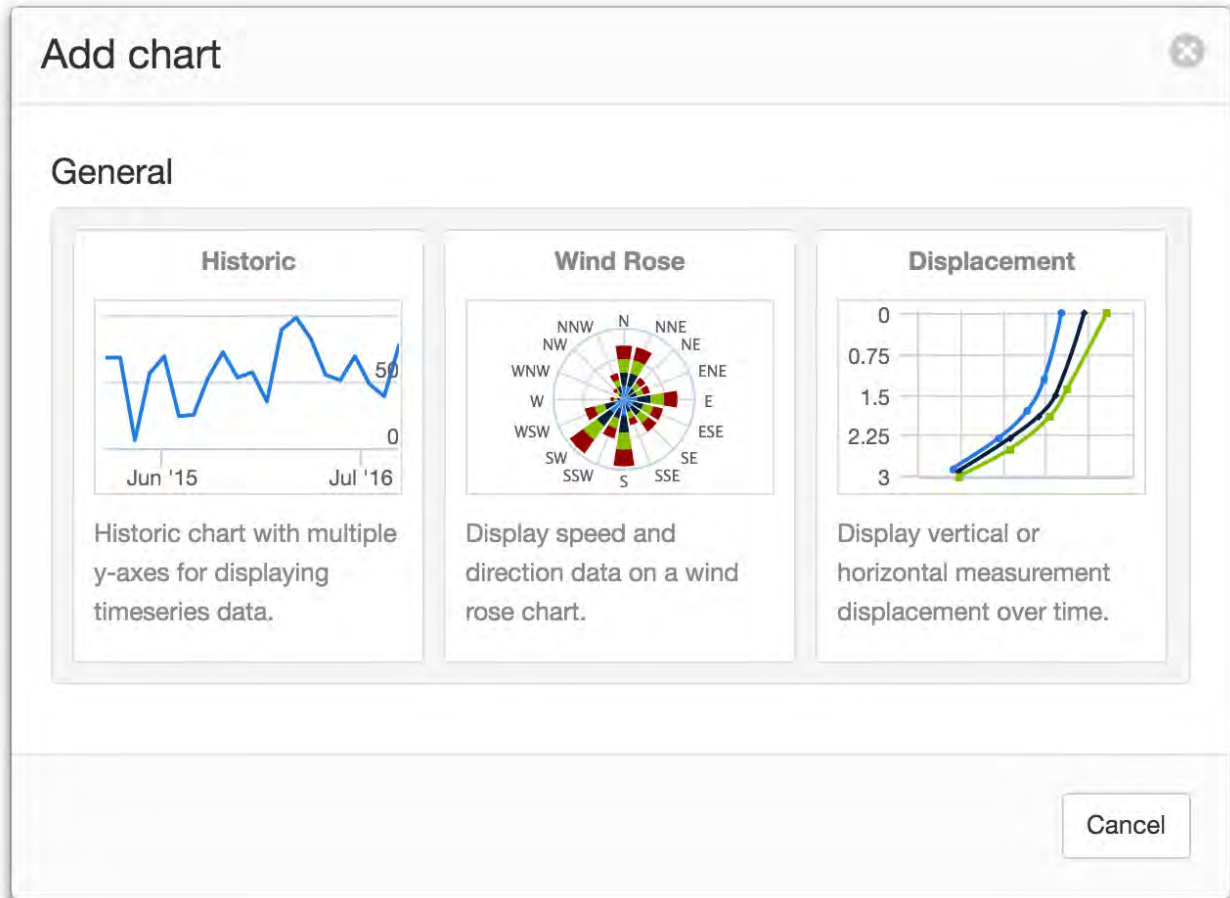
Moving a chart within a Workspace will keep existing links. Moving a chart between Workspaces will invalidate any Link where the linked node is not available in the destination Workspace.

## Link error



If a linked Node is removed or not accessible after a copy/paste or move operation then the Chart will display the Link error icon to indicate a problem with one or more Links. Links can be removed or reconfigured from the Profile & Links section in the chart properties dialog.

Use the *Create* menu to add a new custom chart.



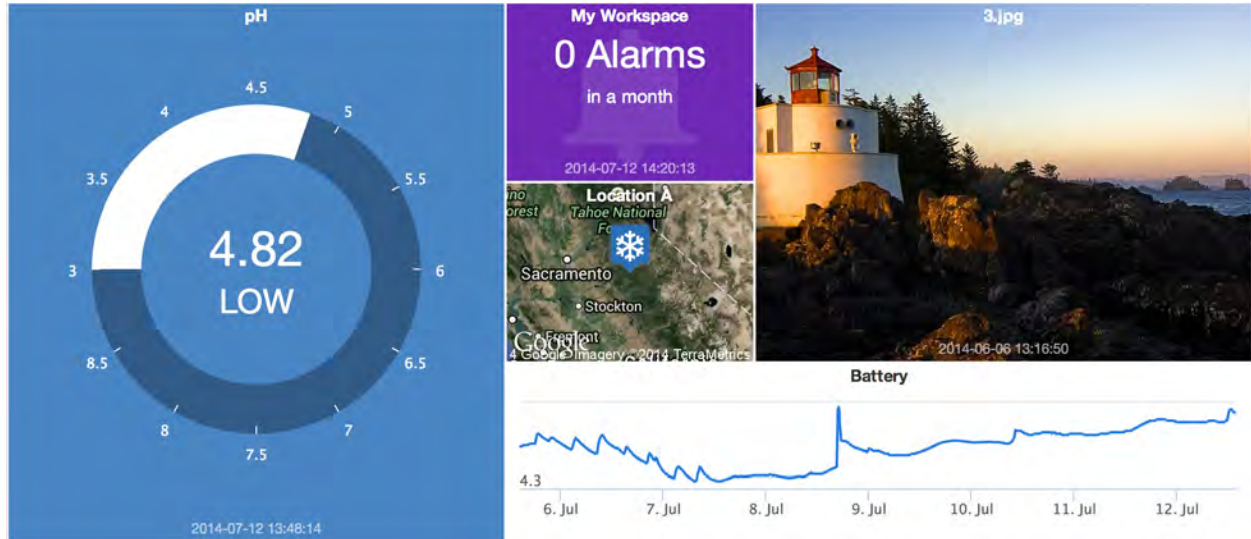
### 2.3.5 Dashboard



Dashboards allow users to create customized views of their data in visually appealing layouts using animated gauges, charts, lists, maps and other graphical controls.

Dashboards contain user configurable tiles which can be linked to Nodes to display your important data in real-time with live updates. Tile layout is automatically adjusted on mobile devices.

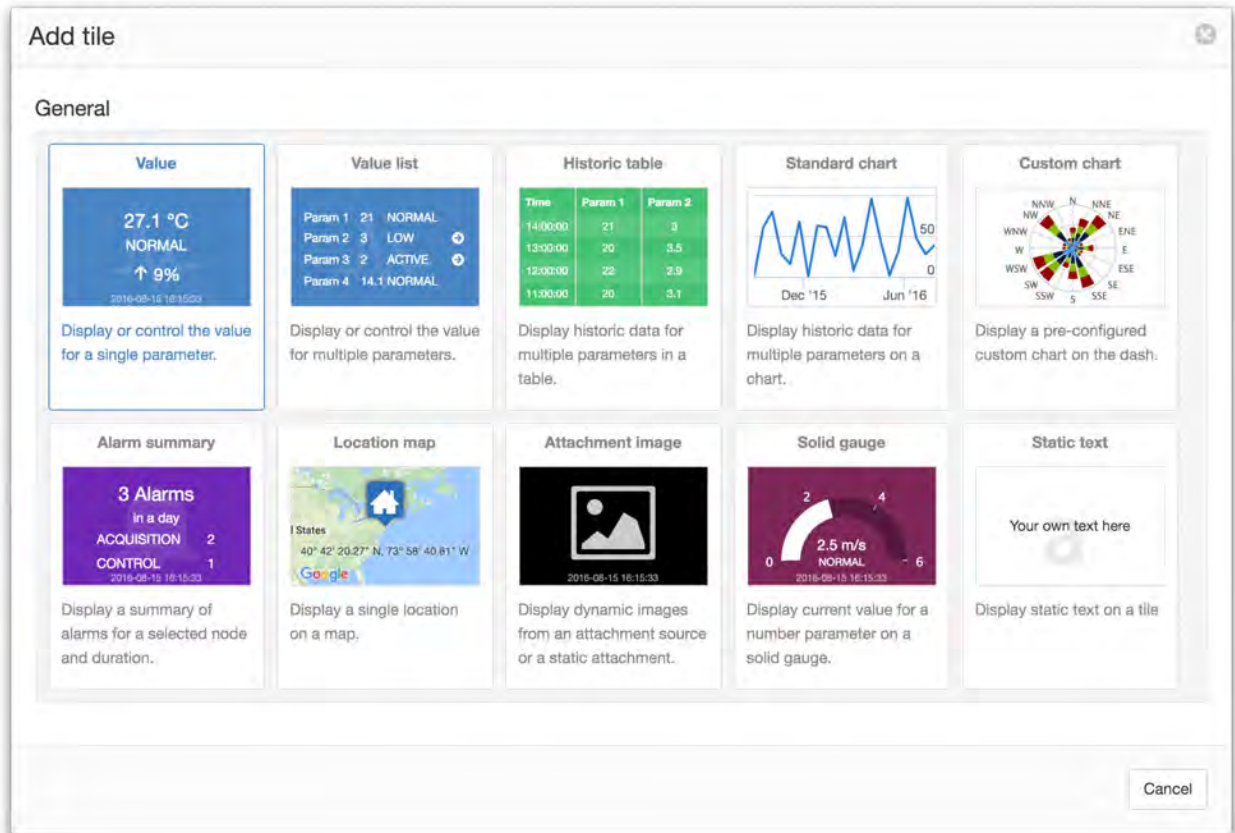




### Add tile

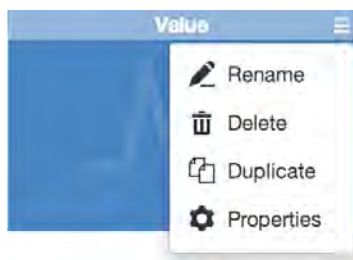


Click the Add tile button from the Toolbar to open the Tile library. Select the type of Tile you would like to add from the presets or click cancel to close the dialog.



## Tile operations

Moving the mouse over or touching a tile will show the title bar and menu button to users with *Configure* permission.



## Reposition

Click or Touch, hold and drag the title bar to reposition the Tile on the Dashboard. Tile layout is automatically adjusted to suit the display device and tiles are always ordered in rows starting from the top left to bottom right of the screen. Requires *Configure* permission. **Note:** Reposition operation not available on small screen touch devices (width <820px).



## Resize

Click or Touch, hold and drag the resize handle to adjust the width and height of the Tile. Requires *Configure* permission. **Note:** Resize operation not available on small screen touch devices (width <820px).



## Rename



Select *Rename* from the tile menu to update or remove the Tile label.



## Delete



Delete the Tile from the Dashboard.

## Duplicate



Duplicate the current Tile and its configuration.

## Properties



Configure the Tile properties to change General display settings and Links.

### General

The general section allows you to configure layout and display options for the Tile. Certain types of Tiles have specific options. For example the Historic Chart tile includes Period and Zoom settings while most other Tiles have selectable color styles and alarm options.

#### Style



☒ Show alarms with orange/red styles.

### Links

Links assign Nodes in the Workspaces Tree to Tiles so the corresponding data can be displayed and updated live. Each type of Tile may enforce specific Link requirements. For example the Location map tile can only be linked to a single Location Node. Some tiles allow for multiple links to be assigned.

Click the *Edit* button to show and modify the configuration for each link. Use the *Add link* button to display the Workspaces Tree and select a new Node to link. Links can be removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the link item).

Dashboard tiles can not have links to Nodes in different Workspace and currently do not support the same nodes being linked more than once to the same tile.

Drag and drop links in the list to change their display order on the tile.

Drag and drop to reorder

**Battery**

demo/Workspace/Sector 3/New Location/Text File Source/Battery

Link will be shown on dash tile as Battery.

Hide

Link

demo/Workspace/Sector 3/New Location/Text File Sour

change

?

Name

Battery

?

**Temperature**

demo/Workspace/Sector 3/New Location/Text File Source/Temperature

Link will be shown on dash tile as Temperature.

Edit

Add link

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

Groups		
Name		Role
Inspection Team	 3 users	No Access 

---

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

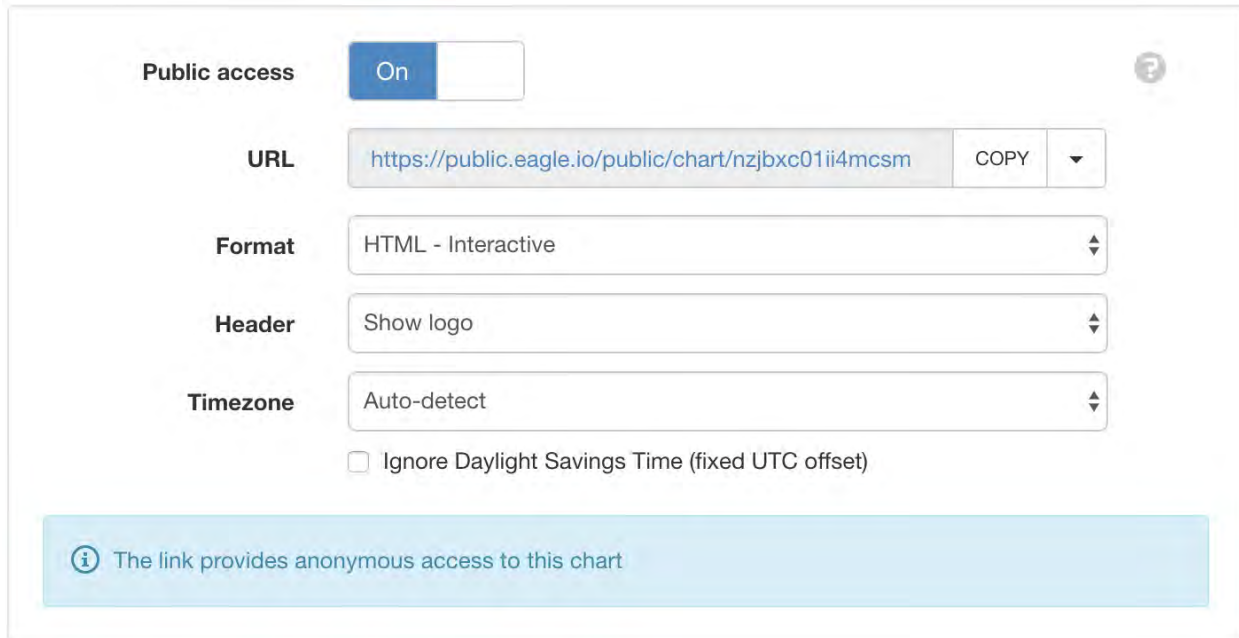
---

## Public access

Enabling public access allows anyone to view the node (attachment, chart, dash, report or historic data) using the generated *URL*.

You can also embed charts and dashboards into your own website with basic HTML. Select *Copy embed HTML* from the URL dropdown menu for a generic code snippet (you will need to paste this in to your website). You can also revoke and replace the automatically assigned URL by selecting the *Generate new URL* option in the menu.

Use the available options to customize the URL based on your display preferences. These options are transient and you can use multiple variations of the URL at the same time.



The screenshot shows the 'Public access' configuration panel in eagle.io. At the top, there is a toggle switch for 'Public access' which is currently turned 'On'. To the right of this toggle is a help icon (question mark). Below the toggle, there are several configuration options:

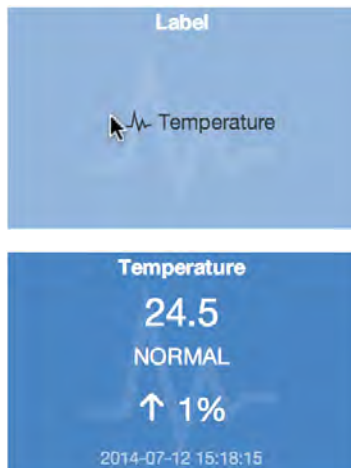
- URL:** A text input field containing the URL `https://public.eagle.io/public/chart/nzjbx01ii4mcs`. To the right of the input is a 'COPY' button and a small downward arrow.
- Format:** A dropdown menu currently set to 'HTML - Interactive'.
- Header:** A dropdown menu currently set to 'Show logo'.
- Timezone:** A dropdown menu currently set to 'Auto-detect'.

Below these options is a checkbox labeled 'Ignore Daylight Savings Time (fixed UTC offset)', which is currently unchecked. At the bottom of the panel, there is a light blue informational box with an information icon and the text: 'The link provides anonymous access to this chart'.

**Note:** This feature is not included in *Starter* accounts.

## Drag and Drop Links

Nodes can be dragged with a mouse directly from the Workspaces Tree and dropped onto applicable Tiles (indicated with drop-accept and drop-blocked mouse cursor icons).



## Copy/Paste Dashboards



Copy/Paste operations can be performed from the Workspaces Tree to quickly duplicate one or more dashboards. Links are automatically re-assigned to corresponding Nodes relative to the Dashboard if they exists at the destination. ie. You can copy/paste complete Locations containing Dashboards without manually re-assigning all Links.

## Move Dashboards



Moving a Dashboard within a Workspace will keep existing links. Moving a Dashboard between Workspaces will invalidate any Link where the linked node is not available in the destination Workspace.

## Link error



If a linked Node is removed or not accessible after a copy/paste or move operation then the Tile will display the Link error icon to indicate a problem with one or more Links. Links can be removed or reconfigured from the Links section in the Tile properties dialog.



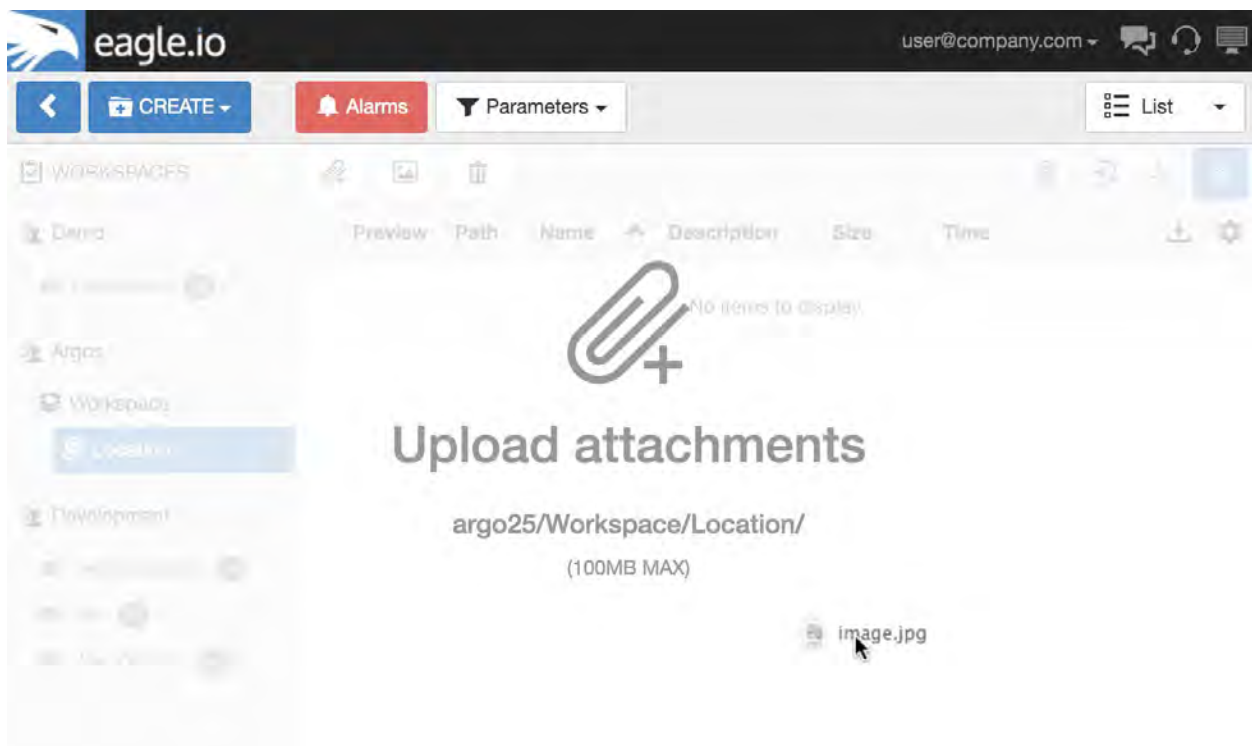
### 2.3.6 Attachment



Attachments include images, documents and other files stored in [eagle.io](https://eagle.io) (<https://eagle.io>). New attachments can be manually uploaded by a user or acquired via an [Attachment Source](#) (page 109). Attachments can be added inside Workspaces, Locations, Groups and Attachment Sources.

To view attachments navigate to *List View* → *Attachments*. You will see a list of all attachments within the Selected Node. Requires *view attachments* permission.

To upload attachments click the *upload attachments* button from the [attachments list view](#) (page 32) toolbar or drag and drop files directly from your desktop into the browser. Attachments will be created in the currently Selected Node.



**Note:** Maximum file size for an attachment is 100MB. *Configure* permission is required to upload an attachment.

## Properties



Configure properties via the Workspaces Tree context menu or List View properties icon. Requires *configure* permission.

### General

The General section allows you to enter a *description* for the attachment which is displayed in List View.


#### Description


Hardware maintenance and operation guide.

### Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

 Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

Groups		
Name		Role
Inspection Team	 3 users	No Access 

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

## Public access


Enabling public access allows anyone to view the node (attachment, chart, dash, report or historic data) using the generated *URL*.

You can also embed charts and dashboards into your own website with basic HTML. Select *Copy embed HTML* from the URL dropdown menu for a generic code snippet (you will need to paste this in to your website). You can also revoke and replace the automatically assigned URL by selecting the *Generate new URL* option in the menu.

Use the available options to customize the URL based on your display preferences. These options are transient and you can use multiple variations of the URL at the same time.

Public access

On



URL

<https://public.eagle.io/public/chart/nzjbx01ii4mcsm>

COPY

▼

Format

HTML - Interactive

⬆

⬇

⬆

Header

Show logo

⬆

⬇

⬆

Timezone


Auto-detect

⬆

⬇

⬆

☐ Ignore Daylight Savings Time (fixed UTC offset)

 The link provides anonymous access to this chart

---

**Note:** This feature is not included in *Starter* accounts.

---

### 2.3.7 Attachment Source



Attachment Sources have the ability to automatically acquire or receive *attachments* (page 105). Attachment Sources can be created inside Locations only.

---

**Note:** There is a maximum limit of 150 files that can be received per Attachment Source per day. This equates to 1 file every 10 minutes. Each file can be up to 100MB in size. Exceeding this limit will trigger an *Overload Alarm* (page 173) on the Source.

---

#### Properties



Configure properties via the Workspaces Tree context menu or List View properties icon. Requires *configure* permission.

#### General

The general section allows you to specify attachment storage options.

<input checked="" type="checkbox"/> Source Enabled		?
<input checked="" type="checkbox"/> Max attachments	10	?
<input checked="" type="checkbox"/> Remove after	30 days	?





**Source Enabled** Uncheck the Source Enabled checkbox to stop collection and prevent alarms and notifications from triggering.

**Max attachments** Specify the maximum number of attachments allowed within this Source. When new attachments arrive and the limit is exceeded, the oldest attachments are removed first. The maximum limit that can be set is 100 attachments per Source.

**Remove after** Enable this setting to specify the maximum number of days that attachments will be stored within this Source. The expired attachments will only be removed when new attachments arrive.

## Connection

Configure how you would like to connect to your attachments.

<b>Transport type</b>	<input type="text" value="Email attachments"/>	
<b>Email address</b>	<input type="text" value="upload+cable-gram-pear@eagle.io"/>	<input type="button" value="change"/>
<b>Sender address filter</b>	<input type="text" value="user@company.com or @company.com or leave blank for no restricti"/>	
<b>Attachment types</b>	<input type="text" value="comma separated list of file extensions or leave blank for no restrictio"/>	
<b>Process email body</b>	<input type="text" value="No"/>	

### Transport type

Select the transport used to acquire attachments.

#### *Email to eagle.io*

Email attachments using the auto-generated address exactly as shown.

**Note:** The maximum accepted size per email (including all attachments) is 25MB.

#### *Upload to ftp.eagle.io*

FTP your files to ftp.eagle.io using the auto-generated user name exactly as shown. Password is optional. Use Tcp port 21 for standard Ftp and Tcp port 990 for implicit SSL.

**Note:** Only one concurrent ftp connection is allowed per Source. The maximum accepted size per file is 100MB.

#### *Upload to sftp.eagle.io*

SFTP your files to sftp.eagle.io using the auto-generated user name exactly as shown. Password is optional. Use Tcp port 22.

**Note:** Only one concurrent sftp connection is allowed per Source. The maximum accepted size per file is 100MB.

**Email address** Email your attachments to this address. Click *change* to generate a new email address.

**Ftp user** The user name to use to login to the Ftp server. Click *change* to generate a new Ftp user.

**Sender address filter** For added security you can filter attachments by sender email address. Restrict to a specific email address or to a specific domain. eg. `user@company.com` or `@company.com`. Leave blank for no restriction.

**Attachment / File types** Restrict which file(s) get processed as Attachments by specifying a comma separated list of file extensions. eg. `jpg, png, mp4` will process standard media files and ignore all other file types.

**Process email body** Specify if the email message body should also be processed:

*No* only the attachments will be processed.

*Plain Text* will store the text content of the email in addition to any attachments.

*HTML* will store the HTML content of the email in addition to any attachments.

**Communications Alarm** See [Alarms and Notifications](#) (page 170).

## Collection

**Outdated Alarm** See [Alarms and Notifications](#) (page 172).



**Overload Alarm** See [Alarms and Notifications](#) (page 173).

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

Groups		
Name		Role
Inspection Team	 3 users	No Access 

---

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

---

## Public access

Enabling public access allows anyone to view the node (attachment, chart, dash, report or historic data) using the generated *URL*.

You can also embed charts and dashboards into your own website with basic HTML. Select *Copy embed HTML* from the URL dropdown menu for a generic code snippet (you will need to paste this in to your website). You can also revoke and replace the automatically assigned URL by selecting the *Generate new URL* option in the menu.

Use the available options to customize the URL based on your display preferences. These options are transient and you can use multiple variations of the URL at the same time.



Public access

On

URL

https://public.eagle.io/public/chart/nzjbx01ii4mcs

COPY

Format

HTML - Interactive

Header

Show logo

Timezone

Auto-detect

☐ Ignore Daylight Savings Time (fixed UTC offset)

The link provides anonymous access to this chart

**Note:** This feature is not included in *Starter* accounts.

### 2.3.8 Data Source (file)



Data Sources automatically acquire or receive timeseries data using a variety of different transport options. Connect to a [data logger](#) or collect data from files. Data Sources can be created inside Locations only.

The type of Data Source (data logger or file) is selected at time of creation and cannot be changed. This section is specific to file Data Sources.

---

**Note:** Eagle.io supports the acquisition and storage of up to 20000 records per [Data Source](#) (page 134) per day. Exceeding this limit will trigger an [Overload Alarm](#) (page 173) on the Source. Refer to [Historic Data Limits](#) (page 579) for more information.

---

#### Properties



Configure properties via the Workspaces Tree context menu or List View properties icon. Requires *configure* permission.

#### General

The general section allows you to specify the type file(s) you would like to collect.

☒ **Source Enabled** ?

**File type** Delimited Text ?

**Write mode** Merge and overwrite ?

**Source Enabled** Uncheck the Source Enabled checkbox to stop collection and prevent alarms and notifications from triggering.

**File type** Select the type of file(s) you would like to collect. The file type must be selected during datasource creation and can not be changed.

##### *Delimited Text*

Data is transmitted as rows of values separated with a specific [delimiter](#) character.

### JSON Time Series

Data is transmitted in the *JSON Time Series* (page 572) format. Note: JTS files to be acquired must contain a *columns* header that specifies *name* and *dataType* of each column included in the *data*.

### Custom Format

Data is transmitted as binary or text data (in any format) and transformed by a *Converter* (<https://github.com/eagle-io/converter>) into *JSON Time Series* (page 572) format. A *Converter* can be constructed and submitted for approval via the *Converter* (<https://github.com/eagle-io/converter>) repository. Once accepted, your custom format will appear in the *File type* list. Note: the eagle.io team can assist with developing a *Converter* for your custom format.

**Write mode** Select how acquired data will be written:

*Merge and overwrite* will insert the acquired data into the existing historic data and overwrite the existing values when timestamps match.

*Merge and preserve* will merge the acquired data into the existing historic data and will not overwrite existing values when timestamps match.

*Replace existing* will remove all existing historic data within the range of data being acquired.

## Connection

Configure how you would like to connect to your file(s).

<b>Transport type</b>	Read from Dropbox	?
<b>Dropbox account</b>	demo@eagle.io	Change
<b>Read mode</b>	Read from last acquire	?
	<input type="checkbox"/> Delete files after acquire	
<b>Retry attempts</b>	3	?
<b>Retry delay</b>	15 minutes	?

**Transport type** Select the transport used to acquire files:

*None*

Select this option when data will arrive exclusively via the *HTTP API* (page 433) or *manually imported* (page 216).

#### *Download from FTP site*

Connect to an external FTP Server to acquire data files. Note: FTP is supported in *passive mode* ([http://en.wikipedia.org/wiki/File\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/File_Transfer_Protocol)) only.

**FTP host** Host name or IP address of the FTP Server.

**Port** TCP Port to use for connection to the FTP Server. Default Port 21 for standard FTP/FTPS or 990 for FTP with Implicit SSL.

**Mode** FTP Secure connection options. Use *FTP* for standard connection or choose an available *FTPS* mode. Consult your network administrator to verify the settings required for connection to your FTP Server.

**Remote path** Specify the base remote path on the FTP Server that you have access to. You can retrieve files from sub-directories within this path.

**User** Specify the name of the user account for the system to use when connecting to the FTP Server. Enter *anonymous* if no user account is required.

**Password** Password for the associated user account (or leave blank for none).

#### *Download via HTTP*

No configuration required.

The File URL (when adding file) must start with *http://* or *https://*. Optionally specify the login username and password as part of the URL if HTTP Basic authentication is required. eg. *http://user:password@company.com/file.csv*.

Parts of the URL can be dynamically generated by including JavaScript expressions surrounded by double braces (`{{ expr }}`).

Processing & Logic *Global Variables* (page 226) can be referenced, along with the `T( expr )` function for time manipulation.

An example URL with expressions to request the most recently acquired data through till the most recently available:

```
http://data.com/?start={{T(SOURCE.currentLocalTime).format(
  'YYYY.MM.DD')}}&end={{T(NOW).format('YYYY.MM.DD')}}
```

Outputs a URL similar to

```
http://data.com/?start=2015.01.01&end=2017.12.01
```

#### *Upload via HTTP*

Data can be submitted as an HTTP POST body to the automatically generated URL. Password is optional.

**IP whitelist** You can optionally restrict incoming connections to this source to a list of approved IP addresses specified using CIDR notation. eg. 192.168.7.52/32  
Leave empty for no IP address restrictions.

**Note:** Please consider using the [API](#) (page 433) for more options uploading via HTTP.

#### *Email to eagle.io*

Email data using the auto-generated email address exactly as shown.

**Sender address filter** For added security you can filter by sender email address. Restrict to a specific email address or to a specific domain. eg. [user@company.com](#) or @company.com. Leave blank for no restriction.

**Note:** The maximum accepted size per email (including all data files) is 25MB.

#### *Send SMS text message*

Send a text message to one of our incoming SMS phone numbers:

**Australia:** +61 488 811 086

**United States:** +1 408 400 3928

**Custom phone numbers:** Please contact us to arrange a custom incoming SMS number to be exclusively associated with your account. This will allow you to choose your own Source Id.

**SMS messages must contain exactly 2 lines with the following content:**

- Source Id
- Data payload

#### *Publish to mqtt.eagle.io*

Publish data with MQTT using the following settings:

**Broker address** mqtt.eagle.io

**Broker port** Use port 1883 for standard connection or port 8883 for SSL.

**Topic** Use the auto-generated topic exactly as shown. eg. *io/eagle/source/fruit-honey-jacket*

**MQTT Password** Optional password (leave blank for none).

**IP whitelist** You can optionally restrict incoming connections to this source to a list of approved IP addresses specified using CIDR notation. eg. 192.168.7.52/32  
Leave empty for no IP address restrictions.

**Note:** The maximum accepted size per mqtt message is 10MB.

#### *Read from Amazon S3*

Connect to an Amazon S3 bucket to acquire data files.

**Bucket** Unique S3 bucket name. eg. my.aws.bucket

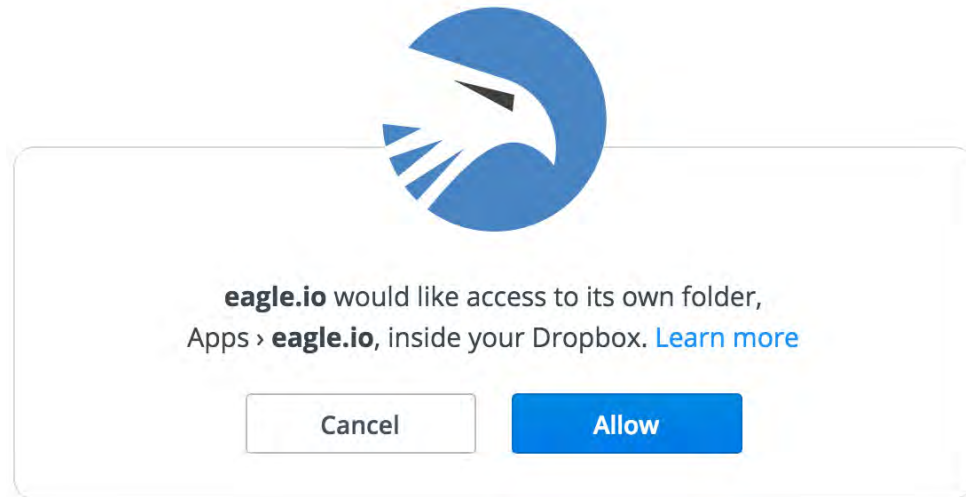
**Access key Id** User access key Id (generated from AWS Console).

**Secret** Secret token associated with access key.

**Note:** Matching files will be removed from S3 after acquire.

#### *Read from Dropbox*

Dropbox is used to connect to a [Dropbox](http://dropbox.com) (<http://dropbox.com>) account. An *eagle.io* folder will be created in your Dropbox *Apps* directory where you can place files for collection.



When changing the account, a popup window will be displayed which allows you to login to Dropbox and authorise access as shown above.

#### *Upload to ftp.eagle.io*

FTP your files to `ftp.eagle.io` using the auto-generated user name exactly as shown. Password is optional. Use Tcp port 21 for standard Ftp and Tcp port 990 for Implicit SSL.

**IP whitelist** You can optionally restrict incoming connections to this source to a list of approved IP addresses specified using CIDR notation. eg. `192.168.7.52/32`  
Leave empty for no IP address restrictions.

**Note:** Only one concurrent ftp connection is allowed per Source. The maximum accepted size per file is 100MB.

#### *Upload to sftp.eagle.io*

SFTP your files to `sftp.eagle.io` using the auto-generated user name exactly as shown. Password is optional. Use Tcp port 22.

**IP whitelist** You can optionally restrict incoming connections to this source to a list of approved IP addresses specified using CIDR notation. eg. `192.168.7.52/32`  
Leave empty for no IP address restrictions.

**Note:** Only one concurrent sftp connection is allowed per Source. The maximum accepted size per file is 100MB.

**Read mode** Select how data should be read from the file:

*Read all records* will read in the entire file on each acquisition.

*Read new records* will only read new data added to the file since last acquisition.

Optionally delete files after acquire (not supported on all transports).

**Retry attempts** Select how many retry attempts are made during an acquisition or scheduled collection before the communications is considered failed.

**Retry delay** Select the delay between retry attempts. The retry attempts multiplied by the retry delay should not exceed the collection interval.

**Communications Alarm** See [Alarms and Notifications](#) (page 170).

## Collection

Collection is used to specify if and when data should be automatically collected from the Source.



☒ **Scheduled** ?

**Collection interval** 5 minutes ?

**Interval offset** Automatic ?

**Scheduled** Enable the schedule for automatic data collection.

**Collection interval** Select the frequency of the collection.

**Interval offset** Offset the collection interval to prevent collection occurring on the interval. eg. 1 hour collection interval will occur at 9am, 10am, etc. Set an interval offset of 5 minutes and the collection times will be 9:05am, 10:05am, etc.

**Outdated Alarm** See [Alarms and Notifications](#) (page 172).

**Overload Alarm** See [Alarms and Notifications](#) (page 173).


## Series


The Series section allows you to assign Series from the Source to New or Existing Parameters.

The screenshot displays the eagle.io configuration interface. At the top, there are three dropdown menus for 'Location coordinates': 'all\_week.csv' for the main location, and 'latitude' and 'longitude' for the respective coordinates. Below this is a table of sensor series. The first series is 'all\_week.csv'. The second series is 'aquarium-sensors.csv', which has a context menu open showing options: 'Modify file match', 'Configure series', 'Set acquire time', and 'Remove file'. The table lists five series: pH, Temperature, Conductivity, and Redox. Each series has a 'New' or 'Existing' button and a parameter assignment field. The 'Conductivity' and 'Redox' series also have unit dropdowns set to 'mS' and 'mV' respectively. At the bottom, there is an 'Add files' button.


**Add Files** Add new files to be acquired by the Source (10 files maximum). A list of files will be displayed which are available using the current connection options. Select the new files to be added. You will need to setup the parser configuration for each file added.


**Location Coordinates** The Source can optionally provide Location Coordinates from a coordinate series or a pair of numeric series (lat/lng). You need to configure the Location and select this Source to be used for its coordinates. Select the file and series to use from the drop down list.

**Modify file match**  Modify the file configuration and matching criteria and select a sample file to use for configuring series. Some transport types such as Email or FTP/SFTP Server require you to upload a local sample file or alternatively *Skip or Reset sample file and retrieve on next acquisition*. The *file name match* should uniquely match the name of uploaded file(s). For example use *.csv* to match all files with a csv extension (case-insensitive). Enable the *Match using regular expression* option for advanced matching.

**Configure series**  Setup or Update the parser configuration for the selected file. See [Text Parser Configuration](#) (page 123) for full details. If a sample data file has not been uploaded or acquired the Configure series menu item will be disabled.



**Set acquire time**  Set or reset the collection pointer for the corresponding table to re-collect its data from the specified date using the Write mode option selected in the General tab.

**Remove file**  Select this option from the drop down table menu to remove this file from the Source. All parameters associated with the series from the file will be disabled.

**Parameter Assignment** You must assign Series to New or Existing parameters and set the Series for collection by ensuring its checkbox is enabled. Any Parameters assigned to Series will be disabled when the Series is unchecked for collection.

Parameters can be re-assigned to new Series at any time without losing existing historic data.

The series icon indicates the *type of parameter* (page 49) that will be created. Rename and Delete operations should be performed from the Workspaces Tree.

## Historic

Click the *Delete* buttons and save to permanently remove all *historic data* or *events* for Parameters contained within this Data Source. Alternatively use the *Parameter Historic* (page 157) section to delete historic data or events for individual Parameters.

**Historic data**

This will permanently remove all historic data for parameters contained within this data source.

**Events**

This will permanently remove all events for this data source and all parameters it contains.

## Time

Time allows you to configure the timezone of the Source and associated options.

Timezone

(GMT+10:00) Eastern Time - Melbourne, Sydney

?

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

**Timezone** Select the timezone the Source uses for storing historic data. Choose *(GMT+00:00) UTC* if data timestamps are in UTC.

**Ignore Daylight Savings Time** Check this option if the Source does not offset the data timestamps to account for DST.

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

Groups		
Name		Role
Inspection Team	<i>i</i> 3 users	No Access

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

## Text Parser Configuration

The Text Parser allows you to define how a delimited text file should be processed by the system including defining series to be assigned to Parameters. All scheduled collection and user acquisition requests will use the saved parser configuration to process any new data that has been appended to the file since last collection.

Parser configuration sample/sensors-1.csv

Column delimiter: | Format: Column series Labels row: None Units row: None Text qualifier: ' Text encoding: UTF-8 (default) Match: Smart Reset to defaults

	Column 1 Record Time	Column 2 pH Number	Column 3 Temperature Number	Column 4 Conductivity Number	Column 5 Redox Number
1	Timestamp	pH	Temperature	Conductivity	Redox
2	2017-01-11 03:07:05	7.38	28.7	42.8	-461
3	2017-01-11 03:08:05	7.38	28.7	42.8	-461
4	2017-01-11 03:09:05	7.38	28.7	42.9	-461
5	2017-01-11 03:10:05	7.38	28.7	42.9	-461
6	2017-01-11 03:11:05	7.38	28.8	42.9	-461
7	2017-01-11 03:12:05	7.38	28.8	42.9	-461
8	2017-01-11 03:13:05	7.38	28.8	42.9	-461
9	2017-01-11 03:14:05	7.38	28.7	42.9	-461
10	2017-01-11 03:15:05	7.38	28.7	42.7	-461
11	2017-01-11 03:16:05	7.38	28.7	42.7	-461

Get help with parser configuration

Apply Cancel

## General Settings

General settings are used to specify file format and encoding options.

Column delimiter	Format	Labels row	Units row	Text qualifier	Text encoding	Match	
,	Column series	None	None	"	UTF-8 (default)	Smart	Reset to defaults

**Column delimiter** Specify the *delimiter* used to separate values in each line of the file. Select a delimiter from the drop down or enter a custom character to use.

**Format** Select the record format of the file. *Column series* should be used when each column in the file represents a single series or sensor. *Row series* should be used when each row (line) in the file represents a single series or sensor.

**Labels row** If your file contains a labels header row you can specify the corresponding line number or click the row number in the Parser Preview. The corresponding row will be highlighted and the column values will be used as the column labels. Labels are shown for convenience, but you can assign any valid names to the parameters during or after creation.

**Units row** If your file contains a units header row (and your using *Column series* format) you can specify the corresponding line number or right-click the row number in the Parser Preview. The corresponding row will be highlighted and the associated column values will be used as the default units. You can change the units during or after parameter creation.

**Text qualifier** Specify the character used to surround text values. This character will be automatically stripped if it exists in a column. Select a qualifier from the drop down or enter a custom character to use.

**Text encoding** The encoding format of the text file. If you notice special characters displayed incorrectly in the Parser Preview, you may need to change this setting.

**Match** *Smart* match will intelligently extract any values from the field that matches the column data type. *Strict* will only extract a value from the field if the entire field contents matches the column data type.

**Success required** When parsing a file via its configured transport, a Data alarm will be raised on the Source (when enabled) if the number of records successfully parsed do not reach this percentage threshold. Defaults to None (0%). ie. do not raise any alarm.

**Reset to defaults** Resets parser configuration and joins to system defaults.

## File Contents

The Parser extracts a sample from the beginning of the input text file and attempts to split the file into columns based on the current Column delimiter and Format.

## Column Series Format

This format should be used when each column represents a single series or sensor. Every row is considered a record and must contain a timestamp.

*Input Text File*

```

Timestamp,pH,Temperature,Conductivity,Redox
2017-01-11 03:07:05,7.38,28.7,42.8,-461
2017-01-11 03:08:05,7.38,28.7,42.8,-461
2017-01-11 03:09:05,7.38,28.7,42.9,-461
2017-01-11 03:10:05,7.38,28.7,42.9,-461
2017-01-11 03:11:05,7.38,28.8,42.9,-461
2017-01-11 03:12:05,7.38,28.8,42.9,-461
2017-01-11 03:13:05,7.38,28.8,42.9,-461
2017-01-11 03:14:05,7.38,28.7,42.9,-461
2017-01-11 03:15:05,7.38,28.7,42.7,-461
2017-01-11 03:16:05,7.38,28.7,42.7,-461
2017-01-11 03:17:05,7.39,28.6,42.6,-461
2017-01-11 03:18:05,7.38,28.6,42.6,-461

```

### Parser Preview

Timestamp, pH, Temperature, Conductivity, Redox					
<div> <div>Timestamp</div> <div>Record Time</div> <div></div> </div> <div> <div>pH</div> <div>Number</div> <div></div> </div> <div> <div>Temperature</div> <div>Number</div> <div></div> </div> <div> <div>Conductivity</div> <div>Number</div> <div></div> </div> <div> <div>Redox</div> <div>Number</div> <div></div> </div>					
LABELS	Timestamp	pH	Temperature	Conductivity	Redox
2	2017-01-11 03:07:05	7.38	28.7	42.8	-461
3	2017-01-11 03:08:05	7.38	28.7	42.8	-461
4	2017-01-11 03:09:05	7.38	28.7	42.9	-461
5	2017-01-11 03:10:05	7.38	28.7	42.9	-461
6	2017-01-11 03:11:05	7.38	28.8	42.9	-461
7	2017-01-11 03:12:05	7.38	28.8	42.9	-461
8	2017-01-11 03:13:05	7.38	28.8	42.9	-461
9	2017-01-11 03:14:05	7.38	28.7	42.9	-461
10	2017-01-11 03:15:05	7.38	28.7	42.7	-461
11	2017-01-11 03:16:05	7.38	28.7	42.7	-461

### Row Series Format

This format should be used when each row (line in file) represents a single series or sensor. Every row must contain a timestamp, unique identifier for the sensor (Series ID) and a value. The parser preview will only display rows with unique Series ID's. The sample input file may not contain all the possible Series ID's, so you can click the *Add new series* button and enter additional Series ID's as required. Ensure the correct data type is set for all series.

### Input Text File

```

Timestamp,SensorID,Value
2017-01-11 03:07:05,pH,7.38
2017-01-11 03:07:05,Temp,28.7
2017-01-11 03:07:05,Cond,42.8
2017-01-11 03:07:05,Redox,-461
2017-01-11 03:08:05,pH,7.38
2017-01-11 03:08:05,Temp,28.7
2017-01-11 03:08:05,Cond,42.8
2017-01-11 03:08:05,Redox,-461
2017-01-11 03:09:05,pH,7.38
2017-01-11 03:09:05,Temp,28.8
2017-01-11 03:09:05,Cond,42.9
2017-01-11 03:09:05,Redox,-461

```

### Parser Preview

Timestamp		SensorID	Value
#	⌚ Record Time	🔑 Series ID	⌚ Value
LABELS	Timestamp	SensorID	Value
2	2017-01-11 03:07:05	pH	7.38
3	2017-01-11 03:07:05	Temp	28.7
4	2017-01-11 03:07:05	Cond	42.8
5	2017-01-11 03:07:05	Redox	-461
🗑		New	
		Add new series	

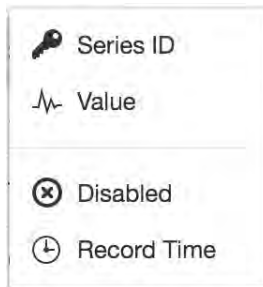
⌚ Number  
 🗑 Boolean  
 a Text  
 ⌚ Time  
 📍 Coordinates

## Columns and Data Type Selection

In *Column Series* format, columns are assigned a series data type which determines how they should be processed and the type of Parameter that will be available for creation.



In *Row Series* format, columns are assigned as *Record Time*, *Series ID*, *Value* or *Disabled*. The column that is assigned as the *Value* column has individual data types assigned per row.



Use the data type drop down menu in the column/row to select a data type from the available options. Individual cells that do not match the selected series data type are displayed in *RED*. Hover over a cell in the Parser Preview to display a tooltip (where valid) showing how the raw data will be interpreted by the parser.

## Series Data Types

**Number** Interpret data as numeric. Can be created as a Number Parameter and suitable for both analog and digital data.

**Boolean** Interpret data as Boolean only. Can be created as a special Number Parameter with predefined states and accepts boolean data as text (TRUE or FALSE) or numeric (0 or 1).

**Text** Interpret data as Text. Create a Text Parameter from the column.

**Time** Interpret data as Time. You need to specify the time format using the Joins and Parser Configuration section. You can create a Time Parameter from the column.

**Disabled** The column will be *GREY* in the Parser Preview and will not be available for Parameter creation. Setting an unused column to Disabled is optional. It has been added for convenience when working with files with lots of columns.



**Record Time** The column to use as the Record Time (timestamp for the line values). There must always be exactly one Record Time column defined. You need to specify the time format using the Joins and Parser Configuration section.

**Coordinates** Interpret column data as Coordinates (Latitude and Longitude). You can only specify one Coordinates column per file (but you can join data from other columns). Use the Joins and Parser Configuration section to specify the Latitude and Longitude fields. If Coordinates (latitude and longitude) are split over multiple columns, you can also just set the data type of the columns as Number and select the individual columns when configuring the Location Coordinates for the Source.

## Joins and Parser Configuration



The Joins and Parser Configuration section (available from the properties icon at the top of each column or in each row of the Value column for Row Series format) is used to customize the parser for individual columns/series including joins, formatting and specifying quality.

Data values in input files are commonly split into separate fields and therefore will be shown as different columns in the Parser Preview. It is necessary to ensure each column you would like to use for Parameter creation has all joins/fields defined.



Column joins and configuration (Column 1) Close

Column joins	Field	Format	Format filter
Column 1	Time	DD/MM/YYYY	
Column 2	Time	H:mm:ss A	
Column 1	★ Quality	format	<div>Include record</div> <div>Add join</div>

Column 1 + Column 2

# Record Time

1	Date Time
2	08/06/2013 4:01:00 PM
3	08/06/2013 5:01:00 PM
4	08/06/2013 6:01:00 PM
5	08/06/2013 7:01:00 PM

You can join additional columns via the *Add join* button. Select the column to join from the Joins drop down and the Field to be assigned. Columns are joined in the order they are displayed (top to bottom) and the result is shown in the Parser Preview.

Re-order any join/field by hovering over the item with a mouse to reveal the grab handle. Click and drag the grab handle to reposition the item in the list.

Column 3

★ Quality

X

You may need to split or exclude data within a single column. Add a Separator field and exclude specific text (as specified in the Separator format field).

Column joins and configuration (Column 2) Close

Column joins	Field	Format	Format filter
Column 2	<div>Latitude</div>		
	<div>Separator</div>	:	
	<div>Longitude</div>		

Column 2

★ Quality

format

Include record

Add join

Column 2

Coordinates

1	GPS	LATITUDE: 40.730610 LONGITUDE: -73.935333
2	40.730610;-73.935333	
3	40.730633;-73.935242	
4	40.730214;-73.935245	
5	40.731006;-73.935251	

When using *Row Series* format, column joins and data type for the *Value* column must be configured per series. Use the dropdown data type selector and properties toggle button for each series/row.

**Column joins and configuration** (Value: New) Close

Column joins	Field	Format	Format filter
Value	a Text	format	Include field
Value	★ Quality	format	Include field
<span>Add join</span>			

#

Value

⚡ Value

LABELS		Value
2	7.38	⚡
3	28.7	⚡
4	42.8	⚡
5	-461	⚡
	(New)	a

Quality codes can be associated with each column, either by splitting the column or joining an additional column containing the quality codes.

All columns can be re-used as joins on other columns. For example you can have multiple parameter columns that all share the same quality code value by adding the quality code column as a join on each parameter column.

Column joins and configuration (Column 3)

Close

Column joins	Field	Format	Format filter
Column 3	Number	format	Include record
Column 6	Quality	format	Include record

Column 3

Separator

:

Add join

Column 3 + Column 6

#

Number

1	Temperature Quality
2	28,5 100
3	28,5 50
4	28,5 50
5	28,5 100

You can optionally specify a **format** and **format filter** to include or exclude the field or the entire record based on the data matching the specified format. In the example below all records will be excluded where the data in *Column 6 (Quality)* equals 50.

**Column joins and configuration** (Column 3) Close

Column joins	Field	Format	Format filter
Column 3	Number	format	Include record
Column 6	Quality	50	Exclude record
Column 3	Quality	format	Include record <span>Add join</span>

**Column 3** **Column 6**

# Number

1	Temperature Quality
2	28.5 100
3	28.5 50
4	28.5 50
5	28.5 100

### 2.3.9 Data Source (data logger)



Data Sources automatically acquire or receive timeseries data using a variety of different transport options. Connect to a *data logger* or collect data from files. Data Sources can be created inside Locations only.

The type of Data Source (data logger or file) is selected at time of creation and cannot be changed. This section is specific to data logger Data Sources.

Refer to the *Device configuration* (page 289) section for a quick guide to connecting a range of industry standard data loggers.

---

**Note:** Eagle.io supports the acquisition and storage of up to 20000 records per *Data Source* (page 134) per day. Exceeding this limit will trigger an *Overload Alarm* (page 173) on the Source. Refer to *Historic Data Limits* (page 579) for more information.

---

#### Properties



Configure properties via the Workspaces Tree context menu or List View properties icon. Requires *configure* permission.

#### General

The general section allows you to specify the type of data logger you would like to connect to.

☒ **Source Enabled**

**Datalogger type** Campbell Scientific [PakBus]

**Model** CR6

**PakBus address** 1

**PakBus neighbor**

**Security code** \*\*\*\*

**Encryption key**

**TCP password**

**Write mode** Replace existing

## Source Enabled

Uncheck the Source Enabled checkbox to stop collection and prevent alarms and notifications from triggering.

## Data logger type

Select the type of data logger you would like to connect to:

*Campbell Scientific [PakBus]*

**Model** Select the model of the data logger.

**PakBus address** Default value is 1.

**PakBus neighbor** PakBus neighbor address when using PakBus routing. Leave blank when connecting directly to the end device.

**Security code** Optional security code pre-configured in the data logger. Leave blank for none.

**Encryption key** Optional PakBus encryption key pre-configured in data logger. Leave blank for none.

**TCP password** Optional PakBus TCP password pre-configured in data logger. Leave blank for none.

*Libelium, OTT netDL, SensorLog, WaterLog*

**Model** Select the model of the data logger.

## Write mode

Select how acquired data will be written:

*Merge and overwrite* will insert the acquired data into the existing historic data and overwrite the existing values when timestamps match.

*Merge and preserve* will merge the acquired data into the existing historic data and will not overwrite existing values when timestamps match.

*Replace existing* will remove all existing historic data within the range of data being acquired.

## Connection

Configure how you would like to connect to your data logger.

Transport type	<div>TCP Client</div>	?
Internet address	<div>eg. 152.10.234.44 or mydevice.dyndns.org</div>	
TCP port	<div>eg. 4128</div>	

## Transport type

The following transports are available for certain data loggers:

### *TCP Client*

TCP Client is used to establish an IP connection to a data logger connected to the internet.

**Internet address** IP address or Domain name where the device can be contacted.

**TCP port** TCP Port of the connected device.

### *TCP Server*



TCP Server is for use with WaterLog Storm3 data loggers, SensorLog and Waspnotes. You should configure your logger to establish a TCP connection to **device.eagle.io**

- WaterLog Storm3 should use port **32213**
- SensorLog should use port **80**
- Waspnotes should use port **5566**

**Site Id, Device Id, Waspnote Id** The unique identification code generated by the system for this Source. You should configure your device with this Id.

#### *TCP Callback*

TCP Callback allows the remote device to establish an IP connection to eagle.io using a unique identifier. The device should connect to **device.eagle.io** port **4332**.

**Device Id** The unique identification number generated by the system for this Source. You can use an existing device id if it was previously assigned to a Data Source and no longer in use.

You can also assign any valid IMEI number from a modem not already assigned to a Source.

#### *Upload to ftp.eagle.io*

The OTT netDL range of loggers use FTP to upload files to ftp.eagle.io. When configuring the logger use the auto-generated user name exactly as shown. Password is optional. Use Tcp port 21 for standard Ftp and Tcp port 990 for Implicit SSL.

**IP whitelist** You can optionally restrict incoming connections to this source to a list of approved IP addresses specified using CIDR notation. eg. 192.168.7.52/32  
Leave empty for no IP address restrictions.

**Note:** Only one concurrent ftp connection is allowed per Source. The maximum accepted size per file is 100MB.

#### *Scout*

Scout should be used with a *Scout modem* to enable all the device smart features.

**Device Id** The unique 15 digit IMEI number assigned to the *Scout modem*. Printed on the back of device.

#### *Meshlium*

Meshlium devices provide efficient routing of data collected from Waspnote devices in large Libelium sensor networks. The eagle.io application in the device needs to be started (and initial connection established with eagle.io) before it can be used as a transport.

**Meshlium Id** The unique serial number of the meshlium device.

**Auth key** Optional user specified authentication key configured in the eagle.io application running on the meshlium device.

**Waspnote** The Waspnote / Plug & Sense! device you would like to acquire data from.

---

**Note:** The same *Scout* or *TCPCallback* transport can be used with multiple data sources when using *PakBus neighbour* routing (and the remote data loggers can be reached). However all Data Sources sharing the transport must be configured in the same Workspace. Moving a Data Source will also move the associated device and invalidate any other Data Sources within the origin Workspace still configured to use the device.

---

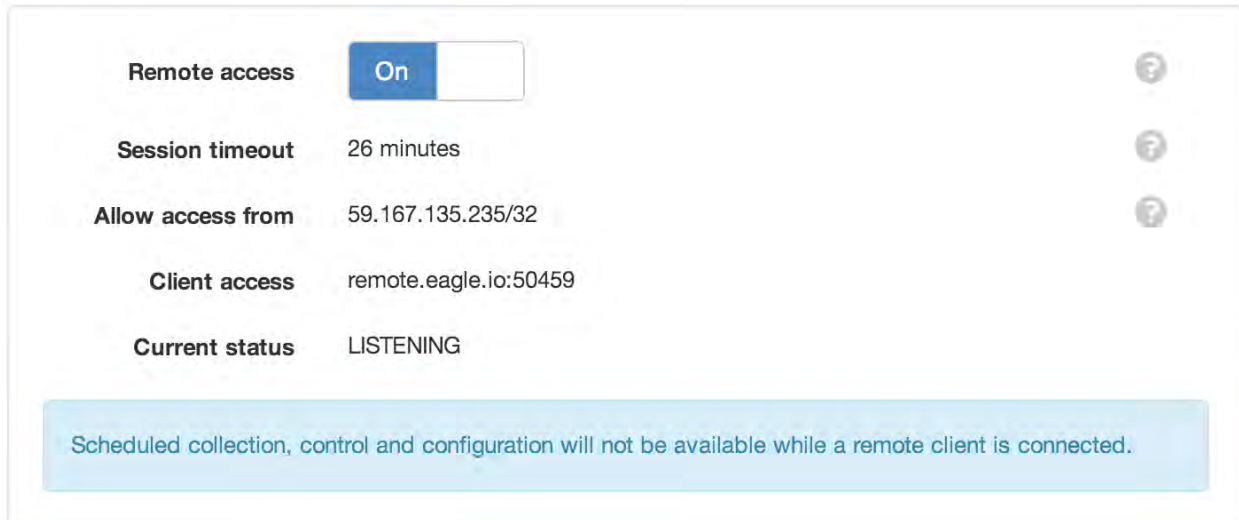
## Remote access

Remote access allows you to connect directly to your data logger over TCP/IP (TCP Callback and Scout transport only). An Internet address and Port will be temporarily allocated when enabling remote access. Scheduled collection, control and configuration will not be available while a remote client is connected, and the Source status will show *REMOTE*.

**Session timeout** Select a duration for the session, after which the remote access connection will be automatically closed.

**Allow access from** Restrict access to your data logger from your current IP address or a range of IP addresses specified using CIDR notation. Select *Anywhere* to allow connections from any Internet enabled device.

If the remote device is not connected or another client is already connected when you try to connect, a status message will be returned and remote access refused.



The screenshot shows a configuration panel for 'Remote access'. It includes a toggle switch set to 'On', a 'Session timeout' of 26 minutes, 'Allow access from' set to 59.167.135.235/32, 'Client access' set to remote.eagle.io:50459, and 'Current status' set to LISTENING. A light blue banner at the bottom states: 'Scheduled collection, control and configuration will not be available while a remote client is connected.'

Remote access	<input checked="" type="checkbox"/> On	?
Session timeout	26 minutes	?
Allow access from	59.167.135.235/32	?
Client access	remote.eagle.io:50459	
Current status	LISTENING	

Scheduled collection, control and configuration will not be available while a remote client is connected.

## Retry attempts

Select how many retry attempts are made during an acquisition or scheduled collection before the communications is considered failed.

## Retry delay

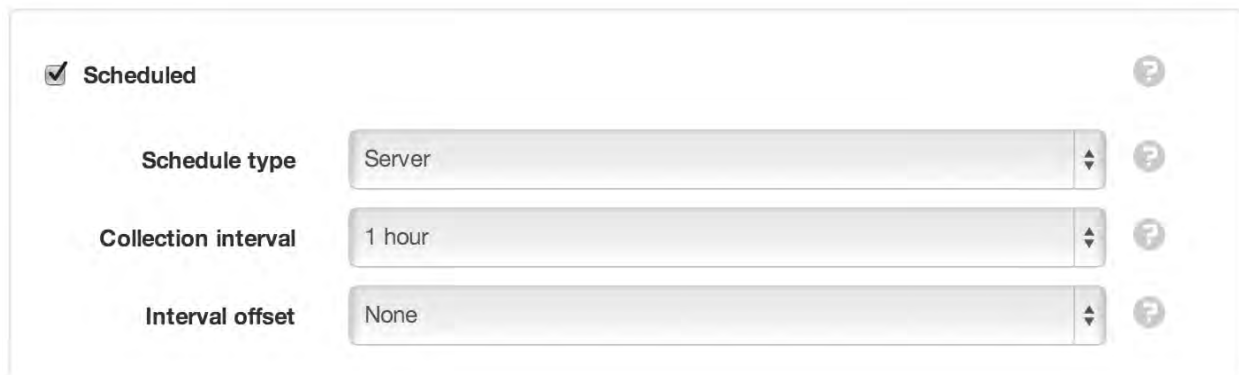
Select the delay between retry attempts. The retry attempts multiplied by the retry delay should not exceed the collection interval.

## Communications Alarm

See *Alarms and Notifications* (page 170).

## Collection

Collection is used to specify if and when data should be automatically collected from the Source. Not all connection transports will have a schedule.



☒ **Scheduled** ?

**Schedule type** Server ?

**Collection interval** 1 hour ?

**Interval offset** None ?

### Scheduled

Enable the schedule for automatic data collection. You should Disable the schedule when using the TCP Callback Transport for event based callbacks (ie. not permanently connected).

### Schedule type

*Server* is the recommended option for always connected devices and allows easy schedule configuration.

*Remote* enables a *Scout modem* to maintain its own schedule allowing for low power mode and customized configuration.

### Collection interval

Select the frequency of the collection.

### Interval offset

Offset the collection interval to prevent collection occurring on the interval. eg. 1 hour collection interval will occur at 9am, 10am, etc. Set an interval offset of 5 minutes and the collection times will be 9:05am, 10:05am, etc.

**Outdated Alarm** See [Alarms and Notifications](#) (page 172).

**Overload Alarm** See *Alarms and Notifications* (page 173).

### In-Sequence Collection

For *Campbell Scientific [PakBus]* dataloggers that share the same connection, data collection will automatically occur in sequence. The sequence is ordered from oldest to newest (the date the datalogger was created in eagle.io).

**In-Sequence Collection for TCP Callback** When using the *TCP Callback* transport, all dataloggers which have the same device id share the same connection, and these will also be collected in sequence whenever the device connects to eagle.io.

## Configuration

Configuration displays configuration set in the remote data logger including program name, firmware, etc. To send new configuration to a supported logger, use the *Manage configuration* (page 185) dialog.

Program name	CPU:DataTables.CR8
Firmware version	CR800.Std.27

**Configuration Alarm** See *Alarms and Notifications* (page 171).

## Series

The Series section allows you to assign Series from the Source to New or Existing Parameters. A list of available tables/series is shown (when not using Automatic parameter configuration) and automatically updated on every connection to the data logger.

Location coordinates

⊗ None

☐ Automatic parameter configuration

ALERT

Chk\_Info

Main

Series

Parameter assignment

Set acquire time

☒

WaterLevel

New
Existing

Main - WaterLevel

☒

Batt\_Volt

New
Existing

Main - Batt Volt

☒

Total\_Rain

New
Existing

Main - Total Rain

Public

Rain

Status

## Automatic Parameter Configuration

Check this option to automatically create, update and disable parameters based on the tables defined in the data logger. When checked, the Source tables will not be displayed.

### Note:

**Campbell Scientific data logger program changes will trigger the following automatic parameter configuration:**

- DataTable field names common between the old and new program will maintain existing parameter assignment
- DataTable field names absent from the new program will cause assigned parameters to be disabled (greyed out)

- DataTable field names newly defined will be associated with new parameters
- 

## Location Coordinates

The Source can optionally provide Location Coordinates from a coordinate series or a pair of numeric series (lat/lng). You need to configure the Location and select this Source to be used for its coordinates. Select the table and series to use from the drop down list.

## Configure series



Re-configure the series/sensors for specific data loggers only.

## Set acquire time



Set or reset the collection pointer for the corresponding table to re-collect its data from the specified date using the Write mode option selected in the General tab.

## Parameter Assignment

You must assign Series to New or Existing parameters and set the Series for collection by ensuring its checkbox is enabled. Any Parameters assigned to Series will be disabled when the Series is unchecked for collection.

Parameters can be re-assigned to new Series at any time without losing existing historic data.

The series icon indicates the *type of Parameter* (page 49) that will be created. Rename and Delete operations should be performed from the Workspaces Tree.

## Historic

Click the *Delete* buttons and save to permanently remove all *historic data* or *events* for Parameters contained within this Data Source. Alternatively use the *Parameter Historic* (page 157) section to delete historic data or events for individual Parameters.



**Historic data**

This will permanently remove all historic data for parameters contained within this data source.

**Events**

This will permanently remove all events for this data source and all parameters it contains.

## Time

Time allows you to configure the timezone of the Source and associated options.

<b>Timezone</b>	<div>(GMT+10:00) Eastern Time - Melbourne, Sydney</div>	<div>?</div>
	<input checked="" type="checkbox"/> Automatically adjust clock for Daylight Savings Time	
	<input checked="" type="checkbox"/> Automatically update datalogger with correct time	

## Timezone

Select the timezone the Source uses for storing historic data. Choose *(GMT+00:00) UTC* if data timestamps are in UTC.

### Automatically adjust clock for Daylight Savings Time

Check this option if you would like to automatically update the data logger clock to account for DST in the selected timezone.


### Automatically update data logger with correct time

Check this option if you would like to automatically update the data logger clock with the correct time based on the selected timezone.

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

 Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

 Groups		
Name		Role
Inspection Team	 3 users	No Access

---

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

---

### 2.3.10 Parameter



Parameters are managed by and contained within Data Sources. Parameters store the individual timeseries acquired by its Data Source. There are a few different types of parameters but essentially they support Monitoring or Controlling of values.

#### Types of Parameters

See [Nodes and Configuration](#) (page 49) for full list of Parameter types.

#### Properties



Configure properties via the Workspaces Tree context menu or List View properties icon. Requires *configure* permission.

---

**Note:** Available options will vary based on Parameter type.

---

#### General

The general section allows you to specify display settings for Number and Time Parameters.

Display type	<input type="text" value="Value"/>	
Default data format for map, charts and tabular data.		
Transform	<input type="text" value="Multiplier / Offset"/>	
Multiplier	<input type="text" value="1.0"/>	
Offset	<input type="text" value="0"/>	
Units	<input type="text" value="VDC"/>	
Format	<input type="text" value="0.###"/>	

**Display type** Select how this parameter should be displayed by default:

*Value* is the default display type for analog data.

*State* can be used for digital or discrete data and displays the state associated with the value (rather than the value).

**Transform** Optionally apply a Multiplier/Offset or Equation to the stored historic data for display, notifications and alarms.

*Multiplier* is applied to raw data for state evaluation and display.

*Offset* is a value applied to raw data for state evaluation and display. Note: Multiplier is always applied before offset.

*Equation* can be used to apply an advanced mathematical transformation of the value.

**Units** The units (suffix) to be displayed with the value or state.

**Format** Specify how the value should be formatted for display.

*Number Parameters* use this setting to set how many decimal places should be shown (leave empty to display the value with all included decimal places). *0* will always display a digit. *#* will only display a digit if it exists.

*Time Parameters* use this setting to determine the formatting of the timestamp (leave empty to display timestamps in your default user time format). Select a preset format from the drop down list or specify a custom format using *time format tokens* (page 583).

**Quality Alarm** See *Alarms and Notifications* (page 174).

## States

States allow you to define unlimited numbers of *Ranges* or *Discrete* thresholds that uniquely describe the current value of the parameter. Each state can optionally be configured to raise an alarm and/or send notifications.

---

**Note:** Number Parameters can use either Range or Discrete states. Text Parameters are limited to Discrete states only.

---

Click the Edit button to show and modify the configuration for each state. Use the *Add state* button to create a new state. States can be removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the state item).

## Range States

Drag and drop the NORMAL state in the list to change its order. The NORMAL state must always exist and can not be removed. Position states above or below the NORMAL state to capture higher or lower than normal values respectively.

States type	Range	?
Hysteresis	0.2	?

## States

Drag and drop NORMAL state to reorder

HIGH

State will be **active** when pH increases to **8** (2x). An **alarm** will be raised.  
State will become **inactive** when pH decreases below **7.8**.  
Notifications will be sent to **Maintenance** category when state becomes **Active or Inactive**.

Hide

Name
HIGH
?

Threshold
8
2x
?

☒ Raise alarm (parameter displayed red)

Notifications
Active or Inactive
?

Category
Maintenance
?

Active message
pH is at a dangerous level. Please take immediate action
?

Inactive message
Message when state becomes inactive. Leave blank for default.
?

NORMAL

State will be **active** when pH is between **6** and **8**.  
Notifications will not be sent.

Edit

LOW

State will be **active** when pH decreases to **6**.  
State will become **inactive** when pH increases above **6.2**.  
Notifications will be sent to **Operations** category when state becomes **Active or Inactive**.

Edit

Add state

## Discrete States

Drag and drop states in the list to change their display/evaluation order.

**States type**

Discrete
⬆️⬆️
?

**States**
Drag and drop to reorder

⚙️ **ON**
☰

State will be **active** when PUMP matches **1**.

Notifications will be sent to **Maintenance** category when state becomes **Active or Inactive**.

Hide

**Name**

ON

?

**Match**

1

1x ▼

?

☐
Raise alarm (parameter displayed red)

**Notifications**

Active or Inactive

?

**Category**

Maintenance

?

**Active message**

PUMP IS RUNNING

?

**Inactive message**

PUMP IS OFF

?

⚙️ **State 1**
☰

State will be **active** when PUMP matches **0**. An **alarm** will be raised.

Notifications will not be sent.

Edit

Add state

**States type** Select the type of States most suitable for your data.

*Range* is the default option for most analog data and allows you to define a State that covers a range

of values. eg. 1-10 = LOW

*Discrete* is most commonly used with digital data where a State represents a specific (or matching) value. eg. 0 = OFF, 1 = ON

**Hysteresis** Hysteresis is the amount the parameter value is required to decrease below the state threshold (states above *NORMAL*) or increase above the state threshold (states below *NORMAL*) for the active state to become inactive as it revert towards *NORMAL*. Hysteresis only applies to *Range States* and must be a number greater than or equal to 0.

**Name** Assign a unique name for each State. The *NORMAL* State (Range States) can not be renamed or removed.

**Threshold** States positioned above *NORMAL* will be active when the parameter value increases to this threshold. States positioned below *NORMAL* will be active when the parameter value decreases to this threshold. Optionally use the dropdown menu to specify the minimum number of **repeat values** at or beyond threshold required to trigger this state. Default *1x*.

Applies to *Range States* only.

## Match

**Number Parameters** Specify the numeric value to match.

**Text Parameters** Specify text contained within the value, or alternatively a [regular expression](https://en.wikipedia.org/wiki/Regular_expression) ([https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression)) which must begin and end with *forward-slash*, e.g. `/^(!message) .*$ /`

Optionally use the dropdown menu to specify the minimum number of matching **repeat values** required to trigger this state. Default *1x*.

Applies to *Discrete States* only.

**Raise alarm** Check this option to raise an [alarm](#) (page 170) when this state becomes active.

**Quality** Optionally select a [quality code](#) (page 266) to be applied to newly acquired data that matches this state.

**Notifications** Select when notifications should be sent.

*Active or Inactive* will send notifications when the state becomes active or inactive.

*Active* will only send notifications when the state becomes active.

*Inactive* will only send notifications when the state becomes inactive.

*Never* will not send notifications.

**Category** Select the category of subscribed users that will receive [notifications](#) (page 179) for this State.

**Active message** Enter a custom message to send when the state becomes active (may contain [expressions](#) (page 523)). Leave blank for a default message that includes the state name, value and timestamp.

**Inactive message** Enter a custom message to send when the state becomes inactive (may contain [expressions](#) (page 523)). Leave blank for a default message that includes the state name, value and timestamp.









## Chart

Charts section allows you to configure the display style and formatting of the parameter when displayed in the Charts View or shown on the adhoc Historic Chart.

## Style

Select the chart type and display preferences.

**Style**

<b>Chart type</b>	<div>Line Chart </div>	<div>Column Chart </div>	
<b>Line type</b>	<div>Normal </div>	<div>Step </div>	<div>Spline </div>
<b>Fill style</b>	<div>None </div>	<div>Solid </div>	<div>Gradient </div>

## Data

Select the data and processing to be used.



Data

Aggregation mode

Automatic

Custom

Raw

Aggregate

Total

?

Aggregate display

Complete intervals only

?

Period

☐ Automatic (based on zoom level)
☒ Fixed Period

?

Interval

1H

?

Base time

D

?

### Aggregation mode *Automatic*

Automatic is the recommended option for fast loading charts and uses *Interpolated* (or *Start* where Display type is State) aggregate and automatic period selection based on zoom level.

### *Custom*

Custom allows you to specify exactly how data should be aggregated for display on the charts.

**Aggregate** Aggregates provide a method to resample raw data over a specified period. See the [list of available aggregates](#) (page 569) for full details. Select RAW to display the original raw data with no aggregation applied (Interval and Base time options will be hidden).

**Aggregate display** Determines if aggregated data will be show only after intervals are complete, or be shown during an aggregation interval when only partial data is available.

**Period** *Automatic* will change the period of the aggregate calculation based on the zoom level of the chart. *Fixed Period* lets you define a fixed Interval and Base time that will be applied regardless of the zoom level.

**Interval** Specify the data range that is included for each calculated value. Example *30M* will calculate a value for each 30 minute window of data.

**Base time** Base time is used to determine the start of the aggregate calculations. Example *D+9H* will start the calculations at 9am (start of current day + 9 hours). See [working with relative time](#) (page 585) for full details.

### *Raw*

Raw does not perform any aggregation of data. This option should **ONLY** be used where you know the period to be displayed does not contain more than a thousand data points. Displaying high resolution raw data can impact on chart performance and in some cases may cause the chart to fail loading or have adverse affects on your web browser.

### **Baseline type (Custom Historic Charts only)** *Absolute*

Data point values are not modified.

### *Relative*

The first data point value in the requested time range is subtracted from all subsequent data point values.

## **Customize**

Select customization options for chart appearance.

Customize

Color

☐ Automatic
 ☒ Fixed Color

Primary color

#2f7ed8

Secondary color

#ffffff

Fill opacity

75%

Gradient style

Linear Top

Line width

2

Line style

Solid

Marker type

Automatic

Marker size

Hidden

Quality style

Marker (hover)

?

State thresholds

All States

?

3d effects

☐ Drop shadow

Reset to defaults

### Settings of Interest

**Color** Select *Fixed Color* to specify a color(s) to use for this parameter or *Automatic* to have it auto-assigned. If the Chart Fill Style is set to *Gradient* then you can also set a Secondary color.

**Line width** Set to *Hidden* to hide the line completely. If the Fill Style is set to *NONE* then you should ensure the Marker size is set.

**Marker position** Step charts can be displayed with the marker Left, Center or Right of the step. Left is the default option.

**Column placement** Column Placement applies to Column Charts only and determine where the columns should be positioned in relation to the x-axis. Options include *On Ticks* and *Between Ticks*.

**Marker size** Data point markers are hidden by default (shown on hover), but can be set to always display by specifying a size between 1-6 (small-large).

**Quality style** Select how quality colors are displayed on the chart. By default chart markers will only show quality color on hover.

**State thresholds** Select which State thresholds should be displayed on the chart:

***Hidden*** Hide all State thresholds.

***All States*** Show all State thresholds.

***Alarm States Only*** Show alarm State thresholds only.

***Non-Alarm States Only*** Show non-alarm State thresholds only.

***User Notification States Only*** Show thresholds for States that have a category assigned that the viewing user is subscribed to receive notifications from.

## Process

See *Processing and logic* (page 223).

## Rating

Rating parameters use an *Input parameter* and one or more ratings to calculate its data.

**Rating tables** should have multiple rows specifying an input *lookup* value and corresponding *result* value. Values that fall between each input lookup will be derived by linear interpolation.

**Rating equations** allow multiple coefficients to be defined which are used to calculate a value using a polynomial equation.

Multiple ratings can be added, with the *start* time used to determine the data range each rating will apply.

Input parameter
Level

Rating polynomial
2020-01-01 00:00:00 to 2020-10-19 23:59:59
Edit

Rating table
2020-10-20 00:00:00 onwards
Hide

Start
2020-10-20 00:00:00

Rating

Level (m)	Discharge (m <sup>3</sup> /s)	
0	0	
0.2	14.55	
0.8	55.04	
lookup	result	Add

Add rating
Import ratings
Export ratings

Ratings can be imported and exported using the fixed [ratings CSV](#) (page 521) format.

**Note:** Only Number parameters within the same Workspace (or [Template](#) (page 60)) can be used as the *Input parameter*.

## Historic

Click the *Delete* buttons and save to permanently remove all *historic data* or *events* for this Parameter.

**Historic data**

This will permanently remove all historic data for this parameter.

**Events**

This will permanently remove all events for this parameter.

## Control

The control section allows you to configure control options for Control Parameters.

<b>Restricted values</b>	<div>Range</div> <div>?</div>
<b>Minimum value</b>	<div>0</div>
<b>Maximum value</b>	<div>100</div>

**Restricted values** Select what values are acceptable for control operations:

*None*

None does not restrict the control value. User can enter any value of the same type as the parameter. eg. Number, Text, Time.

*Range*

Range restricts Number Parameter control values with a minimum and maximum value.

*States*


States restricts control values to the configured state threshold/match values. eg. ON / OFF.

**Control Alarm** See [Alarms and Notifications](#) (page 171).

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the [Workspace Security](#) (page 54) section.

 Users		
Name	Email	Role
John Citizen Last login 17 minutes ago	john@company.com	Owner
Simon Kelly Last login 19 days ago	s.kelly@company.com	Administrator
Peter Tennor Last login a day ago	peter@company.com	No Access

 Groups		
Name		Role
Inspection Team	 3 users	No Access

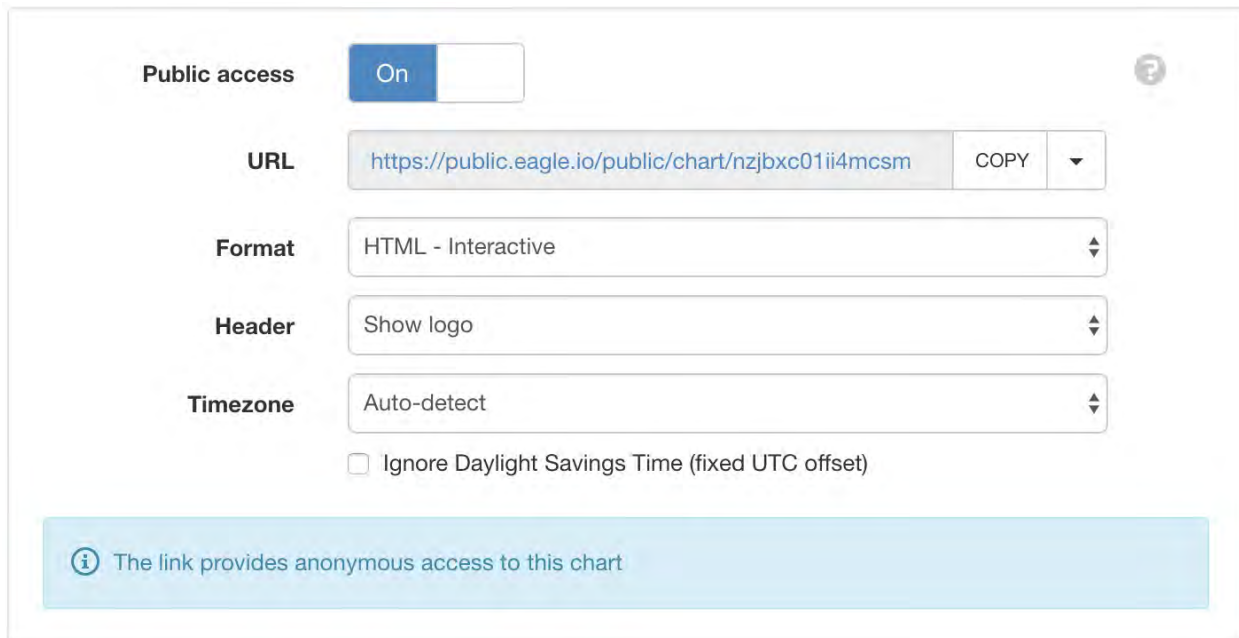
**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

## Public access

Enabling public access allows anyone to view the node (attachment, chart, dash, report or historic data) using the generated *URL*.

You can also embed charts and dashboards into your own website with basic HTML. Select *Copy embed HTML* from the URL dropdown menu for a generic code snippet (you will need to paste this in to your website). You can also revoke and replace the automatically assigned URL by selecting the *Generate new URL* option in the menu.

Use the available options to customize the URL based on your display preferences. These options are transient and you can use multiple variations of the URL at the same time.



The screenshot shows a settings panel for public access. At the top, 'Public access' is set to 'On' with a toggle switch and a help icon. Below this, the 'URL' field displays 'https://public.eagle.io/public/chart/nzjbx01ii4mcsn' with a 'COPY' button and a dropdown arrow. The 'Format' dropdown is set to 'HTML - Interactive'. The 'Header' dropdown is set to 'Show logo'. The 'Timezone' dropdown is set to 'Auto-detect'. Below the timezone dropdown is an unchecked checkbox labeled 'Ignore Daylight Savings Time (fixed UTC offset)'. At the bottom, a light blue informational box contains an information icon and the text 'The link provides anonymous access to this chart'.

---

**Note:** This feature is not included in *Starter* accounts.

---



### 2.3.11 Report



Report nodes generate PDFs and can be scheduled for delivery to subscribed users and groups via email. Please refer to the [Reports](#) (page 234) documentation for a quick start guide on using the reporting feature.

---

**Note:** The reporting feature is currently a **pre-release** version that is subject to change without notice and is only available to limited accounts. This documentation will also be updated prior to final release to expand on reporting features with examples.

---

#### Properties




Configure properties via the Workspaces Tree context menu or List View properties icon. Requires *configure* permission.

#### Links

Links assign Nodes in the Workspaces Tree to the report. They will appear under the **Tags** menu in the report editor and can be used to insert content into the report.

Click the *Edit* button to show and modify the configuration for each link. Use the *Add link* button to display the Workspaces Tree and select a new Node to link. Links can be removed via the dropdown menu (accessible by clicking the menu button on the top-right corner of the link item).


Reports can not have links to Nodes in different Workspace.

 **Displacement\_Chart**

nick48/Development/Displacement/Displacement Chart

Link will be available in report as Displacement\_Chart.


Edit

 **sitejpg**

nick48/Development/Displacement/site.jpg

Link will be available in report as sitejpg.

Edit

 **Historic\_Chart**

nick48/Development/Displacement/Historic Chart

Link will be available in report as Historic\_Chart.

Hide

**Link**

nick48/Development/Displacement/Historic Chart

change

?

**Name**

Historic\_Chart

?

Add link

## Schedule

Enable a daily, weekly or monthly schedule for the report to be generated and delivered to subscribed users and groups via email.

☒ **Scheduled**

?

**Email PDF**

Daily

Weekly

Monthly

MON

TUE

WED

THU

FRI

SAT

SUN

Run at

09

:

00

## Time

Time allows you to configure the timezone the Source should use for rendering timestamps.

Timezone

(GMT+10:00) Eastern Time - Melbourne, Sydney

?

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

**Timezone** Select the default timezone the Source should use for rendering timestamps.

**Ignore Daylight Savings Time** Check this option if the Source should not offset the timestamps to account for DST.

## Metadata


Metadata provides a simple key/value data structure that can be used to store static information about the location. Values can be saved as Number or Text and are displayed in the Locations List View.



Label	Value	Type	
Address	1600 Amphitheatre Parkway, Mounta	Text	
Service Tag	44301	Number	
<b>Add New</b>			
Label	Value	Text	Add

## Security

The Security section displays a list of all users and groups that have access to the Node. Users with *security* permission can restrict user and group access to the node and its descendants by assign the **No Access** role.

New users and groups must be added via the *Workspace Security* (page 54) section.

 Users		
Name	Email	Role
John Citizen <small>Last login 17 minutes ago</small>	john@company.com	Owner
Simon Kelly <small>Last login 19 days ago</small>	s.kelly@company.com	Administrator
Peter Tennor <small>Last login a day ago</small>	peter@company.com	No Access

 Groups		
Name		Role
Inspection Team	 3 users	No Access

---

**Note:** The Account Owner, Administrators or Workspace users/groups that have been assigned a Workspace role with the *configure* permission can not be assigned the *No Access* role.

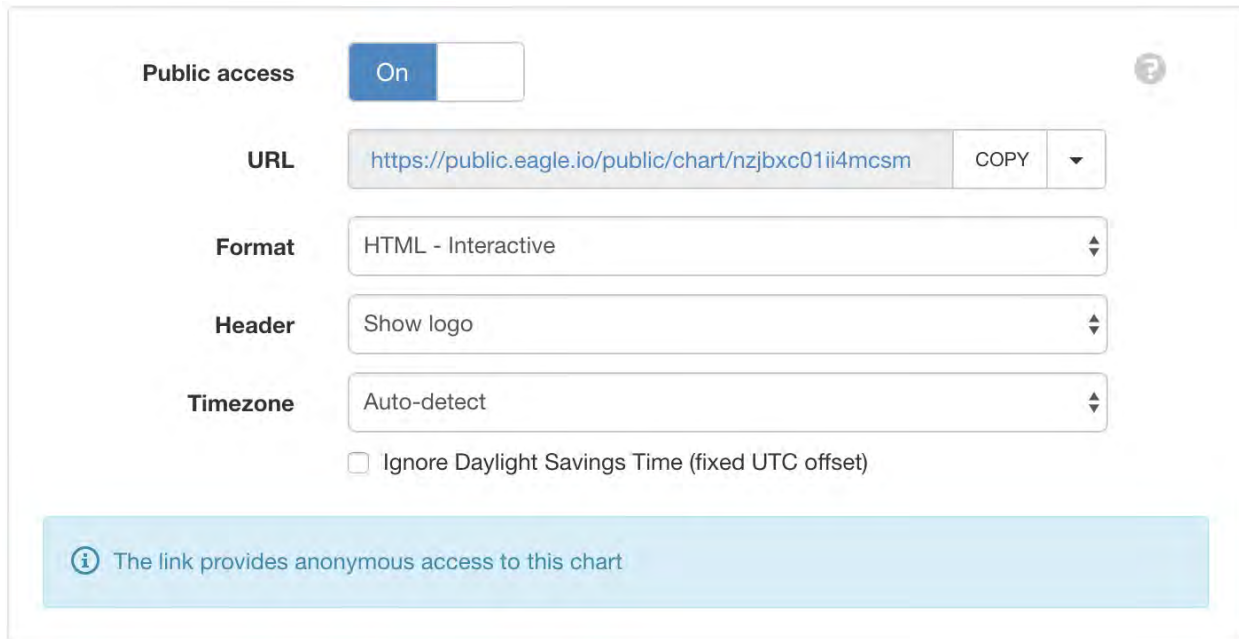
---

## Public access

Enabling public access allows anyone to view the node (attachment, chart, dash, report or historic data) using the generated *URL*.

You can also embed charts and dashboards into your own website with basic HTML. Select *Copy embed HTML* from the URL dropdown menu for a generic code snippet (you will need to paste this in to your website). You can also revoke and replace the automatically assigned URL by selecting the *Generate new URL* option in the menu.

Use the available options to customize the URL based on your display preferences. These options are transient and you can use multiple variations of the URL at the same time.



The screenshot shows a configuration panel for public access. At the top, 'Public access' is set to 'On' with a toggle switch. To the right is a help icon. Below this, the 'URL' field displays 'https://public.eagle.io/public/chart/nzjbx01ii4mcsn' with a 'COPY' button and a dropdown arrow. The 'Format' dropdown is set to 'HTML - Interactive'. The 'Header' dropdown is set to 'Show logo'. The 'Timezone' dropdown is set to 'Auto-detect'. Below the timezone dropdown is a checkbox labeled 'Ignore Daylight Savings Time (fixed UTC offset)'. At the bottom, a light blue informational box contains an information icon and the text: 'The link provides anonymous access to this chart'.

Public access ☒ On ?

URL <https://public.eagle.io/public/chart/nzjbx01ii4mcsn> COPY ▼

Format HTML - Interactive ▼

Header Show logo ▼

Timezone Auto-detect ▼

☐ Ignore Daylight Savings Time (fixed UTC offset)

i The link provides anonymous access to this chart

**Note:** This feature is not included in *Starter* accounts.

## 2.4 User Preferences



User profile, accessible from the User Menu is where all your personal preferences are configured.

User profile

General

Security

Notifications

Groups

Email

john@company.com

Change

First name

John

Last name

Citizen

Phone number

eg. +61400000001

?

Timezone

(GMT+10:00) Eastern Time - Melbourne, Sydney

?

☐ Ignore Daylight Savings Time (fixed UTC offset)

Time format

YYYY-MM-DD HH:mm:ss

?

example: 2019-01-24 13:06:39

App alerts

Show on-screen notifications

?

☒ Play sound for notifications

Announcements

☒ Send me the occasional email regarding product updates and news.

Save

Cancel

### 2.4.1 Change login email

Use the *Change* button to update your user profile email address. An email will be sent to the new address with a one-time verification link to confirm the change. Any re-sends or new address changes will invalidate previously generated links.

Users with *Single Sign-On* (page 274) configured for their domain can not change their email address.

### 2.4.2 Phone number

A valid phone number is required to receive SMS notifications. Please enter a fully qualified international phone number starting with '+'.

The account administrator(s) can optionally select to *Send SMS via email* from the *notifications* (page 276) section in account settings. When enabled, the phone number field indicates if an SMS email address is in-use for your user profile (after sending an initial SMS) and allows you to reset/retry if you change service providers.

### 2.4.3 Choose a Timezone and Time format

Select the *timezone* (page 581) you would like to use for displaying Workspace data. All data and notifications are displayed in your timezone regardless of the data's origin.

Time format is used to specify the default appearance of date & time information. Select a preset format from the drop down list or specify a custom format using *time format tokens* (page 583).

### 2.4.4 Security

The User profile has a *Security* section for immediate password changes and management of two-factor authentication. If you have forgotten your password you can use the *Forgot Password* link on the public site to send a reset link to your user profile email address.

---

**Note:** Users with *Single Sign-On* (page 274) configured for their domain will not have access to the security tab.

---

### Two-Factor Authentication

Two-Factor authentication adds an extra layer of security to your account. This feature is optional, and can be enabled by clicking the *Enable Two-Factor Authentication* button:

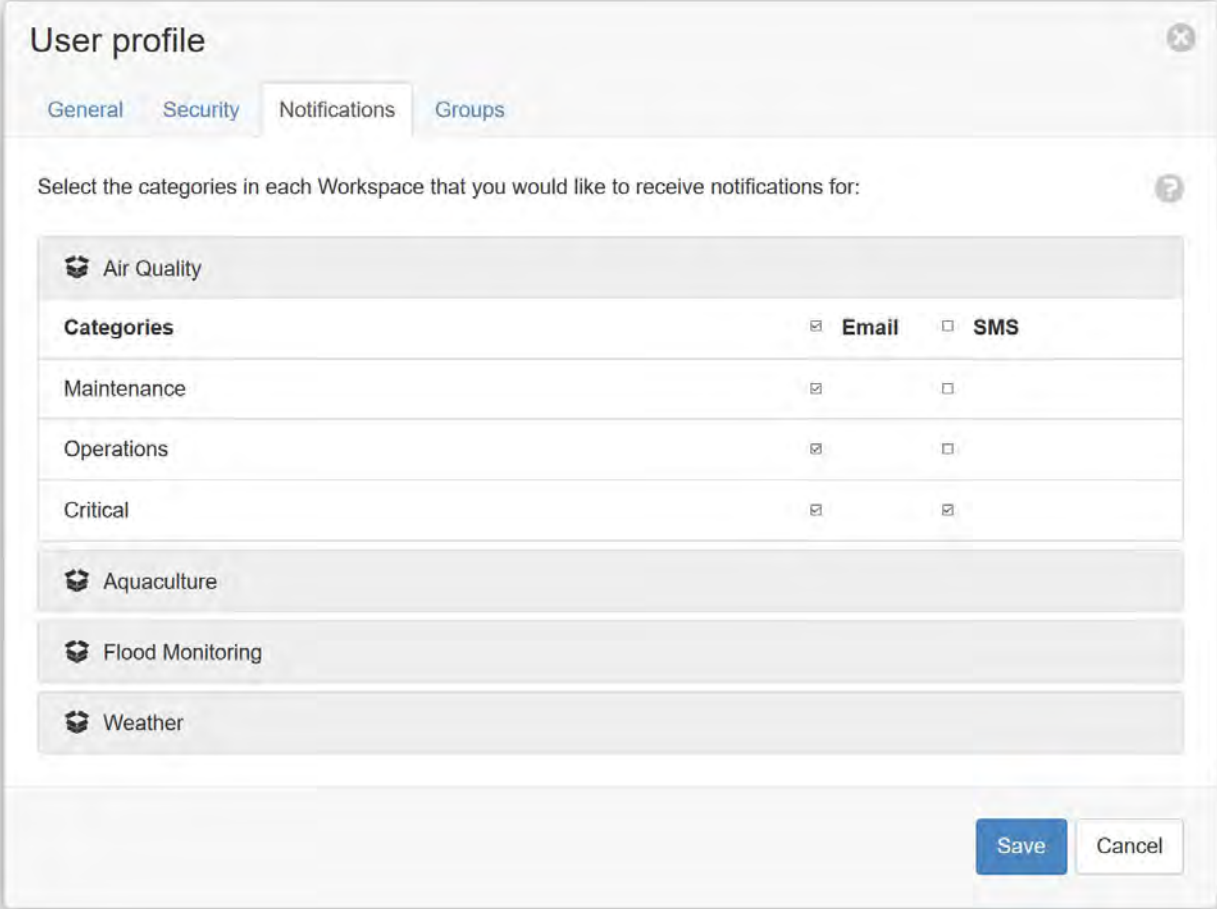
The screenshot shows a 'User profile' dialog box with a close button (X) in the top right corner. It has four tabs: 'General', 'Security', 'Notifications', and 'Groups'. The 'Security' tab is selected. Inside the 'Security' tab, there are three password input fields, each with a lock icon on the left. The first field is labeled 'Current password' and contains the text 'password'. The second field is labeled 'New password' and also contains 'password'. The third field is labeled 'retype password' and contains 'retype password'. Below these fields is a section titled 'Two-Factor Authentication'. It contains an information icon (i) followed by the text: 'Two-Factor Authentication adds an extra layer of security to your account. When logging in, you will need to provide a code along with your username and password.' At the bottom of this section is a button labeled 'Enable Two-Factor Authentication'. At the bottom right of the dialog box are two buttons: 'Save' (in blue) and 'Cancel' (in white).

Once enabled, you will need to provide a code along with your username and password when logging in.

### 2.4.5 Notification preferences

The Notifications section contains a list of Workspaces that you have been given *subscribe notifications* permissions for. Click or touch the Workspace labels to display your preferences.





**User profile**

General Security **Notifications** Groups

Select the categories in each Workspace that you would like to receive notifications for:

Workspace	Categories	Email	SMS
Air Quality	Maintenance	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Operations	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Critical	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Aquaculture			
Flood Monitoring			
Weather			

Save Cancel

Within each Workspace, you will see a list of Notification Categories specific to the Workspace. Use the SMS and Email checkboxes to set your preferences on what types of notifications you would like to receive for each Category.

**Note:** You must also be subscribed to the individual Nodes you would like to receive the notifications for. See [Subscribe to Nodes](#) (page 179) for full details. A valid phone number must also be provided in order to receive SMS notifications.

### 2.4.6 Groups

The *Groups* section displays a list of user groups your user profile is associated with. You can removed yourself from a group but you will no longer have access to Workspaces and notifications associated with the group. Only account administrators can [add users to groups](#) (page 269).

**User profile**

General Security Notifications **Groups**

*i* The list below shows the user groups you are associated with. If you remove yourself from a group you will no longer have access to Workspaces and notifications associated with the group. [Read more](#)

Account	Group
ABC Monitoring International	Engineers
Weather Group LTD	Early warning users

Save Cancel

## 2.5 Alarms



An alarm is an indication of an abnormal condition or problem detected by the system. Alarms are always associated with Nodes and are visually indicated throughout the interface using specific color coding. When an alarm is raised, you should typically acknowledge it so that other users do not attempt to investigate the same alarm condition.

### 2.5.1 Types of Alarms

There are different types of alarms which apply to specific Nodes. [Notifications](#) (page 179) can optionally be configured for Alarms and require a Category to be selected. The notification drop down is used to specify when notifications should be sent - when the alarm is raised and/or cleared. Learn more about [configuring nodes](#) (page 49).

### Communications Alarm

Communications Alarms are specific to Sources and are triggered (if enabled) when communications to the Source has failed. If the communications attempt is not successful during a user acquisition or scheduled collection it will start retrying the communications at the frequency and duration set in the Source *Retry attempts* and *Retry delay* connection properties. Once failed, communications will not be attempted until the next user acquisition or scheduled *Collection interval*.

☒ **Communications Alarm** 

Category

Maintenance 



Notifications

Failed or Restored 



### Configuration Alarm

Configuration Alarms are specific to data logger Data Sources and are triggered (if enabled) when an error is encountered when applying the user-selected [configuration](#) (page 185) file(s).

☒ **Configuration Alarm** 

Category

None 



Notifications

Never 



### Control Alarm

Control Alarms are specific to Control Parameters and are triggered (if enabled) when a control operation has failed for any reason other than a communications failure.

☒ **Control Alarm** 

Category

Maintenance 




Notifications

Failed or Recovered 





## Data Alarm

Data Alarms are specific to Sources and are triggered (if enabled) when an issue is detected with the historic data being ingested. Optionally specify an *Exclude before* and *Exclude after* time to reject (and raise an alarm) when records outside this range are detected.


☒ **Data Alarm** 


Category

Maintenance 





Notifications

Invalid 





Exclude before

None 



Exclude after

7 days 



## Outdated Alarm

Outdated Alarms are specific to Sources and are triggered (if enabled) when no new data has been received by the Source after the specified *Maximum timeout*. Outdated alarms usually indicate a problem with the device or data transmission.

☒ **Outdated Alarm**
?

<b>Category</b>	Maintenance	?
<b>Notifications</b>	Outdated or Up to date	?
<b>Active message</b>	Message when alarm becomes active. Leave blank for default	?
<b>Inactive message</b>	Message when alarm becomes inactive. Leave blank for default	?
<b>Latest data</b>	Newest parameter	?
<b>Maximum timeout</b>	1 day	?

The message can be customised and may contain *expressions* (page 523). Leave blank for a default message that includes the time since last successful collection.

## Overload Alarm

Overload Alarms are specific to Sources and are triggered when a Source has exceeded the allowable limits:

**Attachment Source** Maximum of 150 files received per *Attachment Source* (page 109) per day.

**Data Source** Acquisition and storage of up to 20000 records per *Data Source* (page 134) per day. Refer to Historic Data *Limits* (page 579) for more information.

**Overload Alarm**
?

<b>Category</b>	None	?
<b>Notifications</b>	Never	?

## Process Alarm

Process Alarms are specific to Processor Sources and Process Parameters. The alarm is triggered (if enabled) when execution of the user-defined process logic fails.

☒ **Process Alarm** 

Category

Maintenance 




Notifications

Failed or Recovered 





## Quality Alarm

Quality Alarms are specific to Parameters and Locations and are triggered (if enabled) when an acquired data point quality code matches one of the specified [quality types](#) (page 577).


☒ **Quality Alarm** 


Quality types

Bad 





Category

Maintenance 



Notifications

Always 



## State Alarm

State Alarms are defined in the [state configuration](#) (page 148) for individual Parameter Nodes. Each State can have individual alarm and [notification](#) (page 179) criteria.

**LOW BATTERY**

State will be **active** when Battery decreases to **11 VDC**. An **alarm** will be raised.

State will become **inactive** when Battery increases above **11.5 VDC**.

Notifications will be sent to **Maintenance** category when state becomes **Active**.

Hide

Name

LOW BATTERY

Threshold

11

☒ Raise alarm

Notifications

Active

Category

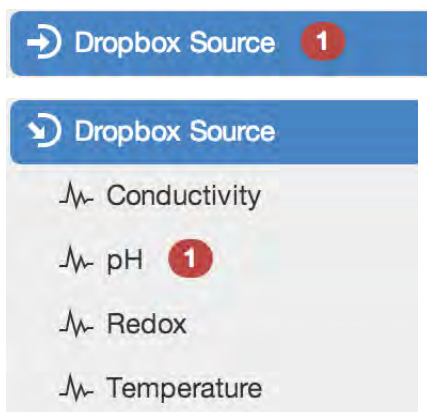
Maintenance

### 2.5.2 Alarm Indicators

Alarms are always shown in *RED*. When an alarm has been Acknowledged it will be shown in *ORANGE*. When the Alarm is Cleared (when the condition or problem has returned to normal, or when the user has manually cleared the alarm) the Node or Content is displayed with its normal color. See below for an example of alarm content.

### Workspaces Tree

Alarms are shown in the Workspaces Tree as numbers indicating the total count of unacknowledged alarms on or within the Node. Expanding the Nodes will show the individual alarm counts next to each contained Node.



## List View

List View has an Alarm Icon which can be clicked or touched to reveal the Alarm Panel. The Alarm Panel displays a quick list of the currently active or acknowledged alarms for the corresponding Node.

### Active Alarm

<input type="checkbox"/>	argo66/Demonstration/Aquarium	Dropbox Source	 pH	7.82	LOW	2014-01-09 11:57:49	  
--------------------------	-------------------------------	----------------	--------------------------------------------------------------------------------------	------	-----	---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Acknowledged Alarm

<input type="checkbox"/>	argo66/Demonstration/Aquarium	Dropbox Source	 pH	7.82	LOW	2014-01-09 12:05:49	  
--------------------------	-------------------------------	----------------	--------------------------------------------------------------------------------------	------	-----	---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Alarm Panel accessible from List View

#### Alarms

**STATE** @ 2014-01-09 11:59:26   

State changed from NORMAL to LOW

#### Alarms

**STATE** @ 2014-01-09 11:59:26 

State changed from NORMAL to LOW

*Acknowledged @ 2014-01-09 12:07:24 by nick@argos.io*

*Water quality sub-optimal. Sent maintenance team to investigate.*

## Map View

Map View applies the alarm color variations to the Location markers, temporarily overwriting the default marker color.

### Active Alarm



### Acknowledged Alarm





### 2.5.3 Alarm Operations

Alarm operations are accessible from the Node and View context menus, List View toolbar and Alarms Panel (accessible from List View). The following alarm operations are available:

#### Acknowledge



Acknowledge an alarm so other users do not attempt to investigate the same alarm condition. Requires *acknowledge alarms* permission.

#### Acknowledge with comment



Acknowledge an alarm and include a comment to indicate the cause or the action that is being taken to resolve the issue. The acknowledge alarms dialog will be displayed and you will be prompted to enter a comment for the acknowledgement. Requires *acknowledge alarms* permission.

#### Clear alarm



Clear alarms to reset any active or acknowledged alarms on the Node. Alarms are automatically cleared by the system when the condition or problem has returned to normal. You should only clear alarms if you have changed the alarm configuration for the Node after the alarm has occurred, otherwise there will

no longer be any visual indication that the Node is in an abnormal condition until the alarm is raised again. Requires *edit states and alarms* permission.

## 2.5.4 Alarm History



Use the [Events View](#) (page 43) with the Alarm Filter enabled to review all Alarm events that have occurred for the Selected Node. Events are logged when an Alarm is raised (becomes active), Acknowledged and Cleared. All information including time, comments and user are logged with the event.

Type	Time	Source	Message	User
↕	2014-01-09 12:10:23	Jw- argo66/Demonstration/Aquarium/Dropbox Source/pH	ⓘ State changed from LOW to NORMAL	System
⬇	2014-01-09 12:07:24	Jw- argo66/Demonstration/Aquarium/Dropbox Source/pH	ⓘ Acknowledged alarm 'STATE' with comment 'Water quality sub-optimal. Sent maintenance team to investigate.'	nick@argos.io
↕	2014-01-09 11:59:37	Jw- argo66/Demonstration/Aquarium/Dropbox Source/pH	ⓘ State changed from NORMAL to LOW	System

## 2.6 Notifications



Notifications are used to inform users and groups of an [alarm](#) (page 170) or other condition via Email and SMS. Users can personally subscribe to specific Nodes and specify their [notification preferences](#) (page 168) or you can [manage notifications](#) (page 180) on behalf of a user or group if you have [manage notifications](#) security permission.

There are 3 steps to configuring notifications:

1. [Configure the Node](#) (page 49) and set the Notification option and Category for an Alarm type or State.
2. Configure your user profile [notification preferences](#) (page 168).
3. [Subscribe to the Nodes](#) (page 179) you would like to receive notifications for. Alternatively use the [manage notifications](#) (page 180) dialog to subscribe other users/groups.

Account administrators can also define custom [notification limits and categories](#) (page 276).

### 2.6.1 Subscribe to Nodes

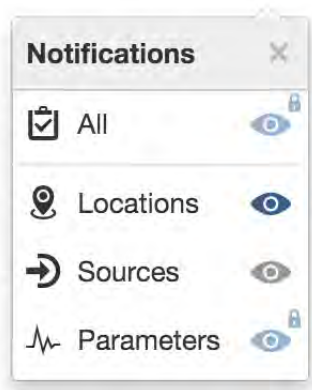


Subscribe to and Unsubscribe from Nodes via the Workspace Menu and View's context menus, List View toolbar or from the subscribe icon shown in the List View rows.

When changing subscription options from the toolbar or context menu the checked or chosen Nodes including all contained Nodes will have the new subscription options applied.

Click or touch the subscribe icon in the List View to toggle your subscription to the corresponding Node(s). If the Node contains other Nodes (ie. Location Node contains Source Node), the Notification Subscription Panel will be displayed to allow you to select which types of Nodes you would like to subscribe to or unsubscribe from. A partially colored subscription icon indicates that not all child Nodes have been subscribed to.

*Notification Subscription Panel accessible from List View*



You may notice a *Lock* symbol next to the subscribe icon(s) if you do not have [subscribe notifications](#) permission or are included in a [group](#) (page 269) that has been subscribed to receive notifications. These locked notifications can only be unsubscribed via the [manage notifications](#) (page 180) dialog. However you

can manage your personal *notification preferences* (page 168) or *remove yourself from a group* (page 169) via the user preferences dialog.

---

**Note:** Users require *subscribe notifications* permission to subscribe/unsubscribe to Nodes themselves. Refer to security *roles* (page 270) for more details.


---

## 2.6.2 Manage notifications


You can manage notification subscriptions on behalf of a user or group via the *manage notifications* dialog, accessible from the Workspace context menu. Select the user or group from the dropdown menu and *subscribe* (page 179) them to nodes. You can also also set *notification categories* (page 168) for individual users for the currently selected Workspace.


The dropdown menu only includes users and groups that have access to the Workspace.


Manage notifications



**User/Group**


John Citizen (john@company.com) ▾



**Water Quality**



Location



Location (2)


Text File Source



Conductivity


pH


Redox


Temperature

Notification categories	<input checked="" type="checkbox"/> Email	<input type="checkbox"/> SMS	<input type="checkbox"/> Voice
Maintenance	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Operations	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Critical	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>


Get help with notification management

Close

**Note:** Users require *manage notifications* permission to subscribe other users and groups to Nodes. Refer to security [roles](#) (page 270) for more details.

## 2.7 Operate and Control



The Operate menu, available from the Node and Content context menus or List View toolbar provides access to the Acquire, Control and Manage configuration operations.

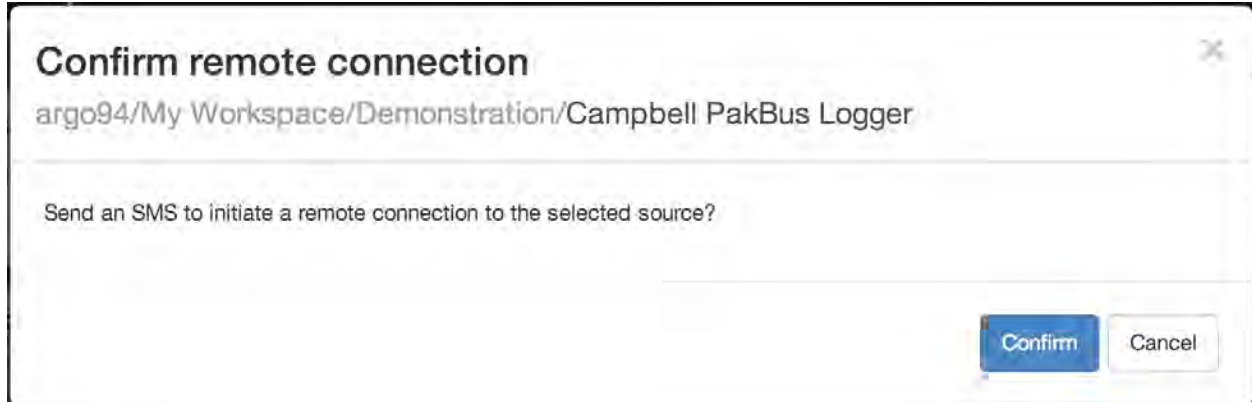
### 2.7.1 Acquire



Acquire performs a one-time collection on the Source(s) chosen. View the current operation status from the Source List View. Requires *operate* permission.

When acquiring from a Source that uses a *Scout modem* you may be prompted to confirm the remote connection. This occurs when the Source Schedule is *Disabled* or set to *Remote* (ie. device not permanently connected). An SMS will be sent to the *Scout modem* to initiate a callback. The account owner will be charged for the SMS at the standard SMS Notification rate applicable to their account type. Read more about [Scout configuration](#) (page 384).

*Scout modem remote connection confirmation*



### 2.7.2 Control



Control allows you to update or set values in a remote data logger. Control operations can only be performed on Control Parameters and requires *operate* permission. Read more about [Node types](#) (page 49).

The control operation will be attempted immediately upon applying or setting the control value. If a connection to the Source can not be established, the control value will be applied on the next successful connection. Any pending control values are indicated with an arrow.

Control Parameters store the history of the control values set by the user, not the value of the parameter in the data logger.

## Control dialog

The Control dialog can be opened from the Operate menu. It displays a list of Control Parameters within the chosen Node. Set the control values and *Apply*. Cancel a pending control operation by clicking the 'x' next to the parameter control value. The current status of the operation is shown for each parameter being updated and the rows will turn green to indicate success or red to indicate failure.

The Control dialog window displays a table with the following columns: Location, Path, Parameter, Control, Value, State, and Status. The table contains four rows of control parameters. The first three rows have their 'Control' values set to 1, TRUE (1), and FALSE (0) respectively. The fourth row has its 'Control' value set to 2.37, and its 'Status' is PENDING. The 'Apply' button is located at the bottom right of the dialog.

Location	Path	Parameter	Control	Value	State	Status
argo66/Demonstration/Flow Monitoring	CR200 Scout	Public - SDI12 Purge	1			
argo66/Demonstration/Flow Monitoring	CR200 Scout	Public - Svc Mode	TRUE (1)			
argo66/Demonstration/Flow Monitoring	CR200 Scout	Public - Transmit ALERT all	FALSE (0)			
argo66/Demonstration/Flow Monitoring	CR200 Scout	Public - WL Offset	2.37	→ 2.37		PENDING

Buttons: Apply, Close

## List View inline Control

Control parameters can have their control value set directly from the Parameters List View. Click or touch the control icon next to the value to update or set the control value. Click or touch the control icon again to apply the new value or cancel the control operation by clicking the 'x'.

Location	Source	Parameter	Value	State	Latest Data			
argo66/Demonstration/Flow Monitoring	CR200 Scout	Public - Transmit ALERT all	TRUE (1)		--			
argo66/Demonstration/Flow Monitoring	CR200 Scout	Public - WL Offset	→ 2.37		--			

---

**Note:** Control Parameters can be configured to restrict the user input for the control value. See [Configuring Parameters](#) (page 147) for full details.

---



### 2.7.3 Manage configuration



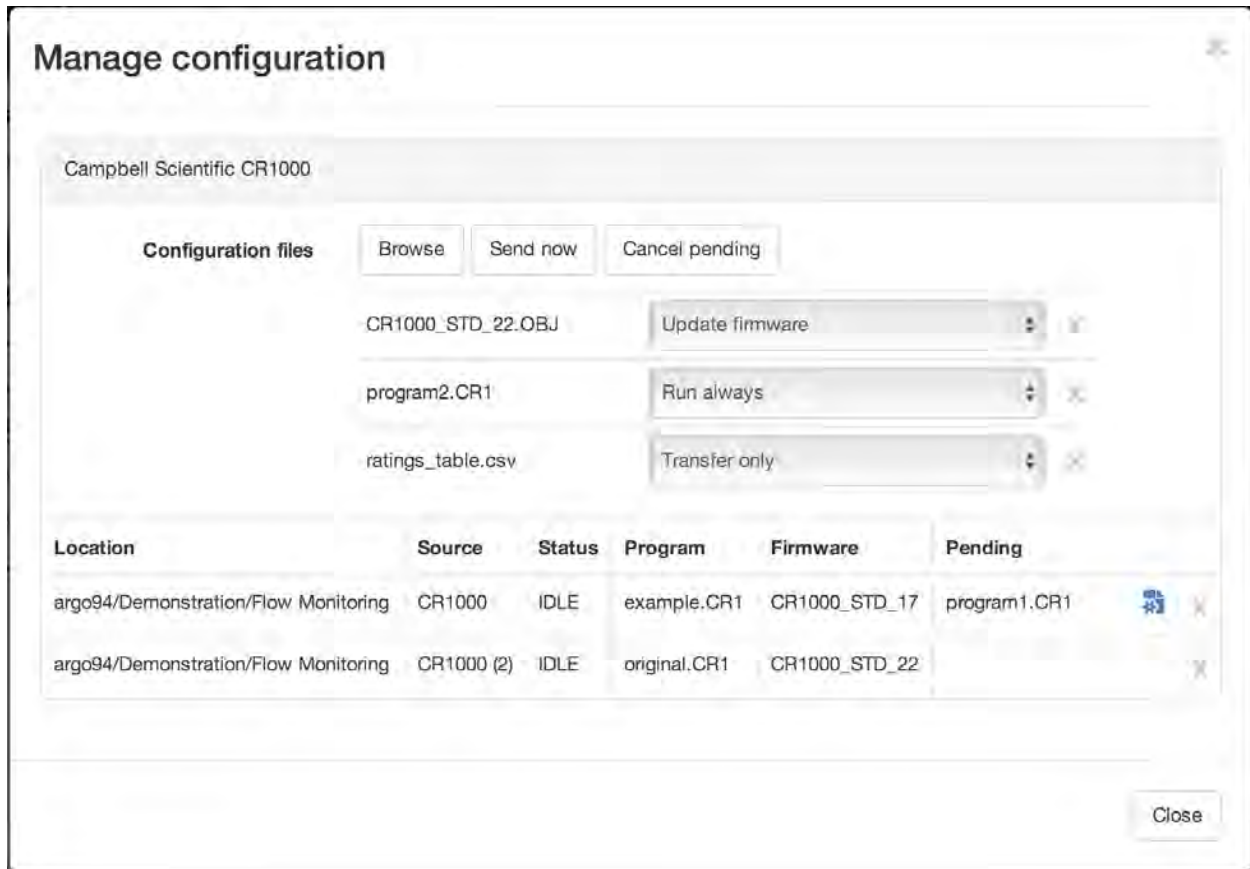
Manage the configuration for a single Data Source (data logger) or perform network-wide configuration updates. Configuration support varies between data loggers, including the ability to send programs, firmware and other files to the remote logger(s).

Manage configuration operations can only be performed on Data Sources (data logger) and requires *configure* permission. Read more about [Node types](#) (page 49).

#### Manage configuration dialog

The Manage configuration dialog can be opened from the Operate menu. It displays a list of Data Sources within the chosen Node that support managed configuration. Data Sources are grouped together based on data logger type/model. If you do not wish to apply the configuration to a Data Source already displayed in the dialog then you can remove the source by clicking its close ‘x’ button.

The current status of the operation is shown for each Data Source being configured and the associated list item will turn green to indicate success or red to indicate failure. If the configuration update fails, check the *Configuration alarm* message for the corresponding Source.



Use the *Browse* button to select file(s) to upload. After the files have been selected you must assign a *configuration option* (page 187) to each file.

Click *Send now* to upload the file(s) and queue the configuration for each Data Source displayed in the immediate list.

Use the *Cancel pending* button to remove all pending configuration for each Data Source displayed in the immediate list.

New configuration will be applied immediately where possible, otherwise it will be applied during the next communication with the Data Source.

---

**Note:** Sending new configuration files will automatically remove any pending configuration from the Data Source.

---

### List View pending configuration

Data Sources with pending configuration will have a configuration icon displayed in the *status* column of the Sources list. You can quickly review any pending configuration files including the assigned *configuration option* and time of upload.

Use the *Send now* button to immediately trigger or retry a configuration update or *Cancel pending* will remove the pending configuration for an individual Source.



## Configuration options

Configuration options vary based on data logger type/model. See below for a list of options available for each supported logger type.

### Campbell PakBus CR200

**Run always** Send a pre-compiled (.bin) CR200 program to the logger and run the program immediately and on startup.

### Campbell PakBus CR300/CR800/CR1000/CR3000/CR6

**Run always** Send a CR300 (.cr300), CR800 (.cr8), CR1000 (.cr1), CR3000 (.cr3) or CR6 (.cr6) program (or pre-compiled .dld file) to the logger and run the program immediately and on startup.

**Run always (preserve data)** Send a program to the logger and run the program immediately and on startup. Data tables will not be reset if possible.

**Run once** Send a program to the logger and run the program immediately but not on startup.

**Update firmware** Send firmware/operating system (.obj) to the logger.

**Transfer only** Send raw files to the logger.

## 2.8 Historic Data



Historic data is stored in a raw unprocessed format with an optional *quality code* (page 577) and *annotation* (page 578). Display, edit and export data in variety of *tabular* (page 190) and *chart* (page 66) formats.

Apply *historic aggregates* (page 569) to process the raw data and apply specific calculations or use *processing and logic* (page 223) for advanced statistical analysis and decision making.

Use the eagle.io *HTTP API* (page 433) to automate importing or extracting data.

*Data editing* (page 188) can be performed in both *historic table* (page 195) and *historic chart* (page 205) formats.

Eagle.io supports the acquisition and storage of up to **20000 records per Data Source** (page 134) **per day**. Exceeding the limit will trigger an *Overload Alarm* (page 173) on the Source. Refer to Historic Data *Limits* (page 579) for more information.

### 2.8.1 Editing



Data editing allows for the modification of your historic data directly from the user interface in *historic table* (page 195) and *historic chart* (page 205) formats.

Historic data is always edited as raw unprocessed records without any *aggregate* (page 569) or *transform* (page 147) options applied. Records can be modified individually or as a selection including:



**Value** Record values can be adjusted to remove spikes or apply offsets.



**Quality** *Quality codes* (page 577) can be assigned to validate data or *filter* (page 202) records from display.



**Annotation** *Annotations* (page 578) can be added to provide context to specific events.

Users must have *edit data* or *edit quality & annotations permission* (page 270) for the parameters they are trying to edit. Data editing is not available on STARTER accounts. Record *limits* (page 580) apply when editing data in chart format.

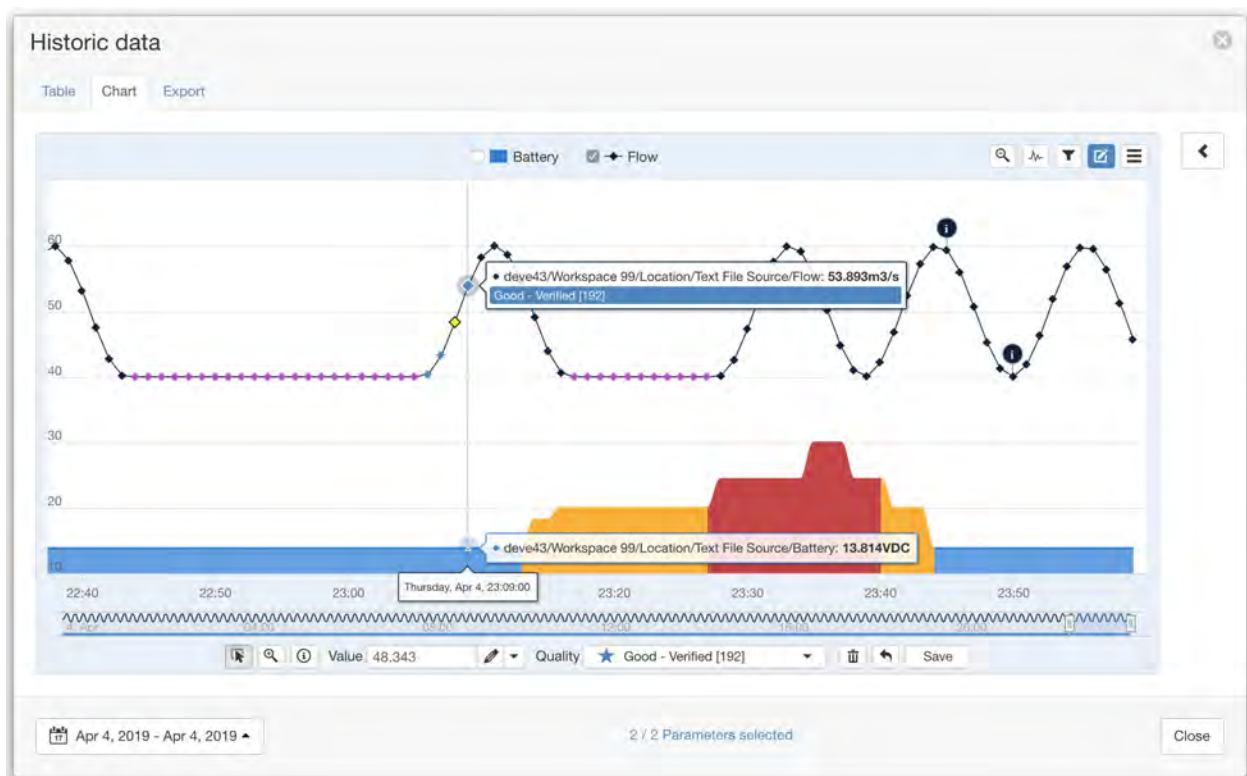
---

**Note:** All saved changes are permanent. You should restrict access with the *edit data* user *permission* (page 270) and audit changes via the *events* (page 43) view.

---

**Note:** *Process Parameters* (page 223) cannot be edited, as the data is generated dynamically whenever the process code is changed.

Refer to the *historic chart* (page 205) documentation for data editing in chart format.



Refer to the *historic table* (page 195) documentation for data editing in tabular format.

Historic data

Table Chart Export

Time	Battery (VDC)	Flow (m3/s)
2019-04-30 10:47:00	13.756	59.564
2019-04-30 10:46:00	13.756	59.638
2019-04-30 10:45:00	13.74	56.702
2019-04-30 10:44:00	13.751	51.673
2019-04-30 10:43:00	13.751	46.121
2019-04-30 10:42:00	13.751	41.781
2019-04-30 10:41:00	13.712	40.008
2019-04-30 10:40:00	13.751	41.355
2019-04-30 10:39:00	13.751	45.402
2019-04-30 10:38:00	0	50.885
2019-04-30 10:37:00	0	56.091

Value 41.781 Quality ★ Good - Verified [192] Save

Jump to date 2 / 2 Parameters selected Close

## 2.8.2 Table



Historic table is accessible from the List View toolbar and Node context menus. The most recent data is always displayed first. Scroll down to automatically load older data or use the *Jump to date* control to quickly move to a specific date.



The *timestamps* are always displayed in your preferred timezone and time format. See configuring *user preferences* (page 166) for full details.


If you have assigned *quality codes* (page 577) to the data the associated quality color is displayed in the table cell(s). Move your mouse over a table cell to display its corresponding quality code and description in the toolbar.

Historic data

Table Chart Export

Time	J <sub>W</sub> Battery (VDC)	J <sub>W</sub> Flow (m3/mg)
2019-04-30 10:47:00	13.756	59.564
2019-04-30 10:46:00	13.756	59.638
2019-04-30 10:45:00	13.74	56.702
2019-04-30 10:44:00	13.751	51.673
2019-04-30 10:43:00	13.751	46.121
2019-04-30 10:42:00	13.751	41.781
2019-04-30 10:41:00	13.712	40.008
2019-04-30 10:40:00	13.751	41.355
2019-04-30 10:39:00	13.751	45.402
2019-04-30 10:38:00	0	50.885
2019-04-30 10:37:00	0	56.091



★ Good - Verified [192]

 Jump to date
 2 / 2 Parameters selected
Close

**Note:** Historic data for all parameters being displayed is automatically interleaved, therefore you may notice empty cells where the parameter(s) do not have a value for the corresponding row timestamp.

## Toolbar



**Filters** Filter the dataset by selecting *quality codes* (page 577) associated with the data points that you want to include. While editing data, *apply* filters to automatically select associated data points and temporarily disable filtered data point cells to prevent modification.

**Filter / select quality**

☒ **Select all**

☒ **Uncertain**

☒ ★ Unknown

☒ ★ Uncertain - Subnormal [149]

☒ **Bad**

☒ ★ Bad - No Data [155]

☒ ★ Bad - No Bounds [156]

☒ ★ Bad - Data Unavailable [158]

☒ **Good**

☒ ☆ Unset

☒ ★ Good - No Data [165]

☒ ★ Good - Verified [192]

Apply

Reset





**Edit data** Use this feature to toggle into [data editing](#) (page 195) mode and display the data editing toolbar.



**Note:** Users must have *edit data* or *edit quality & annotations permission* (page 270) for the parameters they are trying to edit. Data editing is not available on STARTER accounts.

## Parameter selection

The historic data dialog will only display the first 15 selected parameters by default. Use the *Parameter selection* panel accessible via the arrow on the right to refine your selection and change how the data should be displayed, including any aggregate calculations to be applied.

**Parameter selection** Auto Custom

☒ ce/Sample Location/Text File Source/pH RAW ▲

**Display type** Value State ?

**Aggregate** Average ?

**Interval** 1H ?

**Base time** D ?

Apply Apply to all

☒ Sample Location/Text File Source/Redox RAW ▼

☒ Sample Location/Text File Source/Temperature RAW ▼

☒ Sample Location/Text File Source/Conductivity RAW ▼

**Display type** *Value* displays the parameter values.

*State* displays the state name associated with each value.

**Aggregate** Aggregates provide a method to resample raw data over a specified period. See the [list of available aggregates](#) (page 569) for full details. Select RAW to display the original raw data with no aggregation applied (Interval and Base time options will be hidden).

**Interval** Specify the data range that is included for each calculated value. Example *30M* will calculate a value for each 30 minute window of data.

**Base time** Base time is used to determine the start of the aggregate calculations. Example *D+9H* will start the calculations at 9am (start of current day + 9 hours). See [working with relative time](#) (page 585) for full details.

## Data editing



The *edit data* (page 192) button toggles into data editing mode.

Users must have *edit data* or *edit quality & annotations permission* (page 270) for the parameters they are trying to edit. Data editing is not available on STARTER accounts. Table cells or records with a light grey background can not be selected or modified.

Historic data is always edited as raw unprocessed records without any *aggregate* (page 569) or *transform* (page 147) options applied. Data editing can also be performed in *historic charts* (page 205).

---

**Note:** All saved changes are permanent. You should restrict access with the *edit data* user *permission* (page 270) and audit changes via the *events* (page 43) view. *Limits* (page 580) apply when editing historic data.

---

## Selecting records



Click a table cell to select an individual record or click a timestamp to select an entire row.

Selected cells are displayed with a yellow background when the user has both *edit data* and *edit quality & annotations* permissions for the parameter. Alternatively a dark grey background will be displayed when the user only has *edit quality & annotations* permission.

While selecting records, hold the **shift key to add** or **alt key to remove** from the current selection.

Historic data

Table Chart Export

Time	J <sub>1</sub> - Battery (VDC)	J <sub>1</sub> - Flow (m3/s)
2019-04-30 10:47:00	13.756	59.564
2019-04-30 10:46:00	13.756	59.638
2019-04-30 10:45:00	13.74	56.702
2019-04-30 10:44:00	13.751	51.673
2019-04-30 10:43:00	13.751	46.121
2019-04-30 10:42:00	13.751	41.781
2019-04-30 10:41:00	13.712	40.008
2019-04-30 10:40:00	13.751	41.355
2019-04-30 10:39:00	13.751	45.402
2019-04-30 10:38:00	0	50.885
2019-04-30 10:37:00	0	56.091

Value 41.781 Quality ★ Good - Verified [192] Save

Jump to date 2 / 2 Parameters selected Close

Click and drag over the table to quickly select multiple records.

Historic data

Table Chart Export

Time	J <sub>W</sub> Battery (VDC)	J <sub>W</sub> Flow (m <sup>3</sup> /s)
2019-04-30 10:47:00	13.756	59.564
2019-04-30 10:46:00	13.756	59.638
2019-04-30 10:45:00	13.74	56.702
2019-04-30 10:44:00	13.751	51.673
2019-04-30 10:43:00	13.751	46.121
2019-04-30 10:42:00	13.751	41.781
2019-04-30 10:41:00	13.712	40.008
2019-04-30 10:40:00	13.751	41.355
2019-04-30 10:39:00	13.752	45.402
2019-04-30 10:38:00	0	50.885
2019-04-30 10:37:00	0	56.091

Value Quality Save

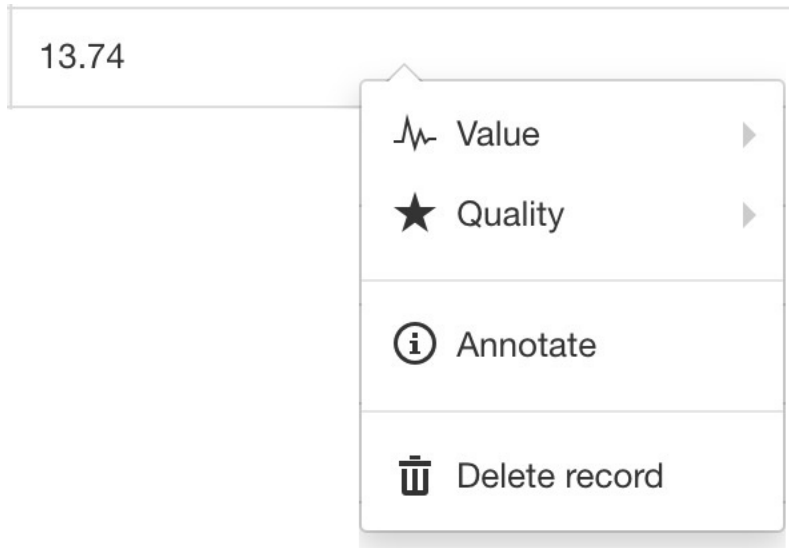
Jump to date 2 / 2 Parameters selected Close

Apply [filters](#) (page 191) to select records based on quality.

## Editing records

An individual record can include **value**, **quality** and **annotation**.

Right-click on an editable cell or selection of cells to quickly perform operations via the cell context-menu.



## Set value

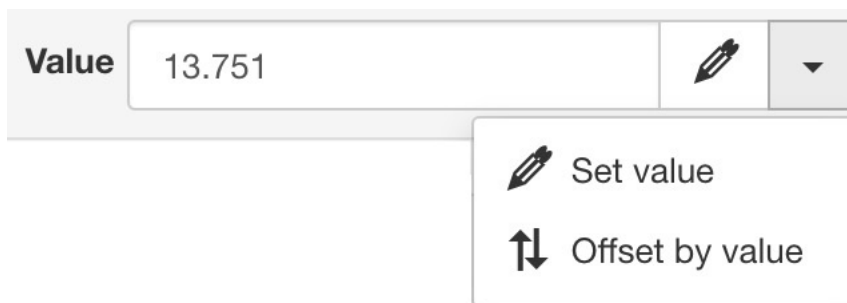


Enter a new value to be assigned to the selected cell(s) and press enter or click the *set value* icon. Change the functionality of the value input by selecting a value operation from the dropdown menu.

`Set value` assigns a specific value to selected records.

`Offset by value` adds a value to to selected records.

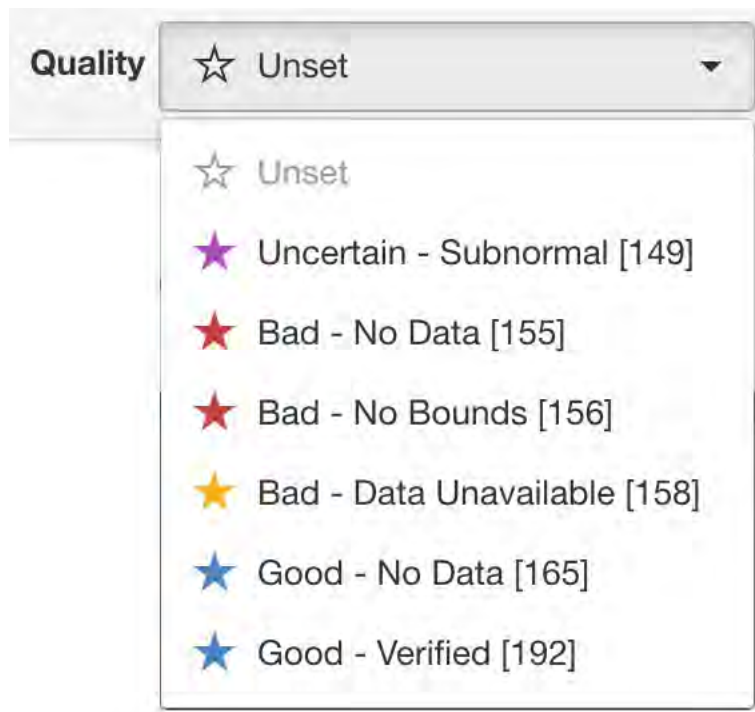
`Scale by value` multiplies selected records by a specific value.



You can also use *set value* to unset a value (leave blank) which will result in historic charts showing a gap in the series when displaying RAW historic data.

## Set quality

☆ Select a quality to be assigned to the selected cell(s) from the dropdown quality menu. You can only assign *quality codes* (page 577) that have been pre-configured in *account settings* (page 266).

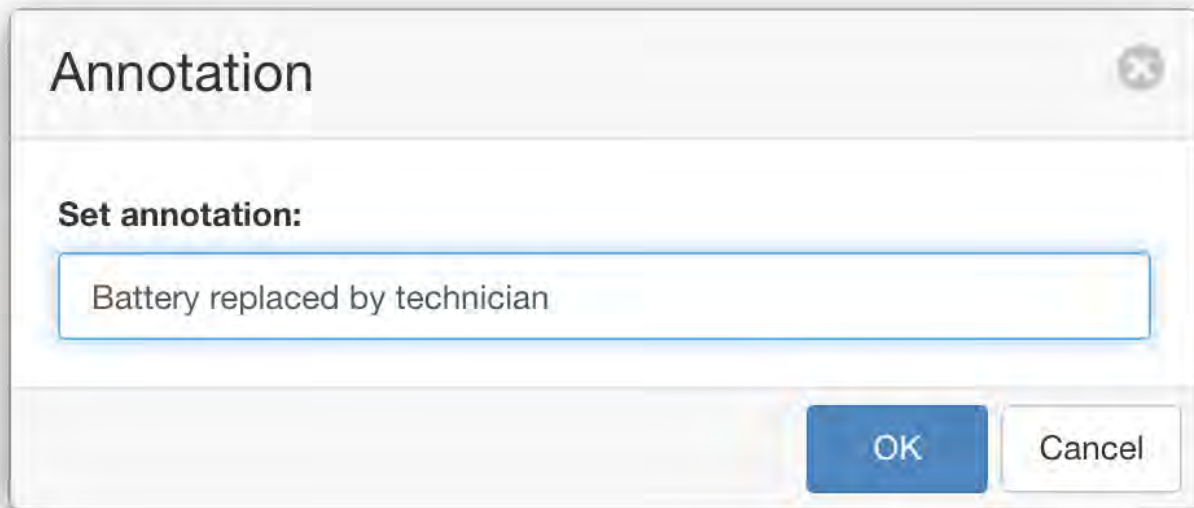


## Set annotation

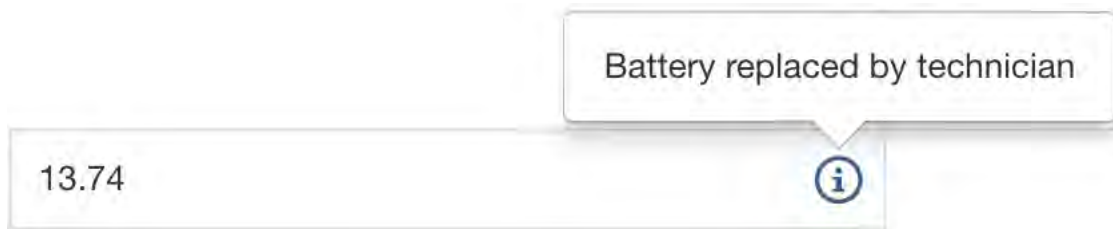


To set an *annotation* (page 578) for a single record either change to *annotate mode* from the edit toolbar and click on a cell or right-click on a cell and select *annotate* from the context menu.

Annotations can be set on multiple cells at once by first selecting the cells then selecting *annotate* from the context menu.



Hover or click the annotation icon to display the annotation text.



To remove an annotation clear the text in the annotation entry dialog or right-click on the cell annotation icon and select *remove annotation* from the context menu.





## Delete record



The delete operation permanently removes the selected records from the historic dataset.

## Undo last edit



The *undo* command reverts the last edit operation. You can continue to *undo* all edit operations since the last save.

## Save changes

Save all changes to the modified records.

---

**Note:** All saved changes are permanent. You should restrict access with the *edit data* user *permission* (page 270) and audit changes via the *events* (page 43) view.

---

## 2.8.3 Chart



Historic chart is accessible from the List View toolbar and Node context menus. The Historic Chart provides the same functionality as the charts displayed in *Chart View* (page 35) but can be displayed directly from the Workspaces Tree, Map or List View without navigating away from your current view.

If you have assigned *quality codes* (page 577) to the data, the associated quality code and color is displayed on hover.

Refer to *working with charts* (page 37) for full details on chart interaction and functionality.



## Toolbar



**Range selector** Use the range selector to adjust the date range being shown on the chart.



**Zoom out** After zooming in on a section of the chart the *zoom out* icon is displayed to reset the zoom range.



**Toggle parameter visibility** Click the parameter icon to quickly toggle the visibility of all parameters. Click parameters in the chart legend to toggle the visibility of individual parameters.



**Toggle parameter editing** After toggling *data editing* (page 205) mode, the toggle parameter editing button is shown. Click this button to quickly select/deselect all editable parameters for editing.



**Filters** Filter the dataset by selecting *quality codes* (page 577) associated with the data points that you want to include. While editing data, *apply* filters to automatically select associated data points and temporarily hide filtered data points to prevent modification.

Filter / select quality

☒ Select all

☒ Uncertain

☒ ★ Unknown

☒ ★ Uncertain - Subnormal [149]

☒ Bad

☒ ★ Bad - No Data [155]

☒ ★ Bad - No Bounds [156]

☒ ★ Bad - Data Unavailable [158]

☒ Good

☒ ☆ Unset

☒ ★ Good - No Data [165]

☒ ★ Good - Verified [192]

Apply
Reset



**Edit data** Use this feature to toggle into [data editing](#) (page 205) mode and display the data editing toolbar.

Value
13.74

Quality
★ Good - Verified [192]


Save

**Note:** Users must have *edit data* or *edit quality & annotations permission* (page 270) for the parameters they are trying to edit. Data editing is not available on STARTER accounts.



**Menu** Export options can be found in the dropdown menu including:

Print the chart.



Save the chart as an image.

## Parameter selection

The historic data dialog will only display the first 15 selected parameters by default. Use the *Parameter selection* panel accessible via the arrow on the right to refine your selection and change how the data should be displayed, including any aggregate calculations to be applied.

**Parameter selection** Auto Custom

☒ ce/Sample Location/Text File Source/pH RAW ▲

**Display type** Value State ?

**Aggregate** Average ?

**Interval** 1H ?

**Base time** D ?

Apply Apply to all

☒ Sample Location/Text File Source/Redox RAW ▼

☒ Sample Location/Text File Source/Temperature RAW ▼

☒ Sample Location/Text File Source/Conductivity RAW ▼

**Auto/Custom** *Auto* will use the default chart options for each parameter (as configured in parameter properties).

*Custom* allows you to specify custom display settings for each parameter. These settings are shared between table and exports for convenience.

**Display type** *Value* displays the parameter values.

*State* displays the state name associated with each value.

**Aggregate** Aggregates provide a method to resample raw data over a specified period. See the [list of available aggregates](#) (page 569) for full details. Select RAW to display the original raw data with no aggregation applied (Interval and Base time options will be hidden).

**Interval** Specify the data range that is included for each calculated value. Example *30M* will calculate a value for each 30 minute window of data.

**Base time** Base time is used to determine the start of the aggregate calculations. Example *D+9H* will start the calculations at 9am (start of current day + 9 hours). See [working with relative time](#) (page 585) for full details.

---

**Note:** Please be aware that displaying RAW data for a large time range may be time consuming to retrieve and process.

---

## Data editing



The [edit data](#) (page 203) button toggles into data editing mode.

Users must have *edit data* or *edit quality & annotations permission* (page 270) for the parameters they are trying to edit. Data editing is not available on STARTER accounts.

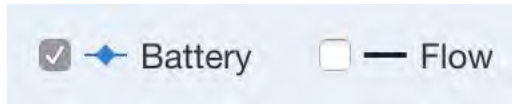
Historic data is always edited as raw unprocessed records without any [aggregate](#) (page 569) or [transform](#) (page 147) options applied. Data editing can also be performed in [tabular](#) (page 195) format.

---

**Note:** All saved changes are permanent. You should restrict access with the *edit data* user [permission](#) (page 270) and audit changes via the [events](#) (page 43) view. [Limits](#) (page 580) apply when editing historic data.

---

The chart legend automatically updates with checkboxes next to parameters than can be edited. Only the first parameter is initially checked for editing. Check additional parameters to edit concurrently or click the *Toggle parameter editing* toolbar button to quickly select/deselect all editable parameters.

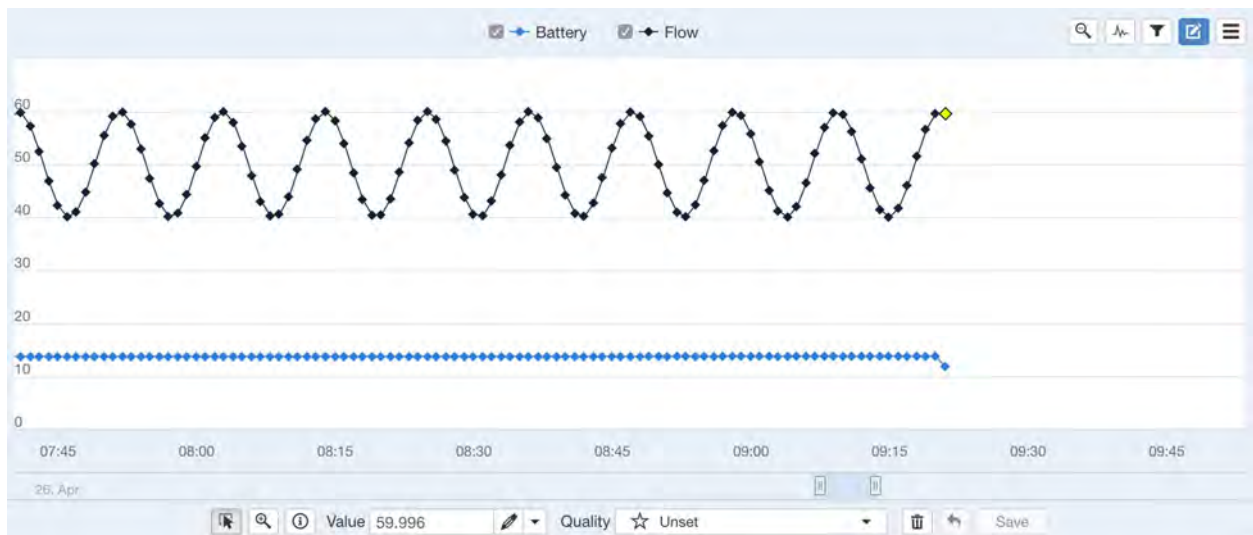


## Selecting records

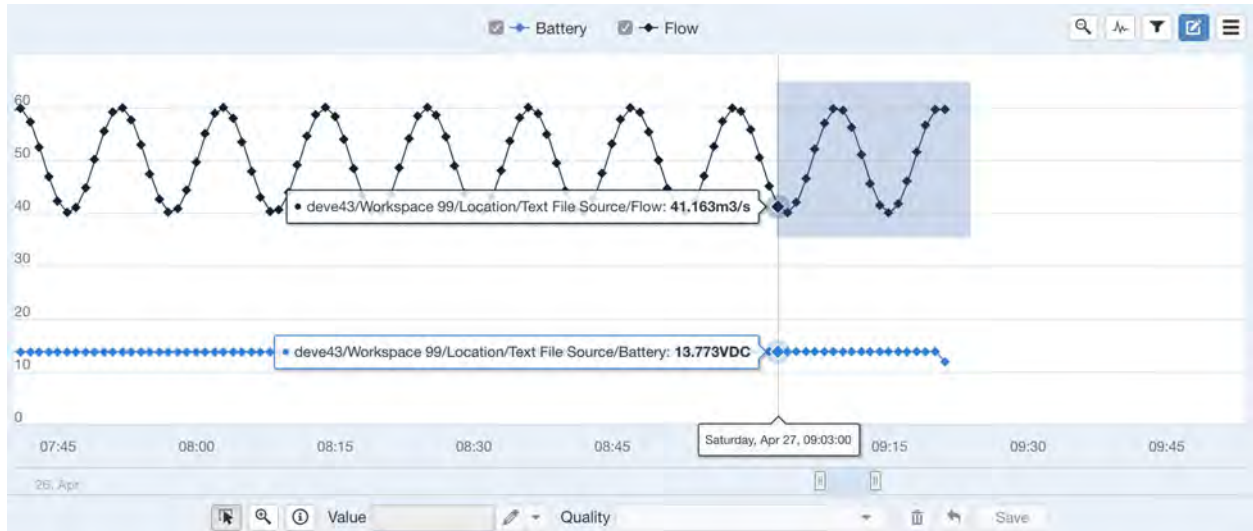
 Click a series data point marker to select an individual record.

Selected data points are displayed with a yellow marker when the user has both *edit data* and *edit quality & annotations* permissions for the parameter. Alternatively a dark grey marker will be displayed when the user only has *edit quality & annotations* permission.

While selecting records, hold the **shift key to add** or **alt key to remove** from the current selection.



Click and drag over the chart to quickly select multiple records.

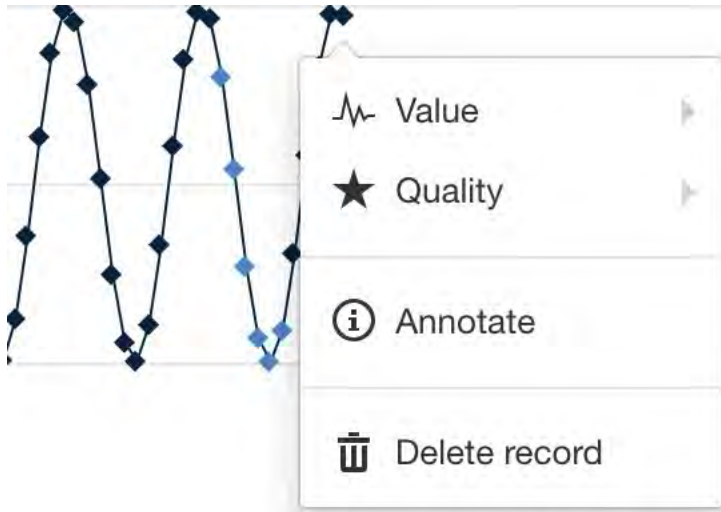


Apply *filters* (page 202) to select records based on quality.

## Editing records

An individual record can include **value**, **quality** and **annotation**.

Right-click on an editable data point or selection of data points to quickly perform operations via the chart context-menu.



## Set value



Enter a new value to be assigned to the selected data point(s) and press enter or click the *set value* icon. Change the functionality of the value input by selecting a value operation from the dropdown menu.

`Set value` assigns a specific value to selected records.

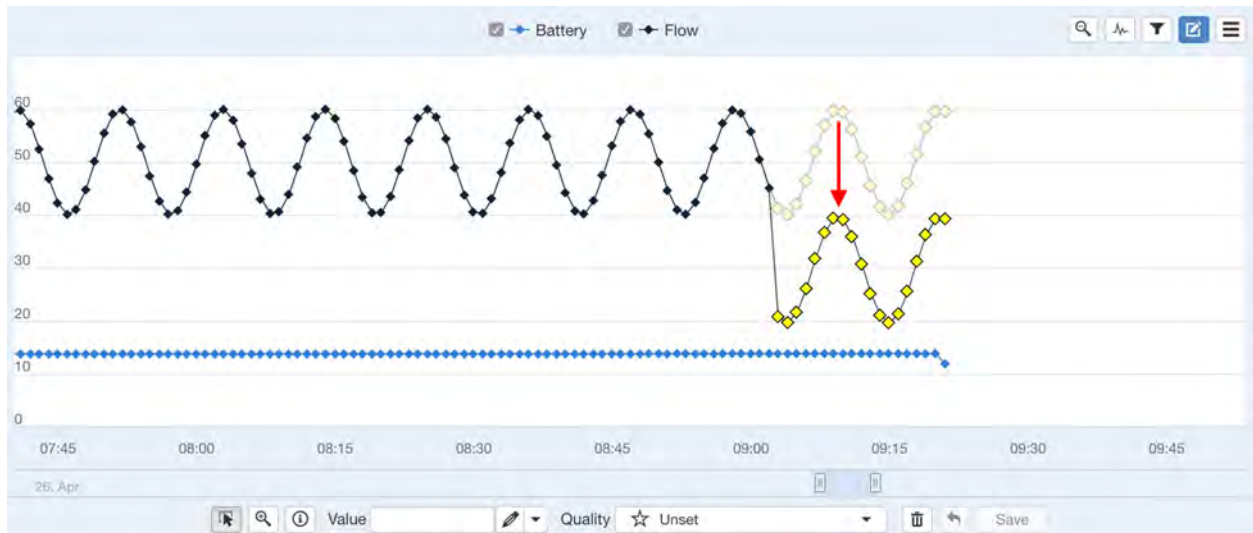
`Offset by value` adds a value to to selected records.

`Scale by value` multiplies selected records by a specific value.



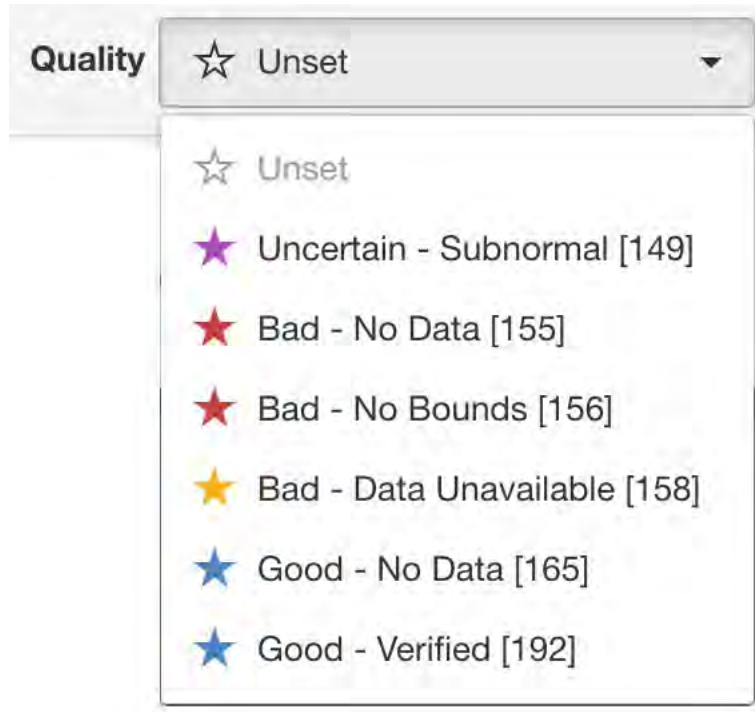
You can also use *set value* to unset a value (leave blank) which will result in historic charts showing a gap in the series when displaying RAW historic data.

Charts support *drag and drop* of selected data point(s) to free-hand edit record values.



## Set quality

☆ Select a quality to be assigned to the selected data point(s) from the dropdown quality menu. You can only assign *quality codes* (page 577) that have been pre-configured in *account settings* (page 266).

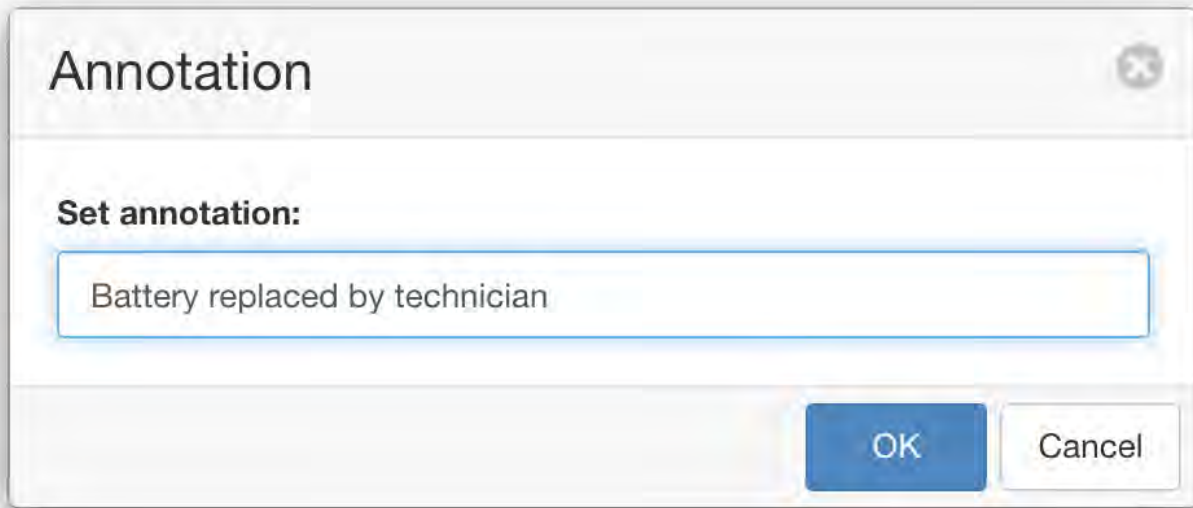


## Set annotation

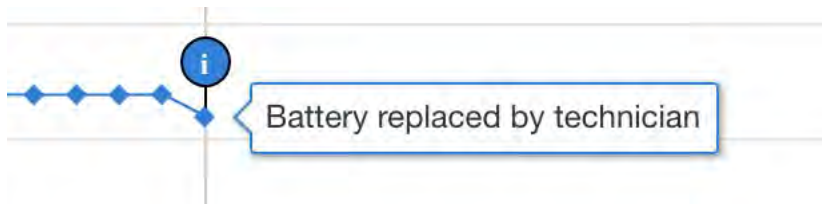


To set an [annotation](#) (page 578) for a single record either change to *annotate mode* from the edit toolbar and click on a data point or right-click on a data point and select *annotate* from the context menu.

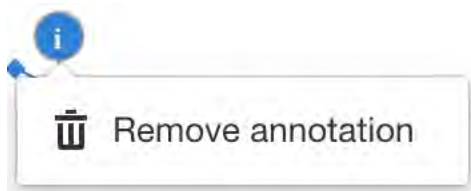
Annotations can be set on multiple data points at once by first selecting the data point then selecting *annotate* from the context menu.



Hover or click the annotation icon to display the annotation text.



To remove an annotation clear the text in the annotation entry dialog or right-click on the annotation icon and select *remove annotation* from the context menu.



## Delete record



The delete operation permanently removes the selected records from the historic dataset.

## Undo last edit



The *undo* command reverts the last edit operation. You can continue to *undo* all edit operations since the last save.

## Save changes

Save all changes to the modified records.

---

**Note:** All saved changes are permanent. You should restrict access with the *edit data* user *permission* (page 270) and audit changes via the *events* (page 43) view.

---

## 2.8.4 Export



Historic export is accessible from the List View toolbar and Node context menus. You can export all data in the system via the *Base Node* context menu or use filters and the List View for precision selection. Exporting historic data requires the *export data* permission.

The export process runs in the background so you can continue your work (including triggering additional exports) and you will be notified via *Messages* upon completion. Historic export data is always packaged into a compressed zip archive and available as a secure download. The exported files will automatically be removed from the server/messages after 12 hours.

Use the *Advanced* button to show/hide the advanced export options.

**Historic data**

Table Chart **Export**

**Format** Delimited Text

**Period** 7 Days

**ADVANCED**

**Output** File per parameter

☒ Include column headers

**Delimiter** ,

**Text qualifier** "

**Number format** Parameter format

**Time format** YYYY-MM-DD HH:mm:ss

**Timezone** (GMT+10:00) Eastern Time - Melbourne, Sydney

☒ Ignore Daylight Savings Time (fixed UTC offset)

**Quality** Do not include

**Annotations** Include as separate value

Export

**Historic Export**  
Successfully completed  
2018-07-11 14:44:43.295

Download

6 / 6 Parameters selected

Close

**Period** Select the date range for the data to be exported.

**Format** Select the export format of the file(s):

*Delimited Text*

Data is stored as rows of values separated with a specific *delimiter* character.

*JSON Time Series*

Data is stored in the *JSON Time Series* (page 572) format.

**Output** Select how the parameter data should be grouped:

*File per parameter*

Every parameter will be exported into a separate file.

*Interleaved*

All parameters will be exported into a single file with values interleaved on timestamp.

Optionally include header labels for each column.

**Delimiter** Specify the *delimiter* character to use for separating the values.

**Text qualifier** Specify a character to use for surrounding text values.

**Number format** Specify how number parameters should format values:

*Parameter format* Use the format configured in each parameter.

*Full resolution* Will output the value with all available decimal places.

*Custom* Specify a custom *format* (page 147) to be used.

**Time format** Specify the output format for timestamps. Select a preset format from the drop down list or specify a custom format using *time format tokens* (page 583).

**Timezone** Select the *timezone* (page 581) to use for the exported data.

Optionally ignore Daylight Savings Time if the data should use a fixed UTC offset.

**Quality** Specify how quality information should be exported:

*Do not include* Quality is ignored.

*Include with value* Quality code will be included in the same column as the value, separated with a Quality delimiter.

*Include as separate value* Quality code will be included as a separate column in the file.

**Quality delimiter** Specify the *delimiter* character to use for separating the value and quality code when *Include with value* is specified for the *Quality* export option.

**Annotations** Specify how annotations should be exported:

*Do not include* Annotations are ignored.

*Include as separate value* Annotations will be included as a separate column in the file.

## Parameter selection

The historic export dialog will only display the first 15 selected parameters by default. Use the *Parameter selection* panel accessible via the arrow on the right to refine your selection and change how the data should

be exported, including any aggregate calculations to be applied.

The screenshot shows the 'Parameter selection' dialog box. It has two tabs: 'Auto' and 'Custom'. Below the tabs, there is a list of parameters. Each parameter has a checkbox, a 'RAW' button, and a dropdown arrow. The first parameter, 'ce/Sample Location/Text File Source/pH', is expanded, showing its configuration options: 'Display type' (Value/State), 'Aggregate' (Average), 'Interval' (1H), and 'Base time' (D). There are 'Apply' and 'Apply to all' buttons at the bottom of the expanded section. The other three parameters are also checked and have 'RAW' buttons, but their dropdown arrows are not expanded.

**Display type** *Value* displays the parameter values.

*State* displays the state name associated with each value.

**Aggregate** Aggregates provide a method to resample raw data over a specified period. See the [list of available aggregates](#) (page 569) for full details. Select RAW to display the original raw data with no aggregation applied (Interval and Base time options will be hidden).

**Interval** Specify the data range that is included for each calculated value. Example *30M* will calculate a value for each 30 minute window of data.

**Base time** Base time is used to determine the start of the aggregate calculations. Example *D+9H* will start the calculations at 9am (start of current day + 9 hours). See [working with relative time](#) (page 585) for full details.

## 2.8.5 Import



Historic Import is accessible from the Data Source context menu in the Workspaces Tree. You can upload/import historic data from a locally accessible file, into New or Existing [Parameters](#) (page 147), or update the historic coordinate data for [Locations](#) (page 56).

For automatic collection of your data you should configure the [Connection](#) (page 115) and [Collection](#) (page 119) properties of a file Data Source.

The Historic Import Wizard will guide you through the following steps to successfully import your time series data:

### Select a file

Select a file to import from your computer or device and configure import options.

Historic Import Wizard

1 File selection 2 Configure parser 3 Configure parameters 4 Process and upload

File type: Delimited Text

Select file: Browse aqua\_data.csv

Write mode: Merge and overwrite

Timezone: (GMT+00:00) UTC

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Next Cancel

**File type** Select the type of file you would like to import:

*Delimited Text*

Data is stored as rows of values separated with a specific *delimiter* character.



### *JSON Time Series*

Data is stored in the *JSON Time Series* (page 572) format. Note: JTS files to be imported must contain a *columns* header that specifies *name* and *dataType* of each column included in the *data*.

**Select file** Click or touch the *Browse* button to select a file from your computer or device. There is a 250MB file size limit per import. To import a file exceeding the size limit you should split it into smaller chunks (< 250MB) and import separately.

**Write mode** Select how the new data will be imported:

*Merge and overwrite* [API: MERGE\_OVERWRITE\_EXISTING]

Insert the imported data into the existing historic data and overwrite the existing values when timestamps match.

*Merge and preserve* [API: MERGE\_PRESERVE\_EXISTING]

Merge the imported data into the existing historic data and do not overwrite existing values when timestamps match.

*Replace* [API: INSERT\_DELETE\_EXISTING]

Remove all existing historic data within the range of data being imported and replace with imported data.

[API: MERGE\_UPDATE\_EXISTING]

Merge the imported data into the existing historic data and update specific record attributes when timestamps match.

[API: MERGE\_FAIL\_ON\_EXISTING]

Merge the imported data into the existing historic data but fail if timestamps match.

[API: INSERT\_FAIL\_ON\_EXISTING]

Insert new data only if there is no existing historic data within the range being imported.

[API: DELETE\_RANGE]

Delete historic data encapsulated by the range of the imported data.

[API: DELETE]

Delete historic data that matches the timestamps of the imported data.

**Timezone** Select the timezone of the data being imported. This is generally the location of the sensor or device, but can be adjusted as required.

**Ignore Daylight Savings Time** Check this option if the Source does not offset the data timestamps to account for DST.

## Use the Text Parser to define Series

The Text Parser allows you to define how the delimited text file being imported should be processed including defining series to be assigned to Parameters. See [Text Parser Configuration](#) (page 123) for a detailed guide to using the Text Parser.

Historic Import Wizard Parser Configuration aqua\_data.csv

1 File selection 2 **Configure parser** 3 Configure parameters 4 Process and upload

Column delimiter: , Labels row: 1 Units row: None Text qualifier: " Text encoding: UTF-8 (default) Reset to defaults

	Timestamp	pH	Temperature	Conductivity	Redox
2	2013-10-02 16:10:40	8.09	28.5	44.8	-158
3	2013-10-02 16:11:40	8.09	28.5	44.7	-158
4	2013-10-02 16:12:40	8.09	28.5	44.7	-158
5	2013-10-02 16:13:40	8.09	28.5	44.7	-158
6	2013-10-02 16:14:40	8.09	28.5	44.7	-158
7	2013-10-02 16:15:40	8.09	28.5	44.7	-158
8	2013-10-02 16:16:40	8.09	28.4	44.8	-158
9	2013-10-02 16:17:40	8.09	28.4	44.8	-158
10	2013-10-02 16:18:40	8.09	28.4	44.7	-158
11	2013-10-02 16:19:40	8.09	28.4	44.7	-158

Get help with parser configuration Previous Next Cancel

## Assign Series to Parameters and Location

Assign the newly defined Series to New or Existing Parameters. You can also assign the series to the Source's Location for the import of historic coordinate data.

**Historic Import Wizard**

1 File selection 2 Configure parser 3 **Configure parameters** 4 Process and upload

Location coordinates: ⊗ None

aqua\_data.csv

<input checked="" type="checkbox"/> Series	Parameter assignment		Units
<input checked="" type="checkbox"/> pH	New	Existing	pH
<input checked="" type="checkbox"/> Temperature	New	Existing	Temperature °C
<input checked="" type="checkbox"/> Conductivity	New	Existing	Conductivity mS
<input checked="" type="checkbox"/> Redox	New	Existing	Redox mV

Previous Next Cancel

**Location Coordinates** The imported file can optionally provide Location Coordinates from a coordinate series or a pair of numeric series (lat/lng) and will import the data into the Location Node this Source is contained within.

**Parameter Assignment** Assign Series to New or Existing parameters by toggling the New/Existing button (only enable when there are existing Parameters matching the data type of the Series). Uncheck any Series you do not want to import.

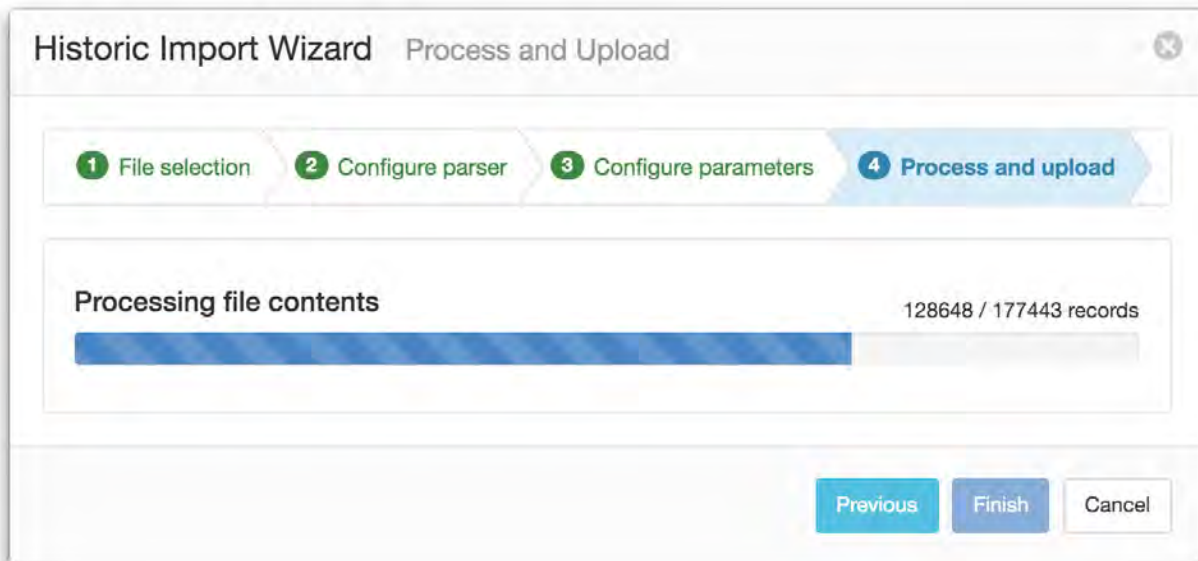
Any newly created parameters will be created within the Source and shown as GREYED out in the Workspaces Tree and Content Views to distinguish them from parameters assigned to series being automatically collected by the Data Source.

Parameters can be re-assigned to new Series at any time without losing existing historic data.

The series icon indicates the *type of parameter* (page 49) that can be assigned to the Series.

## Process and Upload

The Process and Upload screen displays a progress bar and status information about the current operation. If any problem is encountered an error message and *Retry* button will be displayed, otherwise the *Successfully Completed* status message will be shown. Click *Finish* to close the import wizard.



---

**Note:** Files containing a large number of records may be time consuming to process. Some browsers/devices may also enforce certain limits which prevent successful import of large files even if they are smaller than the 250MB limit.

---

## 2.9 Sharing and Security

Every interaction with the system is authenticated to ensure the user performing the interaction or operation has permission to do so.

Eagle.io provides 3 distinct types of user security in addition to anonymous public access:

### 2.9.1 Owner

The user that signed up for the account. Only this user can close the account or change the subscription plan. The user has full unrestricted access to all assets and account settings including *Managed accounts* (page 283).

### 2.9.2 Administrator

A user can be *added as an Administrator* (page 269) of an account which provides them full unrestricted access to all assets and account settings excluding closing the account or changing the subscription plan. Administrators also have full access to *Managed accounts* (page 283).

### 2.9.3 Workspace

Individual users or *groups* (page 269) can be *added to a Workspace* (page 54) and assigned a security *role* (page 270) specific to the Workspace (role-based security). The security roles are maintained by administrators and the owner. Each role has a set of security permissions assigned (e.g. acknowledge alarms, export data, configure) which define the interactions allowed by a user or group for all assets (nodes) in the Workspace.

Workspaces can be shared with anyone. Users new to [eagle.io](https://eagle.io) (<https://eagle.io>) will receive an email notification prompting them to confirm their email address and configure their personal preferences. Select *Security* from the Workspace Node context menu to display the Security dialog. See *Workspace Security* (page 54) for full details.

You can restrict user and group access to any Node in the Workspace by opening the corresponding *Node Security* (page 58) dialog and assigning the *No Access* role.

---

**Note:** The *Events View* (page 43) provides an audit trail for all user interactions and system events.

---

### 2.9.4 Public Access

You can *enable public access* (page 102) on specific nodes to provide a URL for anonymous public access to the resource. Public access can be enabled on custom charts, dashboards, attachments and parameters.

Use the public access link for standalone viewing of the content in a web browser or embed the content in your own website.

---

**Note:** The Public Access feature is not included in *Starter* accounts.

---

## 2.10 Processing and Logic

Processing and Logic provides a programming environment to perform calculations, make decisions and take actions.

Some examples include:

- *Calculate average of multiple Parameters* (page 224)
- *Transform a series using an equation* (page 224)
- *Generate a series forecast to provide predictive alarming* (page 226)

Programs are expressed as **JavaScript** (<http://www.ecma-international.org/ecma-262/5.1/>), stored as configuration in Process Nodes, and executed on a schedule or automatically as new data is acquired. Each program can interact with all Nodes in the current Workspace and any associated time-series data.

### 2.10.1 Process Nodes

There are two types of Nodes that can have a program defined as part of their configuration. The first is a Process Parameter that allows you to transform existing data; the second is a type of Data Source called a Processor that allows you to create new data values.

A program should include three distinct stages:

- Define input nodes (using the **NODE**, **NUMBER**, **TEXT**, **TIME** or **BOOLEAN** functions)
- Apply processing to input nodes (or associated time-series data)
- Output results (either by returning a value or setting an attribute of a node)

	Process Parameter	Processor
<b>Node Type</b>	Parameter	Data Source
<b>Number of inputs</b>	3	25
<b>Number of outputs</b>	1	100
<b>Execution trigger</b>	On input update	On input update or schedule
<b>Maximum Duration</b>	0.2 seconds	2 seconds
<b>Output timestamps</b>	Defined by first input	Defined by first input or schedule



#### Process Parameter

A Process Parameter is a type of Parameter that has a program defined as part of its configuration and can be created under any Data Source. The first input referred to by the program (known as the primary input) will be used to define the timestamps used by the output of the process. Each execution of the program must return a single output value of the expected type. This output can then be viewed in the same way as any other Parameter (for example, it can be shown as a table or chart, exported, used as input for a different process etc.). The output can also have states defined which will trigger alarms.

## Examples

```
1 // Calculate the average currentValue of Parameters from different Locations
2 var param1 = NODE('Location 1/Source/Param');
3 var param2 = NODE('Location 2/Source/Param');
4 var param3 = NODE('Location 3/Source/Param');
5
6 return (param1 + param2 + param3) / 3;
```

```
1 // Transform the currentValue of Param using a 3rd order polynomial
2 var a = 7.24;
3 var b = -10.004;
4 var c = 4.328;
5 var d = -0.4667;
6 var v = NODE('Param').currentValue;
7
8 return a + (b*v) + (c * Math.pow(v,2)) + (d * Math.pow(v,3));
```

```
1 // Assign a bad quality code to value spikes and offset the timestamp by one_
   ↪hour
2 var param = NODE('Param');
3 var ts = T(param.currentTime).subtract(1, 'hours');
4 var v = param.currentValue;
5 var q;
6
7 if( v > 999 ) {
8     q = 156;
9 }
10
11 return {"time": ts, "value": v, "quality": q};
```

---

**Note:** Assigning quality codes to value spikes can also be achieved by configuring the Quality of a *Parameter State* (page 150).

---



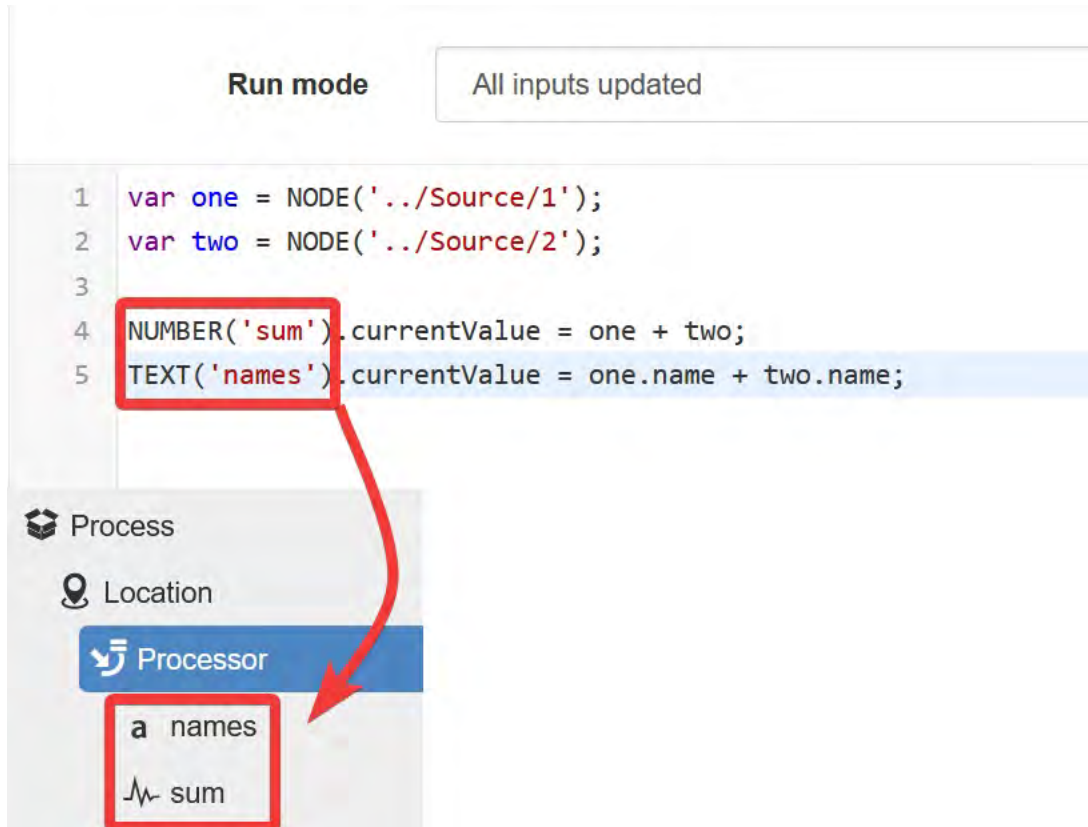
A Processor is a type of Data Source containing a program that generates multiple output values, which are stored as new Parameters. The timestamps of the output data can be generated by scheduling the execution of the process, in the same way that data acquisition of a regular Data Source can be scheduled. Alternatively, the timestamps can be defined by the first input, just as with Process Parameters.

## Output Creation

Outputs are defined as part of your program and automatically created as Parameters under your Processor when the program is saved. To update a Processor output, use one of the four Node creation functions



(NUMBER, TEXT, TIME or BOOLEAN) to obtain a reference to the output, then assign a value to a specific attribute, e.g. `TEXT('output').currentValue = "Hello world";`.



## Examples

```
1 // Assign values to multiple outputs
2 var quotient = NODE('../Source/Param') / 5;
3 var remainder = NODE('../Source/Param') % 5;
4
5 NUMBER('quotient').currentValue = quotient;
6 NUMBER('remainder').currentValue = remainder;
```

```
1 // Assign a bad quality code to value spikes and offset the timestamp by one_
  ↳hour
2 var param = NODE('Param');
3 var output = NUMBER('Output');
4 var q;
5
6 if( v > 999 ) {
7     q = 156;
```

(continues on next page)

(continued from previous page)

```
8 }
9
10 // NOTE: currentTime & currentQuality must be assigned prior to currentValue
11 output.currentTime = T(param.currentTime).subtract(1, 'hours');
12 output.currentQuality = q;
13 output.currentValue = param.currentValue;

1 // Calculate average currentValue of all nodes ending with Temperature in the
  ↳current Workspace
2 var sum = 0;
3 var temps = NODES(WORKSPACE + '/*Temperature');
4 var avg = NUMBER('Output');
5
6 for( var i=0; i<temps.length; i++ ) {
7     sum = sum + temps[i];
8 }
9
10 avg.currentValue = sum / nodes.length;
```

## 2.10.2 Environment

### Global Variables

Global variables are references to Nodes and other values that are related to the currently executing process.

<b>NOW</b>	Current timestamp being evaluated by the process
<b>THIS</b>	Reference to currently executing <i>Process Node</i> (page 223)
<b>SOURCE</b>	Reference to Data Source of currently executing process
<b>LOCATION</b>	Reference to Location of currently executing process
<b>WORKSPACE</b>	Reference to Workspace of currently executing process

### Global Functions

Global functions can be used to obtain a reference to a Node, authorize access to a Workspace or convert a time expression.

<b>NODE</b> ( <i>path</i> )	Retrieve node by path
<b>NODES</b> ( <i>path expr</i> )	<p>Retrieve one or more nodes by path expression</p> <p>A path expression begins with an ordinary <i>path</i> (page 227) and includes a <i>regular expression</i> (<a href="https://en.wikipedia.org/wiki/Regular_expression">https://en.wikipedia.org/wiki/Regular_expression</a>) suffix matching one or more node names, e.g. retrieve all workspace nodes ending with <i>input</i>: /Workspace/.<i>*input</i></p> <p>Retrieving nodes from outside the current Workspace is not supported</p>
<b>NUMBER</b> ( <i>path</i> )	Create or retrieve NUMBER Parameter by path
<b>TEXT</b> ( <i>path</i> )	Create or retrieve TEXT Parameter by path
<b>TIME</b> ( <i>path</i> )	Create or retrieve TIME Parameter by path
<b>BOOLEAN</b> ( <i>path</i> )	Create or retrieve BOOLEAN Parameter by path
<b>AUTH</b> ( <i>slug, api-key</i> )	<i>Authorise access to a Workspace</i> (page 228) using an API key
<b>T</b> ( <i>expr</i> )	<p>Convert a time expression to a <i>Moment.js</i> (<a href="https://momentjs.com">https://momentjs.com</a>) timestamp</p> <p>A time expression can be any of the following:</p> <ul style="list-style-type: none"> <li>• ISO8601 time string, e.g. '2018-08-03T16:27:58+10:00'</li> <li>• Number of milliseconds since Unix epoch, e.g. 1533277715816</li> <li>• Node time attribute, e.g. <code>NODE('param').currentTime</code></li> <li>• Current timestamp, e.g. NOW</li> </ul>
<b>Q</b> ( <i>name</i> )	Convert a quality name to a quality code

## Paths

A path describes a reference to a Node and can be expressed as an absolute path, or a path that is relative to the current Process Node. Standard UNIX style path syntax is used, so `..` refers to the parent Node in the Workspace tree.

In addition to using names to identify Nodes, a path may also contain either a *Node Id* (page 444) or *Custom Id* (page 444). Named elements may also be included after the Node identifier. Note that Node identi-

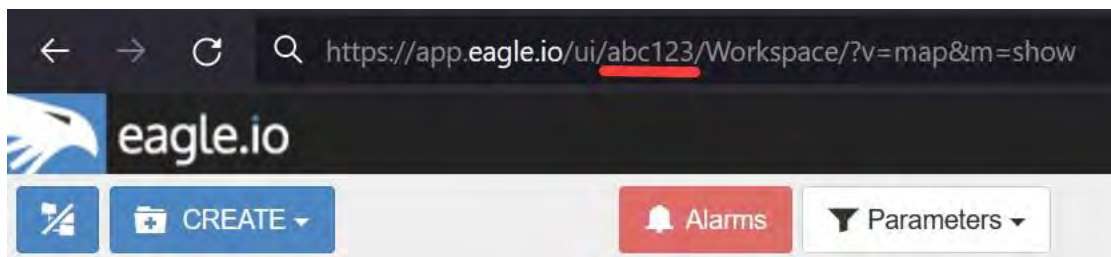
ifiers should be enclosed by braces when included in a path, e.g. {5ae92a139097830ee5711d94} or {@customId}.

Examples	
/Workspace/Location/Source/Parameter	Absolute path to a Parameter
../Location 2/Source/Parameter	Path to a Parameter, relative to the Process Node
{5b6a3fc24e960d0e7497b4b4}	Path to a Node, identified by <b>Node Id</b>
{@myParam}	Path to a Parameter, identified by <b>Custom Id</b>
{5ae92a139097830ee5711d94}/Parameter	Path to a Parameter, relative to Source <b>Node Id</b>
{@mySource}/Parameter	Path to a Parameter, relative to Source <b>Custom Id</b>

## Workspace Authorisation

Paths that reference a Node outside the current Workspace (including Nodes in other accounts) require authorisation to verify your process has the appropriate permission. This is achieved using the *AUTH* function, which must be used in your program before referencing any Nodes from the foreign Workspace.

The *AUTH* function declares an association between an account identifier (or *slug*) and an *API key* (page 272). The *slug* for an account is visible in your browser address bar, immediately preceding the Workspace name, when any Node is selected.



The following code snippet demonstrates declaring authorisation to an account (with slug abc123) and accessing a Node within the foreign Workspace.

```
1 // Authorise access to a node in a foreign Workspace
2 AUTH('abc123', 'eMli7Pugp8EpSQLAXiDT3DJJUAk2mOn17guVeTu');
3
4 return NODE('/abc123/Workspace/Location/Source/Parameter').currentValue;
```

---

**Note:** If all the Node paths in your program are in the same Workspace as your program, then the *AUTH* function is not required.

---

## Aggregate Expressions

Aggregate Expressions provide a means of aggregating input data supplied to your program and may be optionally included as a suffix to any parameter path.

The three components of an aggregate expression must be expressed in order and separated by a semi-colon:

- *Aggregate Type* (page 569)
- *Base Time* (page 585)
- *Interval* (page 585)

Examples	
Param 1	Raw data
Param 1; AVERAGE; D; 1H	Hourly average
Param 1; TOTAL; D+9H; 1D	Daily total calculated at 9am
Param 1; COUNT; W; 1W	Number of values since the start of the week

## Node Attributes and Values

A Node reference can be used to access the attributes of that Node using dot notation, including the data value if the Node is a Parameter.

Examples	
LOCATION.name	Name of the Location
WORKSPACE.createdTime	Creation time of the Workspace
NUMBER("param1").offset	Numeric offset of the Number Parameter
NODE("param2").currentValue	Current data value of the Parameter
NODE("param3").newestTime	Newest timestamp of the Parameter

A full reference of *Node attributes* (page 442) is documented as part of the HTTP API. Note that for calculations requiring the time of the newest value, this can be accessed via the `newestTime` attribute.

## Implicit Node Values

Each type of Node reference can be used as an implicit value without using dot notation. For example, the implicit value field of a Parameter is `currentValue`, so the Node reference can be used as a direct substitute for the current data value of the Node. This means the following two statements will return the same result:

Statement 1, access the current data value of a Node reference using dot notation, add 10 and return the result:

```
return NODE("param1").currentValue + 10;
```

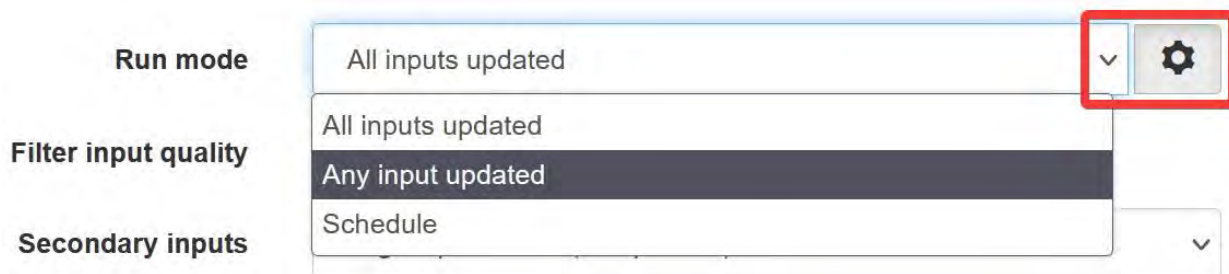
Statement 2, access the current data value of a Node reference using the implicit Node value, add 10 and return the result:

```
return NODE("param1") + 10;
```

The above example is able to treat the Node reference for **param1** as if it were a number, because this Node is a Number Parameter. Note that the type of any specific Node is always the same regardless of how the Node is referenced. This means that using the global functions `NUMBER("param1")` and `NODE("param1")` will both return a Node reference of type Number Parameter, assuming **param1** is a Number. Use care when relying on implicit Node values, because the implicit value field and type is different for different types of Nodes.

Node Type	Implicit value field	Implicit value type
<b>Number Parameter</b>	currentValue	Number
<b>Text Parameter</b>	currentValue	String
<b>Time Parameter</b>	currentValue	Time
<b>Location</b>	currentValue	Array of [latitude,longitude] decimal values
<b>Source</b>	name	String
<b>Folder</b>	name	String
<b>Workspace</b>	name	String

## Process Trigger



A process can be triggered to run in a number of ways:

Schedule	At a configurable interval (Processor only)
Any input updated	When any input advances beyond the current time of the process
All inputs updated	When all inputs have advanced beyond the current time of the process
Acquire Now	User action (causes Process Parameters to recalculate entire series)

---

**Note:** Each process timestamp is evaluated exactly once and only considers input data available at that time. Input data arriving after a process has advanced (e.g. Any input updated) may yield unexpected results.

---

## Input Quality Filter

Input data to your program can be filtered according to its quality type to ensure only desired values are considered.

**Run mode** All inputs updated

**Filter input quality**

**Secondary inputs**

**Aggregate inputs**

- ☐ Select all
- ☒ Uncertain
- ☒ ★ Unknown
- ☒ ★ Uncertain - Subnormal [149]
- ☐ Bad
- ☐ ★ Bad - Invalid [0]
- ☐ ★ Bad - No Data [155]
- ☐ ★ Bad - No Bounds [156]

By default, your program will exclude quality types according to the [Quality Codes](#) (page 266) configuration in your Account Settings. To ensure specific quality types are considered, a filter can be applied.

## Secondary inputs

**Run mode** All inputs updated

**Filter input quality**

**Secondary inputs**

**Aggregate inputs**

- Align input values (interpolated)
- Retain most recent input values

Secondary inputs may have records that occur at different times than the primary input, e.g. parameters from different data sources. Input data not aligned with the current execution timestamp can be interpreted by a process in two ways:

Align input values (interpolated)	Secondary values interpolated to align with the current execution timestamp
Retain most recent input values	Secondary values prior to the current execution timestamp are retained

Aggregate inputs

Run mode

All inputs updated

Filter input quality

Secondary inputs

Align input values (interpolated)

Aggregate inputs

Complete & partial intervals

Complete & partial intervals

Complete intervals only

Process aggregate inputs will have values that occur at a defined interval. Process execution can be delayed until the latest aggregate interval is complete:

Complete & partial intervals	Process will execute even during a partial aggregate interval
Complete intervals only	Process will only execute when the aggregate interval is complete

Shared Functions

Common processing routines can be defined as Shared Functions on the Processing tab of your Workspace configuration and accessed from any Process Node. Note that a Shared Function should not reference Global Variables or Global Functions.

```
1 // Convert Fahrenheit to Celsius
2 function toCelsius( f )
3 {
4     return (5/9) * (f-32);
5 }
```



## Third Party Libraries

A number of useful third-party libraries have been included to simplify common processing operations. Please refer to the documentation provided by each library for specific examples.

Li-brary	Ver-sion	Description
everpolate	0.0.3	<a href="http://borischumichev.github.io/everpolate/#step">Step Function Interpolation</a> ( <a href="http://borischumichev.github.io/everpolate/#step">http://borischumichev.github.io/everpolate/#step</a> ) <a href="http://borischumichev.github.io/everpolate/#linear">Linear Interpolation</a> ( <a href="http://borischumichev.github.io/everpolate/#linear">http://borischumichev.github.io/everpolate/#linear</a> ) <a href="http://borischumichev.github.io/everpolate/#poly">Polynomial Interpolation</a> ( <a href="http://borischumichev.github.io/everpolate/#poly">http://borischumichev.github.io/everpolate/#poly</a> ) <a href="http://borischumichev.github.io/everpolate/#regression">Linear Regression</a> ( <a href="http://borischumichev.github.io/everpolate/#regression">http://borischumichev.github.io/everpolate/#regression</a> )
moment	2.22.2	<a href="https://momentjs.com">Parse, validate, manipulate, and display dates and times</a> ( <a href="https://momentjs.com">https://momentjs.com</a> )

These libraries can be accessed by your program using the `require` keyword, for example:

```
var linear = require('everpolate').linear;
```

### 2.10.3 Process Alarm

A Process Alarm is raised when a Process Node encounters an error either during validation or execution. A subsequent successful validation or execution of the process will clear the alarm.

### 2.10.4 Errors

The two general category of errors that can be encountered with Processing and Logic are validation errors and runtime errors.

#### Validation Errors

Validation errors are caused either by incorrect syntax or some other error condition that can be detected. These errors are experienced as immediate feedback when validating a program, and contain a specific error message which can be used to remedy the problem. A program will not be executed until it can be validated without errors.

#### Runtime Errors

Runtime errors can occur during the execution of a program even when it validates successfully. For example, if an input node referenced by the program is deleted from the workspace, the program will no longer be able to run successfully. These types of errors will be expressed as process alarms, and will contain a specific error message to help remedy the problem.

It is also possible to manually trigger a Runtime Error by using the `throw` keyword in conjunction with a custom error message:

```
throw 'custom error message';
```

## 2.10.5 Best Practices

- Program syntax should confirm to [ECMAScript 5.1](http://www.ecma-international.org/ecma-262/5.1/) (<http://www.ecma-international.org/ecma-262/5.1/>)
- Inputs should be declared before they are referenced, so that any line numbers in error messages will clearly refer to the declaration of a missing input.
- Very complex or time-consuming calculations may cause the process to exceed the allowed processing time limit.
- **Any** input that is referenced by a process will trigger execution of the process when that input is updated. Therefore, a large number of inputs being updated frequently or on different schedules can trigger a process to run very frequently. For example, if 9 inputs are updated every hour, but the 10th input is updated every minute, then the process will execute every minute.
- As the first referenced input is used to determine the output timestamp for a Process Parameter, the input which updates most frequently should be the first input.
- If the same algorithms are used repeatedly for different Process Nodes, this code should be expressed as a function and stored in the Workspace *Shared Code* (page 232).

## 2.11 Reports



Generate PDF reports using your data for download on-demand or schedule for email delivery.

---

**Note:** The reporting feature is currently a **pre-release** version that is subject to change without notice and is only available to limited accounts.

---

Refer to the *Quick Start* (page 234) guide to get up and running with reporting.

### 2.11.1 Quick start

This documentation is intended as a quick introduction to the reporting feature. Please refer to the *using the report editor* (page 238) section for a comprehensive guide to building reports.

#### Creating a report

1. Using the `Create` menu in the workspaces tree, select `Report`. The *report node* (page 161) will be created and the report editor will open with a blank report template.
2. Click (or drag) a report element from the editor toolbar to insert it.
3. Drag nodes from the Workspaces tree to quickly insert them into the report template. eg. Drag an image attachment node or chart node to insert as an image in the report.

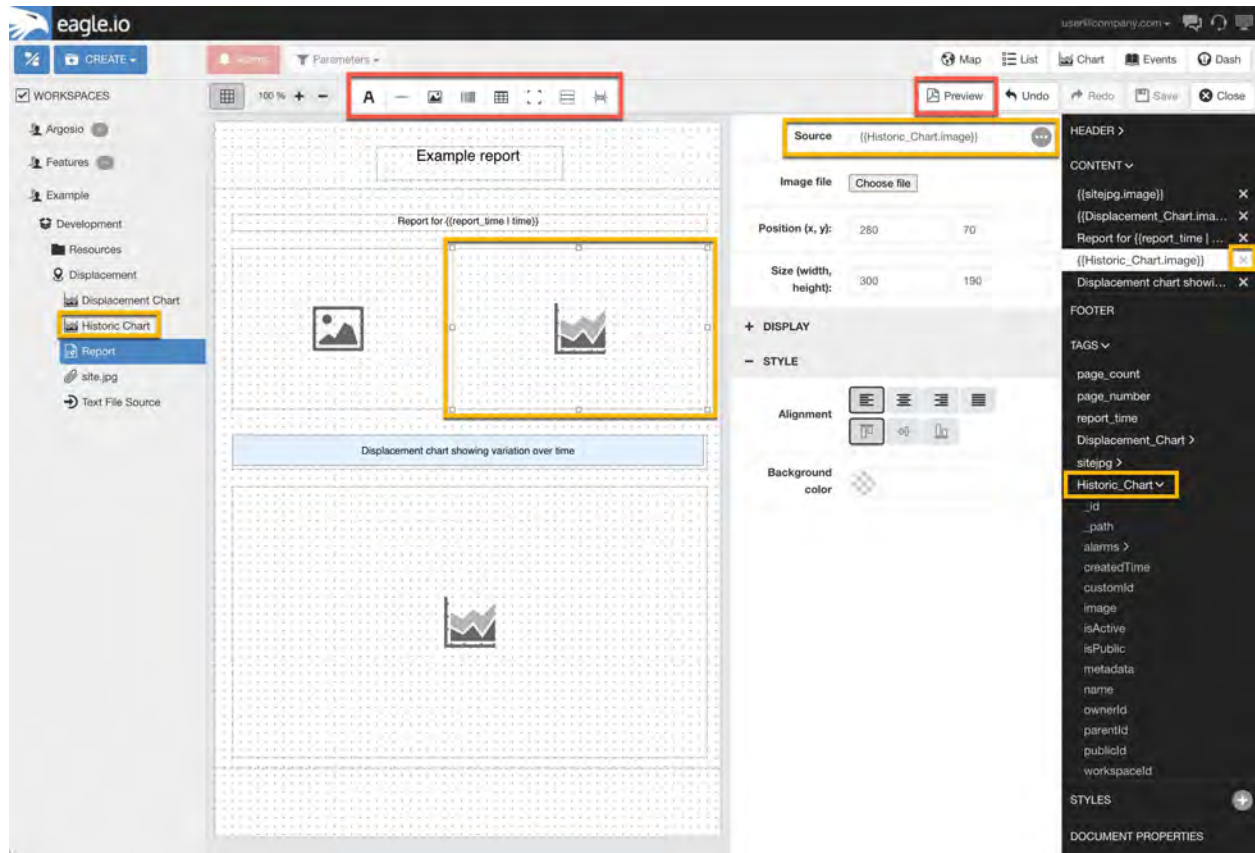
4. Remove an element by clicking the `×` next to the item in the `CONTENT` menu (on the right-hand side of the screen) or selecting the element(s) in the editor and pressing the `delete` key.
5. Insert page break elements to split content over multiple pages.
6. Click and drag the resize handle at the bottom of the page to adjust the content height so that more elements can be added to the report.
7. Click on the name of a linked node (automatically added after drag-drop from Workspaces tree) in the `TAGS` menu to display all of its available attributes.
8. Use *expressions* (page 523) directly inside report element content to control the output of the content and its formatting. A common scenario is to apply the *time* (page 545) expression filter to format the display of timestamps. eg. `{{report_time | time}}` or `{{report_time | time: "DD MMMM YYYY"}}`
9. Select *Preview* from the editor toolbar to generate a preview of the PDF report using the latest data.

---

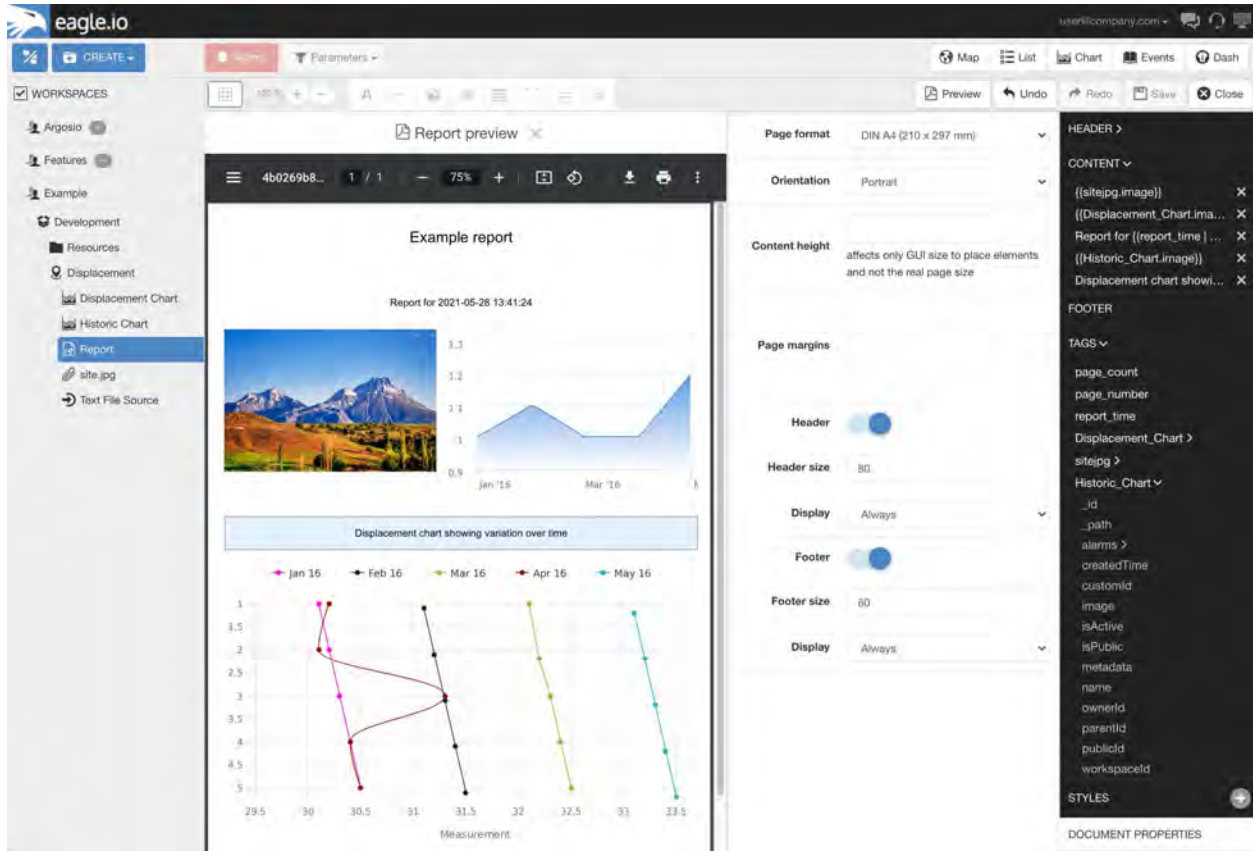
**Note:** The report editor will automatically save changes made to the template.

---

## Report editor



## Report preview



## Downloading and Scheduling

1. Locate the Report node in the Workspaces tree or from the *Reports* List content view.
2. To edit an existing report, select *Edit* from the context menu.
3. Alternatively select *Download* to generate a PDF report using the latest data.
4. Enable *Public* access to get a public URL to a report that will be generated on demand.
5. Use the [HTTP API](#) (page 483) to generate a PDF report on demand.

### To schedule a report

1. Open the [report properties](#) (page 161) from the Workspaces tree context menu.
2. From the *Schedule* section, click the checkbox to enable a schedule and select either *Daily*, *Weekly* or *Monthly*.

3. The report will be generated at the scheduled time (check *debug* level Events).
4. Users must be subscribed to the [report node](#) (page 161) to receive the report at the scheduled time (the same way users and groups are subscribed to receive sms/email notifications). See [notification subscriptions](#) (page 179) for details.
5. Use the [Events](#) (page 43) view with debug detail level to audit when scheduled reports are generated and sent to subscribed users.

*Report properties - schedule*

☒ Scheduled ?

Email PDF    Daily    **Weekly**    Monthly

MON   TUE   WED   THU   FRI   SAT   SUN

Run at   09   :   00

---

**Note:** Dashboards currently can not be displayed as images on the report. Historic table data display will be available at a later time.

---

### 2.11.2 Using the report editor

The report editor is used to build the report including the document properties, content, layout and styles.

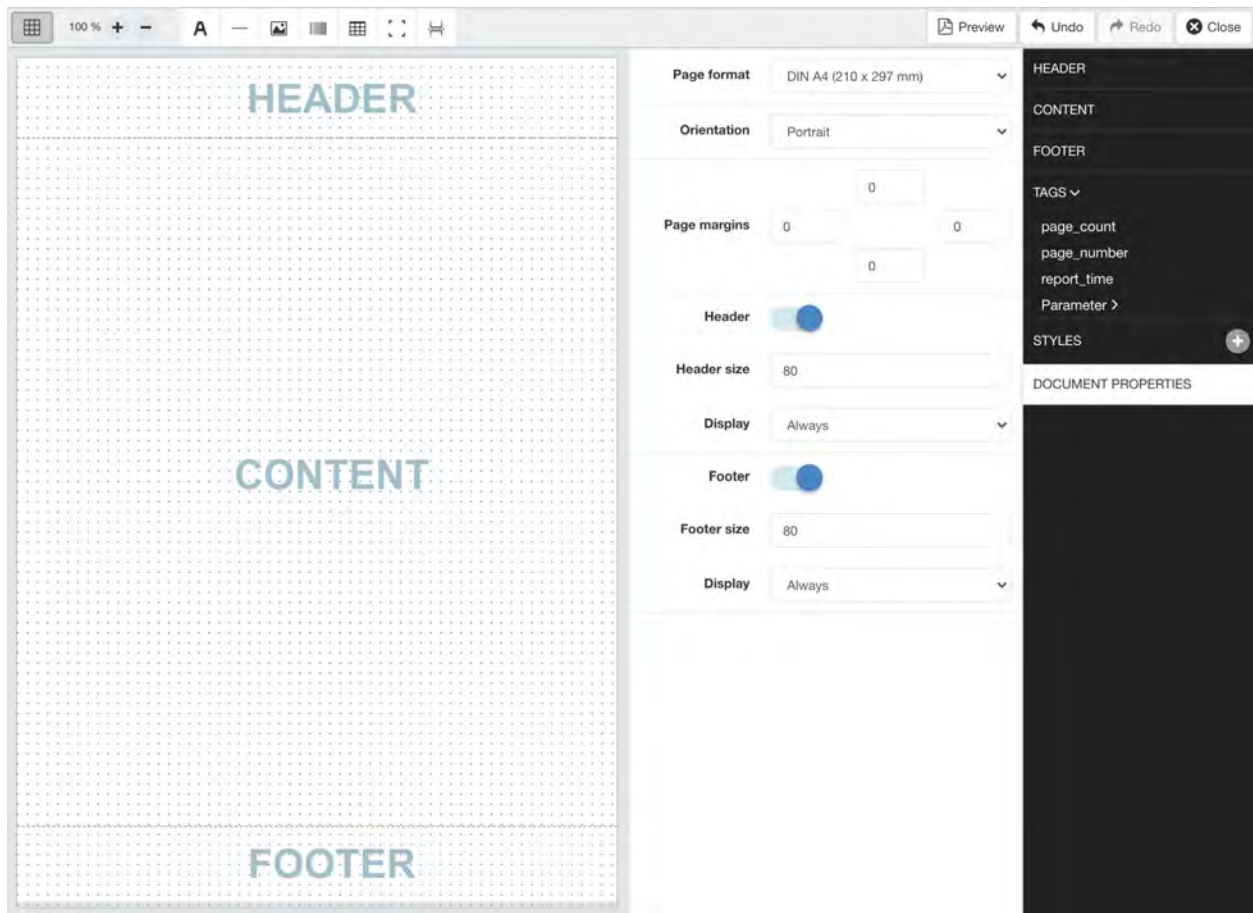
#### Quick Links

- [Navigation](#) (page 239)
- [Page basics](#) (page 245)
- [Content elements](#) (page 248)
- [Styles](#) (page 256)

- [Tags](#) (page 258)

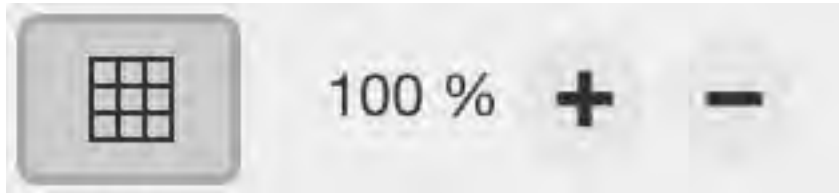
## Navigation

The report editor consists of the *toolbar* (page 239), *page designer* (page 241), *detail panel* (page 242) and *main panel* (page 243).



## Editor toolbar

## Layout controls



The `Grid` control can be toggled to show or hide the designer grid. When the grid is enabled page elements will snap to grid when moving or resizing making it easier to position elements on the page.

Use the `Zoom` `+/–` to magnify the report document.

## Content elements



*Content elements* (page 248) are added to the page designer to build the report. The available elements are covered in detail *below* (page 248).

## Operations



`Preview` will build the report and display the PDF. Click the `Design` button to close the preview and return to the *page designer* (page 241).





Reports are automatically saved as changes are made.

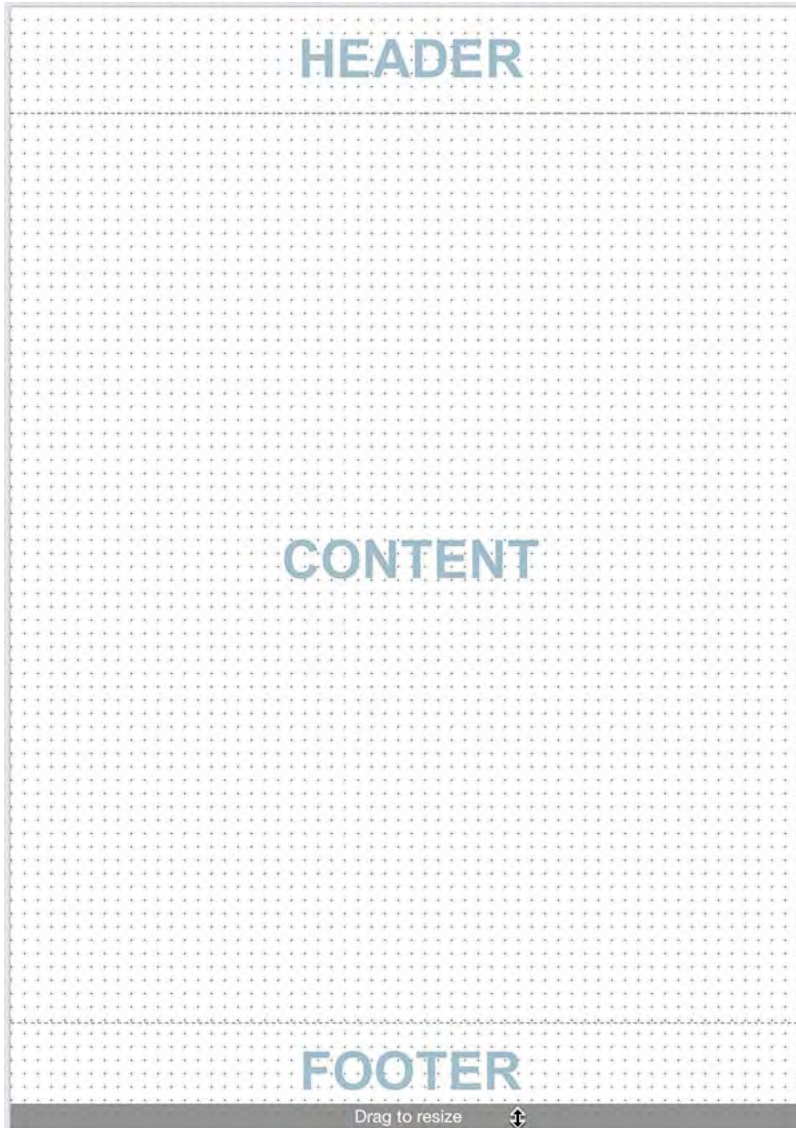
Concurrent editing of a report by multiple clients is partially supported. Any modification will trigger the report to be reloaded in all browsers with the most recent changes.

Use the `Undo` and `Redo` buttons to revert changes during the editing session and hit `Close` to exit the report editor.

## Page designer

The page designer is where the report content is assembled by adding *content elements* (page 248) to the content, *header* (page 247) and *footer* (page 247) sections.

Click and drag the resize handle at the bottom of the page to adjust the content height so that more elements can be added to the report.



## Detail panel

Selecting an item from the *main panel* (page 243) will display the associated configuration in the detail panel. Use this panel to modify the configuration of the document or the page elements and see the results in the *page designer* (page 241).

The screenshot shows a settings panel with the following sections:

- Page format:** A dropdown menu set to "DIN A4 (210 x 297 mm)".
- Orientation:** A dropdown menu set to "Portrait".
- Page margins:** Four input fields, all containing the value "0".
- Header:** A toggle switch that is turned on (blue).
- Header size:** An input field containing the value "80".
- Display:** A dropdown menu set to "Always".
- Footer:** A toggle switch that is turned on (blue).
- Footer size:** An input field containing the value "80".
- Display:** A dropdown menu set to "Always".

## Main panel

The main panel allows for easy selection of content elements listed under the HEADER, CONTENT and FOOTER menus. [Tags](#) (page 258), [Styles](#) (page 256) and [Document properties](#) (page 245) can also be accessed from this panel.

HEADER

CONTENT

FOOTER

TAGS ▾

page\_count

page\_number

report\_time

Parameter >

STYLES



## Page basics

The DOCUMENT PROPERTIES menu contains general page and display options.

<b>Page format</b>	DIN A4 (210 x 297 mm) ▼		
<b>Orientation</b>	Portrait ▼		
<b>Page margins</b>	0	0	0
<b>Header</b>	<input checked="" type="checkbox"/>		
<b>Header size</b>	80		
<b>Display</b>	Always ▼		
<b>Footer</b>	<input checked="" type="checkbox"/>		
<b>Footer size</b>	80		
<b>Display</b>	Always ▼		

## Page size and document height

The `Page format` and `Orientation` options specify the output page settings for PDF documents. Typically a report would use *DIN A4*. Select *Own dimensions* to specify a custom *width* and *height*.

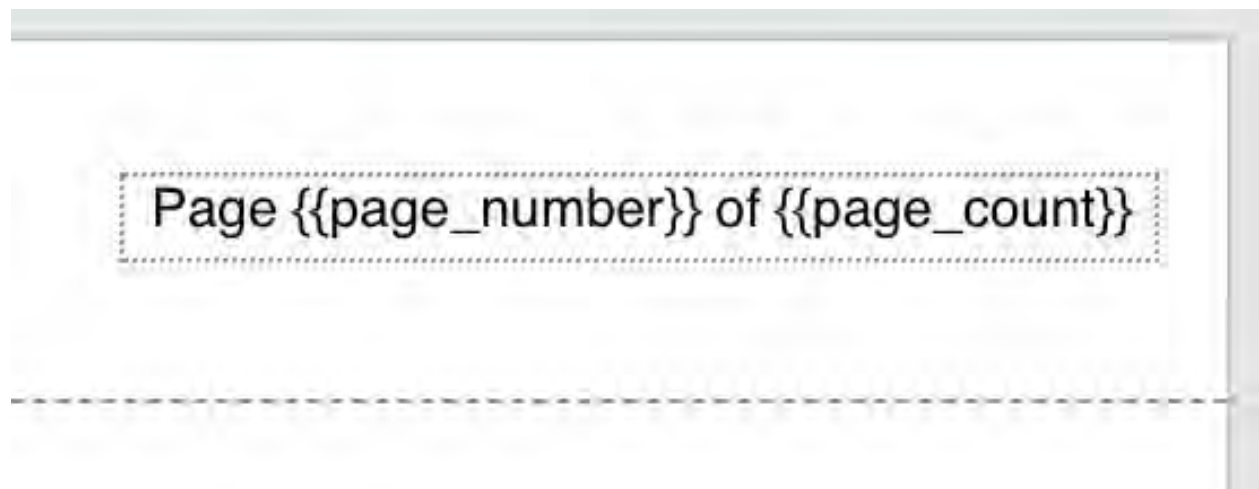
Click and drag the resize handle at the bottom of the page designer to adjust the content height so that more elements can be added to the report.

Content will automatically span over multiple pages as necessary. Alternatively insert a *page break* (page 254) element to force content to flow to a new page.

## Headers, footers and page number

Headers and footers are special content sections at the top and bottom of the page respectively. These sections are automatically included on every page by default. Use the `Header` and `Footer` toggle switches to enable or disable these sections. The `Display` option can be set to *Always* or *Do not display on first page*.

To include the **Page number** and **Page count** at the top of every page, create a new `Text` element and drag it into the document `Header`. Set the text of the element to use the `page_number` and `page_count` *tags* (page 258) as required. The text `Page {{page_number}} of {{page_count}}` will output similar to `Page 1 of 5`.



## Content elements

Click or drag-drop an element icon from the *editor toolbar* (page 239) to insert it into the *page designer* (page 241).



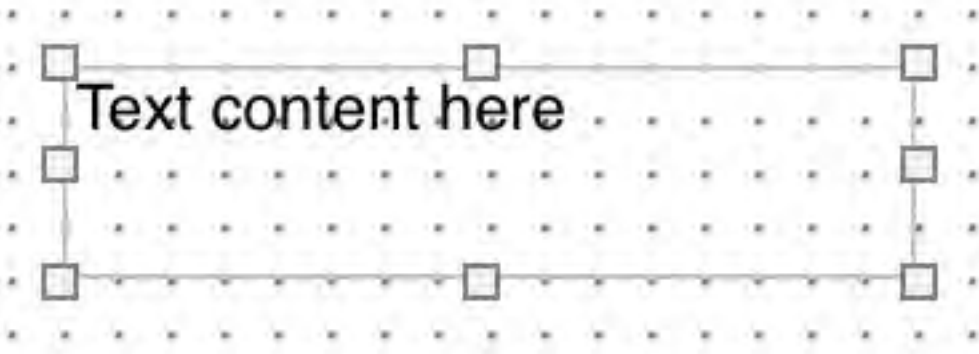
## Types of elements

Elements include *text* (page 248), *horizontal line* (page 249), *image* (page 249), *bar code* (page 250), *table* (page 251), *frame* (page 253) and *page break* (page 254).

## Text

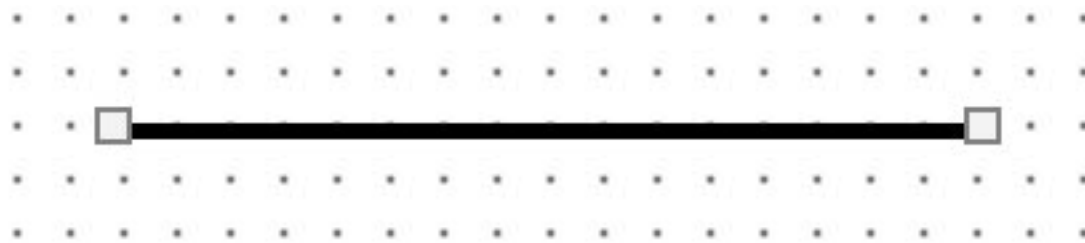
Text elements are used to add text to your report. This includes headings, labels and paragraphs.





### Horizontal line

Insert a horizontal line and adjust width and height as required.



### Image

Add images to your report. Select **Choose file** from the *detail panel* (page 242) to upload a local **JPG** or **PNG** file from your device.

Alternatively specify a **URL** or *tag* (page 258) as the image **Source**. **Charts** and **attachments** from the workspaces tree can be added to the report via *tags* (page 258).



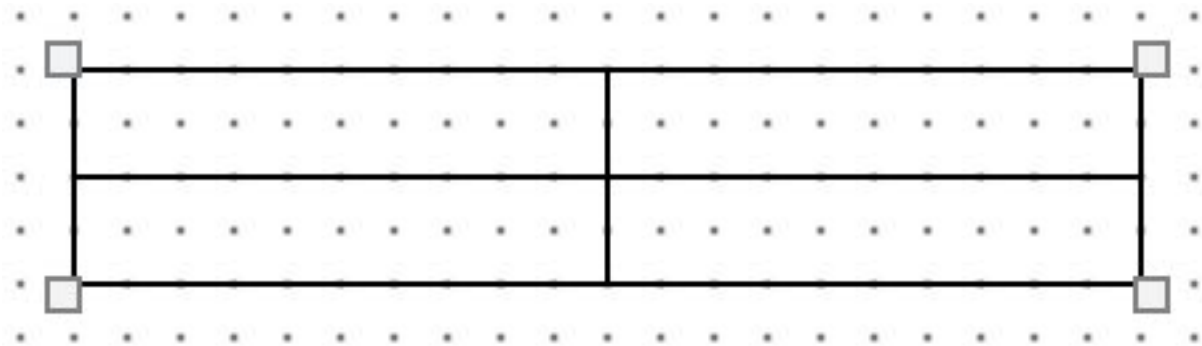
## Bar code

Insert **CODE 128** bar codes into your report. Use the `Text` field to enter the alphanumeric characters for the bar code or use *tags* (page 258) for dynamic content.



## Table

Tables make it easy to align related content.



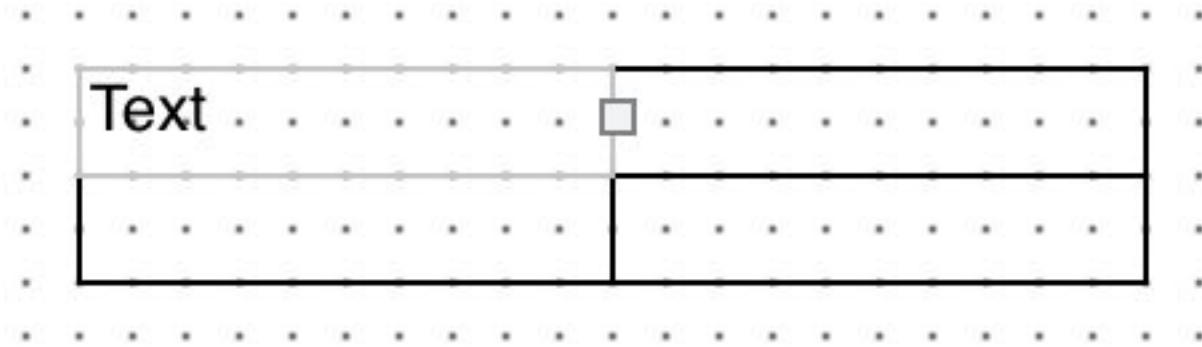
Drag the table around to position it within the report. To resize or configure the table you must select it and edit its settings from the [main panel](#) (page 243). Set the required number of `columns`, `content rows` and toggle header and footer rows as required.

Position (x, y):	<input type="text" value="370"/>	<input type="text" value="20"/>	HEADER
Columns	<input type="text" value="2"/>		CONTENT ▾
Header	<input checked="" type="checkbox"/>		Table ▾ <span>✕</span>
Content rows	<input type="text" value="1"/>		Header ▾
Footer	<input type="checkbox"/>		Text
			Text
			Content ▾
			Text
			Text

Clicking the *Table* item from the *main panel* (page 243) will expand to show individual cells grouped by header, content and footer. Select an individual cell from this panel to set its *Text* or column *Width* in pixels.

Text	<input type="text" value="Text"/>	...	HEADER
Position (x)	<input type="text" value="0"/>		CONTENT ▾
Width:	<input type="text" value="100"/>		Table ▾ <span>✕</span>
Colspan	<input type="text"/>		Header ▾
			Text
			Text
			Content ▾
			Text
			Text

As you make changes from the *main panel* (page 243), the designer will update the table accordingly.



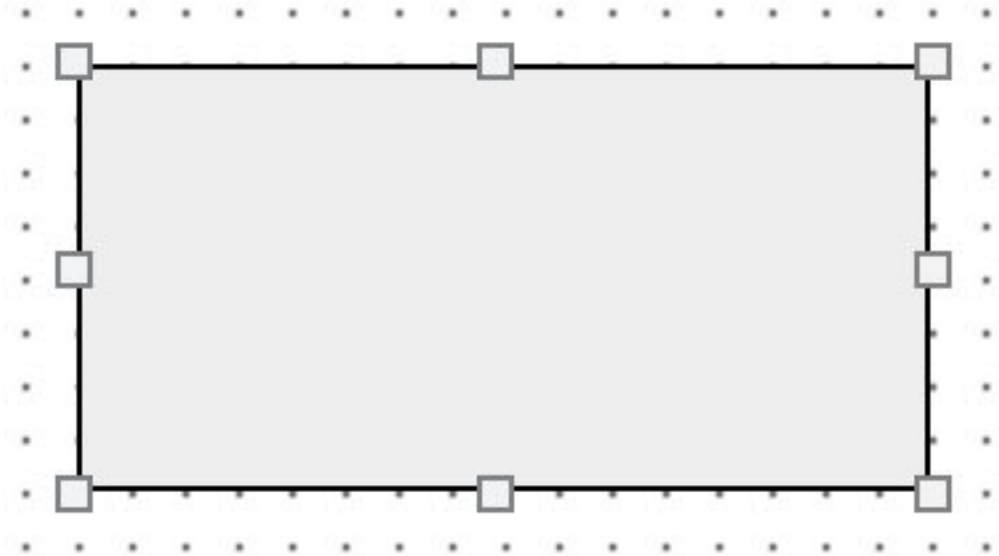
---

**Note:** You can add current parameter data values to individual cells using [tags](#) (page 258) however historic data in a table is not yet supported.

---

## Frame

Use frames to group other content elements or add backgrounds and borders to sections of your report. Drag elements into the frame to group them, then simply position the frame on the page as necessary. Select the Frame from the [main panel](#) (page 243) to modify its settings or select its grouped elements.



## Page break

Use page breaks to force content appearing below the break to flow to a new page.



## Create and delete

Click or drag-drop an element icon from the [editor toolbar](#) (page 239) to insert it into the [page designer](#) (page 241).

An element can be removed by clicking the *x* delete button next to the element in the [main panel](#) (page 243) or selecting the element and pressing the DELETE key.

Copy and Paste elements using CTRL+C and CTRL+V shortcut keys to quickly duplicate selected elements.

## Select, resize and align

Select the element by either clicking directly on the element's bounding box in the *page designer* (page 241), click and drag a selection box around elements, or clicking the element in the *main panel* (page 243). Additional elements can be added to the selection by holding the **SHIFT** key.

Drag the element(s) around the *page designer* (page 241) to **reposition** or use the **Arrow** keys for small adjustments. Click and drag the bounding box handles to **resize** an element.

**Note:** elements will automatically snap to the designer grid during reposition and resize operations but you can temporarily disable this behaviour by *toggling the grid* (page 239) off.



When multiple elements are selected, the **alignment toolbar** is available to quickly align elements together on the page.

## Element properties

When an element is selected its properties, display, style and conditional style options are displayed in the *detail panel* (page 242).

The general properties section includes the element content, position and size.

**DISPLAY** section options customise the behaviour of the element in the PDF including preventing the content from being split over multiple pages, adding hyperlinks, etc.

**STYLE** allows a predefined *style* (page 256) to be applied or customise the appearance of elements individually.

**CONDITIONAL STYLE** allows a different style (predefined or individually applied) to be set when a user-defined Condition is met. An example of a **conditional statement** is `11 > 10`, or using *tags* (page 258) `{{Parameter.currentValue}} > 10`.

## Styles

The + add style button in the *main panel* (page 243) adds new predefined styles to the report.

Headings, paragraphs and other content elements generally share similar styles. You can set the styles on every element individually but this can be tedious.

Its more efficient to predefine a set of styles and then simply select the `Style` name on each element you want to apply it to.



Name	<div>Style 1</div>				
Text style	<div><div>B</div><div>I</div><div>U</div><div>S</div></div>				
Alignment	<div><div><div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div></div>				
Text color	<div><div></div><div>#000000</div></div>				
Background color	<div><div></div></div>				
Font	<div><div>Helvetica</div><div>▼</div><div>12</div><div>▼</div><div>pt</div></div>				
Line spacing	<div><div>1</div><div>▼</div></div>				
Border	<div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div></div>				
Border color	<div><div></div><div>#000000</div></div>				
Border width	<div><div>1</div></div>				

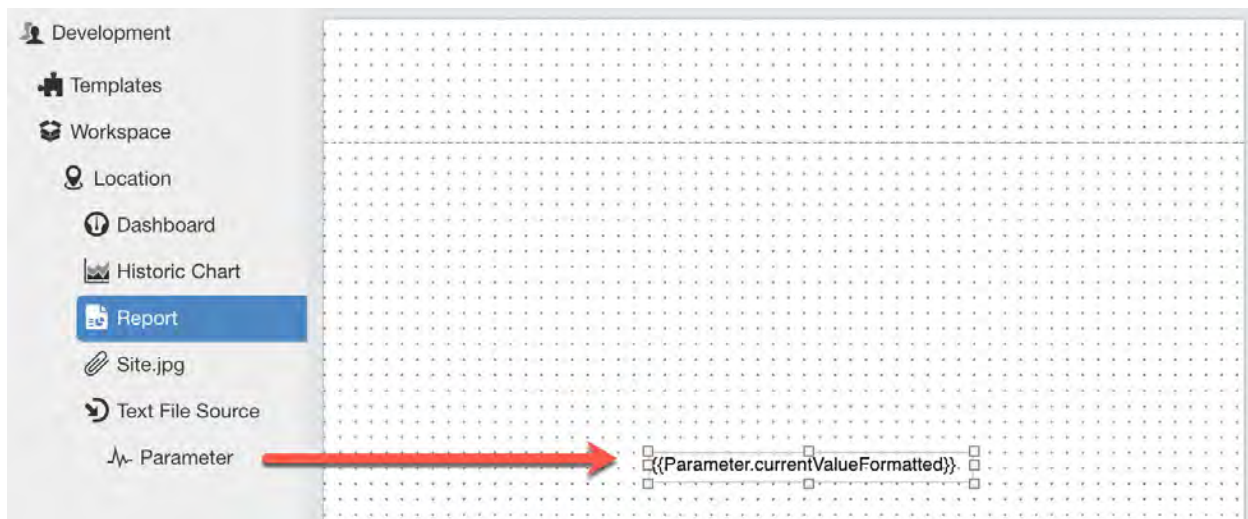
## Tags

Tags are the mechanism in which dynamic attributes related to nodes in the workspaces tree can be inserted into the report. This can include tags such as parameter value text, location coordinates, chart images, and many more.

Tags are available for all nodes that has been linked to the report.

## Drag & drop from workspaces tree

The easiest way to link a node and insert a default tag is to drag and drop the node from the workspaces tree on to the *page designer* (page 241).



---

**Note:** Only nodes within the same workspace (or template) as the report can be linked.

---

## Rename or remove links

You can remove, modify or rename links from the *report properties* (page 161). If the Name of the link is changed you will need to use the new name when assigning tags in the report.

Properties dev001/Workspace/Location/Report

Schedule Links Time Metadata Security

Drag and drop to reorder

**Parameter**  
dev001/Workspace/Location/Text File Source/Parameter  
Link will be available in report as Parameter. Hide

**Link** dev001/Workspace/Location/Text File Source/Parameter change

**Name** Parameter

Add link

ID: 60cace39fde6635ec2242df5  
@ - optional custom id - Copy URL Save Cancel

## Show available attributes

Click on the > expand arrow next to a tag in the *main panel* (page 243) to show its available attributes.

## TAGS ▾

page\_count

page\_number

report\_time

## Parameter ▾

\_id

\_path

alarms >

createdTime

currentQuality

currentStateName

currentTime

currentValue

currentValueFormatted

## Insert tags in an element

Use the content field of an element (eg. Text or Source) to either manually enter a tag or click the . . . tag list button to select from a list of available tags. Tags are always surrounded by double curly braces {{ and }}. eg. {{Parameter.currentValue}}

When the report is generated the tags are replaced with the corresponding data. Multiple tags can be inserted in a single element and are rendered inline. eg. The value is {{Parameter.currentValue}} on {{Parameter.currentTime | time: "Do MMMM"}} may be rendered in the PDF as The value is 7.12 on 10th July

**Text**`{{Parameter.currentValue}}`

Search tags...

**Tags**

page\_count

page\_number

report\_time

**Parameter**

Parameter.\_id

Parameter.\_path

Parameter.alarms

Parameter.createdTime

**Expressions**

Use *expressions* (page 523) directly inside report elements to control the output of the content and its formatting.

## Common scenarios

- Apply the [time](#) (page 545) filter to format a timestamp: `{{report_time | time}}` or `{{report_time | time: "DD MMMM YYYY"}}`
- [Round](#) (page 540) a number to 2 decimal places: `{{Parameter.currentValue | round: 2}}`
- Set a [default](#) (page 530) to insert when no value exists: `{{Parameter.currentValue | default: 0}}`
- [Truncate](#) (page 548) long text to a maximum number of characters: `{{Parameter.currentStateName | truncate: 15}}`
- Use expression [if/else](#) (page 561) tags to display alternate content based on a value

```
{% if Parameter.currentValue > 10 %}  
  Value is above expected range  
{% else %}  
  Value is within expected range  
{% endif %}
```

### 2.11.3 Scheduling

Reports can be downloaded on-demand from the [List view](#) (page 27), via the [Public access](#) (page 164) feature or using the [HTTP API](#) (page 483).

Reports can also be scheduled for **automatic delivery** to subscribed users and groups via email.

Properties dev001/Workspace/Location/Report

Schedule Links Time Metadata Security

Set up a schedule to email a report pdf to subscribed users and groups at a specified time. [Read more...](#)

☒ Scheduled

Email PDF Daily Weekly Monthly

MON TUE WED THU FRI SAT SUN

Run at 09 : 00

ID 60cace39fde6635ac2242df5

@ - optional custom id - Copy URL

Save Cancel

To schedule a report, open the *report properties* (page 161) from the Workspaces tree context menu and select the **Schedule** tab.

Enable a schedule by checking the **Scheduled** checkbox and select either **Daily**, **Weekly** or **Monthly**. Set the schedule required (maximum once per day) and **Save**.

Users and groups must be *subscribed* (page 179) to the *report node* (page 161) to receive the report at the scheduled time.

Use the *Events* (page 43) view with debug detail level to audit when scheduled reports are generated and sent to subscribed users.

## 2.12 Account Settings



Overview and Guide to Account Settings:

Use the Account selector located in the User Menu of the Account Settings section to switch between all the Accounts you own or have Administration access to.

Manage account

My Company (personal)



## 2.12.1 General



General section provides an overview of the account and allows you to change your contact details and setup quality code information.

### Overview

The account owner or manager can change the account name or view a summary of current usage.

Account

Name
Change

Account name is used in notifications sent to administrators and workspace users. It should uniquely identify your company or organization.

Type

STANDARD

Close account

Current usage

1

1

1

1

2

6

0

1

0

Billing Overview

Billing period
23rd February 2016 - 23rd March 2016

Current charges
Your current charges will be calculated within 24 hours.

**Note:** Only the account owner or manager can change the account name or close the account.

### Contact

Please ensure your contact details are always up to date so that we can contact you when required. You can optionally specify a different support phone number and email address for any public enquires or customer support requests.

**Phone** number should follow the standard [international format](http://en.wikipedia.org/wiki/National_conventions_for_writing_phone_numbers) ([http://en.wikipedia.org/wiki/National\\_conventions\\_for\\_writing\\_phone\\_numbers](http://en.wikipedia.org/wiki/National_conventions_for_writing_phone_numbers)) starting with '+' and including country and area code.

**Email** address is used for all account related correspondence with eagle.io. You should use your company admin email rather than your personal email address. eg. *admin@company.com*

### Contact Information

<b>Address</b>	<input type="text" value="Suite A"/>
	<input type="text" value="364A Harbour Drive"/>
<b>City</b>	<input type="text" value="Coffs Harbour"/>
<b>State/Province</b>	<input type="text" value="NSW"/>
<b>Zip/Postal code</b>	<input type="text" value="2450"/>
<b>Country</b>	<input type="text" value="Australia"/>
<b>Phone</b>	<input type="text" value="eg. +61400000001"/>
<b>Email</b>	<input type="text" value="info@eagle.io"/>

### Support Information

<b>Phone</b>	<input type="text" value="eg. +61400000001"/>
<b>Email</b>	<input type="text" value="support@eagle.io"/>

Public enquires and support requests will be sent to this address

## Quality Codes

The Quality Codes section displays a list of the historic *Qualities* (page 577) for this Account. Quality Code settings apply to all Workspaces in the account. Use the *Add* button to create custom quality codes.

Source and Quality Codes must be whole numbers between 0 - 65535.

**Exclude quality**  
  
**Unspecified quality**  
  
**Unknown quality**

?

?

?

Name	Source Quality	Quality Code	Type	Color	
Uncertain - Subnormal	<input style="width: 80%;" type="text" value="149"/>	<input style="width: 80%;" type="text" value="149"/>	Uncertain		
Bad - No Data	<input style="width: 80%;" type="text" value="155"/>	<input style="width: 80%;" type="text" value="155"/>	Bad		
Bad - No Bounds	<input style="width: 80%;" type="text" value="156"/>	<input style="width: 80%;" type="text" value="156"/>	Bad		
Bad - Data Unavailable	<input style="width: 80%;" type="text" value="158"/>	<input style="width: 80%;" type="text" value="158"/>	Bad		
Good - No Data	<input style="width: 80%;" type="text" value="165"/>	<input style="width: 80%;" type="text" value="165"/>	Good		
<input style="width: 80%;" type="text" value="Good - Verified"/>	<input style="width: 80%;" type="text" value="192"/>	<input style="width: 80%;" type="text" value="192"/>	Good <div>⬆</div> <div>⬇</div> <div>⬆</div>		

**Add New**

Good 

⬆

⬇

⬆

Add

**Exclude quality** You can optionally choose to exclude *Bad* and *Uncertain* quality coded data from all historic data requests. Excluded data points will not be visible on Charts, Tabular data and Historic API requests.

**Unspecified quality** Quality to assume when no quality code has been specified.

**Unknown quality** Quality to assume when the quality code specified does not exist.

**Source quality** Quality code(s) to match on acquired data. Separate multiple quality codes with a comma.

**Quality code** Quality code to apply to the data for display and export. You can use the same Code as the Source or specify a custom code. Must be numeric.

**Color** Select a color to be assigned for this quality for display with current and historic data.

Refer to the [Quality reference](#) (page 577) for further details.

---

**Note:** System qualities can not be removed, but can be updated with new Quality Codes.

---

## 2.12.2 Security



Security allows you to configure Administrators of the account and User roles.

### Administrators

Administrators have full access to your account including unrestricted access to all Workspaces and most configuration settings for the account. There is no need to explicitly *Share a Workspace* (page 54) with any Administrators of the same account. If you would like a user to have restricted access to Workspaces, then *do not* add them as Administrators. Instead use *Workspace Sharing* (page 54) to add them as a User of a Workspace with an assigned role.

Add a new administrators by entering their email addresses and click *Add*. Separate multiple email addresses with a comma. Remove administrators by clicking the trash icons. Click *Save* to confirm.

Add new user
Add

Users with Administrator access		
John Citizen Last login an hour ago	john@company.com	Account owner
user@company.com Never logged in [resend invite]	user@company.com	


**Note:** You should only add trusted users as Administrators to keep your account secure.

### Groups

Groups are used to organize your users and provide a convenient way to manage *Workspace access* (page 54) and *notifications* (page 179). To use groups, simply click *Add group*, assign a unique name and enter the email address of user(s) that you want to include in the group. New users will receive a welcome email with links to setup their user profile and login. A user can be assigned to more than one group.


After saving the newly created group, you can proceed to *share workspace access* (page 54) and *manage notifications* (page 180).

You can now add/remove users to the group and they will automatically receive/lose Workspace access and notifications associated with the group.

 **Engineers**

249 users

Edit

 **Inspection Team**

3 users

Hide

**Name**

Inspection Team

?

**Users**

John Citizen

Last login 2 hours ago

john@company.com

Simon Kelly

Last login 3 days ago

s.kelly@company.com

sam@company.com

Save required

sam@company.com

Type email address of user

Add user

?

Add group

## Roles

Roles are assigned when [Sharing a Workspace](#) (page 54) to restrict user and group access to certain features. Create unlimited new roles, or change existing role permissions as needed. Removing a role that has been assigned will restrict associated users to *View only* access.

### Default Roles

Roles	Permissions			
Manage	<input checked="" type="checkbox"/> view <input checked="" type="checkbox"/> send messages <input checked="" type="checkbox"/> edit quality & annotations <input checked="" type="checkbox"/> subscribe notifications <input checked="" type="checkbox"/> security	<input checked="" type="checkbox"/> read attachments <input checked="" type="checkbox"/> acknowledge alarms <input checked="" type="checkbox"/> edit data <input checked="" type="checkbox"/> manage notifications	<input checked="" type="checkbox"/> export data <input checked="" type="checkbox"/> edit states & alarms <input checked="" type="checkbox"/> operate <input checked="" type="checkbox"/> configure	
Configure	<input checked="" type="checkbox"/> view <input checked="" type="checkbox"/> send messages <input checked="" type="checkbox"/> edit quality & annotations <input checked="" type="checkbox"/> subscribe notifications <input type="checkbox"/> security	<input checked="" type="checkbox"/> read attachments <input checked="" type="checkbox"/> acknowledge alarms <input checked="" type="checkbox"/> edit data <input checked="" type="checkbox"/> manage notifications	<input checked="" type="checkbox"/> export data <input checked="" type="checkbox"/> edit states & alarms <input checked="" type="checkbox"/> operate <input checked="" type="checkbox"/> configure	
Operate	<input checked="" type="checkbox"/> view <input checked="" type="checkbox"/> send messages <input checked="" type="checkbox"/> edit quality & annotations <input checked="" type="checkbox"/> subscribe notifications <input type="checkbox"/> security	<input checked="" type="checkbox"/> read attachments <input checked="" type="checkbox"/> acknowledge alarms <input type="checkbox"/> edit data <input type="checkbox"/> manage notifications	<input checked="" type="checkbox"/> export data <input checked="" type="checkbox"/> edit states & alarms <input checked="" type="checkbox"/> operate <input type="checkbox"/> configure	
View	<input checked="" type="checkbox"/> view <input type="checkbox"/> send messages <input type="checkbox"/> edit quality & annotations <input type="checkbox"/> subscribe notifications <input type="checkbox"/> security	<input type="checkbox"/> read attachments <input type="checkbox"/> acknowledge alarms <input type="checkbox"/> edit data <input type="checkbox"/> manage notifications	<input type="checkbox"/> export data <input type="checkbox"/> edit states & alarms <input type="checkbox"/> operate <input type="checkbox"/> configure	
Add New	<input checked="" type="checkbox"/> view <input type="checkbox"/> send messages <input type="checkbox"/> edit quality & annotations <input type="checkbox"/> subscribe notifications <input type="checkbox"/> security	<input type="checkbox"/> read attachments <input type="checkbox"/> acknowledge alarms <input type="checkbox"/> edit data <input type="checkbox"/> manage notifications	<input type="checkbox"/> export data <input type="checkbox"/> edit states & alarms <input type="checkbox"/> operate <input type="checkbox"/> configure	<input type="text" value="Label"/> <input type="button" value="Add"/>

### Permissions

**View** Minimum required permissions to log in and view content.

**Read attachments** Preview and Download attachments.

**Export data** Export historic data associated with parameters and locations.

**Send messages** Send messages to users of the Workspace.

**Acknowledge alarms** Acknowledge alarms with or without comment.

**Edit states & alarms** Configure Node states and alarms, or Clear alarms.

**Edit quality & annotations** Assign quality and annotations to timeseries records.

**Edit data** Modify or remove timeseries record values.

**Operate** User can Acquire and Control Datasources and Parameters.

**Subscribe notifications** Subscribe to Nodes and receive Email and SMS notifications.

**Manage notifications** Manage user notification subscriptions on behalf of other users.

**Configure** Create, Modify & Delete Nodes and associated configuration or Upload Attachments.

**Security** Workspace Sharing and assignment of User Roles.

---

**Note:** Security and Sharing changes take effect immediately on save.

---

## API keys

API keys are used to authenticate eagle.io HTTP API requests. The API keys section allows you to create, delete or modify the permissions for API keys associated with your user account. You can have multiple API keys active at once.

API resource requests are restricted to the assets within the associated account and are assigned an *access level* and *permission* to restrict the actions an API request can perform.



The screenshot shows a web interface for managing API keys. It features two main sections, each with a key icon and a title. The first section, 'Default', shows an API key with 'read only' access to all workspaces, created on 2016-10-31. The second section, 'Remote Logger', shows an API key with 'modify' access to 'limited' workspaces, with a 'Save required' status. Below this, there are form fields for 'API key', 'Label', 'Permission', 'Access level', and 'Workspaces'. The 'API key' field contains the value '08SWwWW2eDLegUIQHJJ9yKuTvVIANAEqXuUvkbME'. The 'Label' field contains 'Remote Logger'. The 'Permission' dropdown is set to 'Modify'. The 'Access level' dropdown is set to 'Limited workspaces'. The 'Workspaces' section has three checkboxes: 'Workspace 01' (checked), 'Workspace 02' (checked), and 'Workspace 03' (unchecked). At the bottom, there is an 'Add API key' button.

**Default**  
API key has **read only** access to **all** workspaces in the account.  
Created 2016-10-31 15:41:13

**Remote Logger**  
API key has **modify** access to **limited** workspaces in the account.  
Save required

**API key** 08SWwWW2eDLegUIQHJJ9yKuTvVIANAEqXuUvkbME

**Label** Remote Logger

**Permission** Modify

**Access level** Limited workspaces

**Workspaces**

- ☒ Workspace 01
- ☒ Workspace 02
- ☐ Workspace 03

Add API key

*Access level*

**Account** API requests will have access to all account resources including owner and account information and all Workspaces.

**Workspace** API requests will have access to all Workspace resources.

**Limited Workspaces** API requests will be limited to a pre-selected set of Workspaces.

*Permission*

**Read only** The API key can only be used to perform read-only (GET) requests on the resources specified by *Access level*.

**Modify** The API key can perform all requests on the resources specified by *Access level*.

---

**Note:** Your API keys carry many privileges, so be sure to keep them secret!

---

## Single Sign-On

Single Sign-On (SSO) enables users from nominated domains to login via an enterprise authentication service.

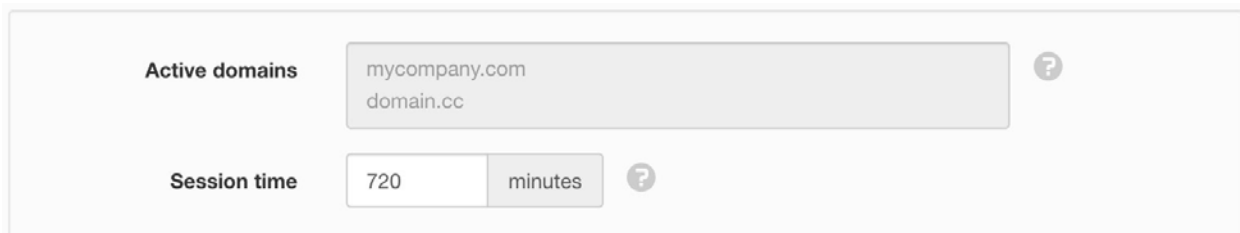
We support a range of SSO identity providers including:

- Auth0 SAML
- AzureAD
- CyberArk SAML
- Generic SAML
- Google SAML
- JumpCloud SAML
- Microsoft AD FS
- Okta
- OneLogin
- OpenID Connect
- PingFederate
- PingOne
- Salesforce
- VMware Workspace One

---

**Note:** This is an enterprise level feature. Please contact support for more information.

---



The screenshot shows a configuration interface for Single Sign-On. It has two main sections: 'Active domains' and 'Session time'. The 'Active domains' section has a text input field containing 'mycompany.com' and 'domain.cc', with a question mark icon to its right. The 'Session time' section has a numeric input field containing '720', a dropdown menu currently set to 'minutes', and a question mark icon to its right.

**Active domains** Users that login with an email address from an active domain are automatically redirected to a pre-configured Single Sign-On identify provider.

**Session time** Set the number of minutes after login before the user is automatically re-authenticated against the identity provider. Enter a value between 5 and 50000 minutes. The default is 720 minutes.

### 2.12.3 Workspaces



The workspaces section displays a list of all workspaces in the account. You can add, remove or configure workspaces.

The usage overview shows a current snapshot of usage for each workspace in the account. The user count does not include the account owner, administrators or account managers.

Add new workspace

Workspace name

Add

Workspaces	Usage overview								Created
Demonstration	2	1	1	2	12	2	0	1	2014-01-12 13:49:00
California	1	1	1	2	6	0	1	0	2016-02-23 15:48:46
Europe	3	1	3	4	24	2	0	0	2015-03-11 10:05:33

**Note:** Workspaces can also be created, removed or configured directly from the Workspaces Tree.

## 2.12.4 Notifications



Notifications section is used to configure notification preferences for the account.

### Notification Limits

Email notifications are free and unlimited.

All account plans are bundled with a specific number of included monthly SMS. Additional SMS Packs can be purchased in quantities of 100 / month.

The *Send SMS via email* option will send SMS notifications for free to phone numbers in the USA using the recipients' service provider email to SMS transport where possible. This option is only visible on primary accounts and is automatically applied to all managed accounts. The phone number field in [user preferences](#) (page 166) indicates if an SMS email address is in-use for your user profile (after sending an initial SMS) and allows you to reset/retry if you change service providers.

The screenshot shows a configuration panel for SMS. It includes two rows: 'Included monthly SMS' with a value of 250, and 'Additional SMS pack' with a value of 100 and a price of \$10.00 / month. Below these is a button labeled 'Change SMS pack'. At the bottom, there is a checkbox labeled 'Send SMS via email where possible (USA only)' which is checked. A help icon (?) is visible on the right side of the panel.

You can optionally set a monthly *SMS limit* or disable SMS completely. The SMS limit is applied to assets configured directly within the account only. Managed accounts have an SMS Limit setting independent of the primary account.

The screenshot shows a configuration panel for the SMS limit. It has a label 'SMS limit' followed by a dropdown menu. The dropdown menu is currently set to 'SMS Disabled. No notification SMS will be sent'. A help icon (?) is visible to the right of the dropdown.

---

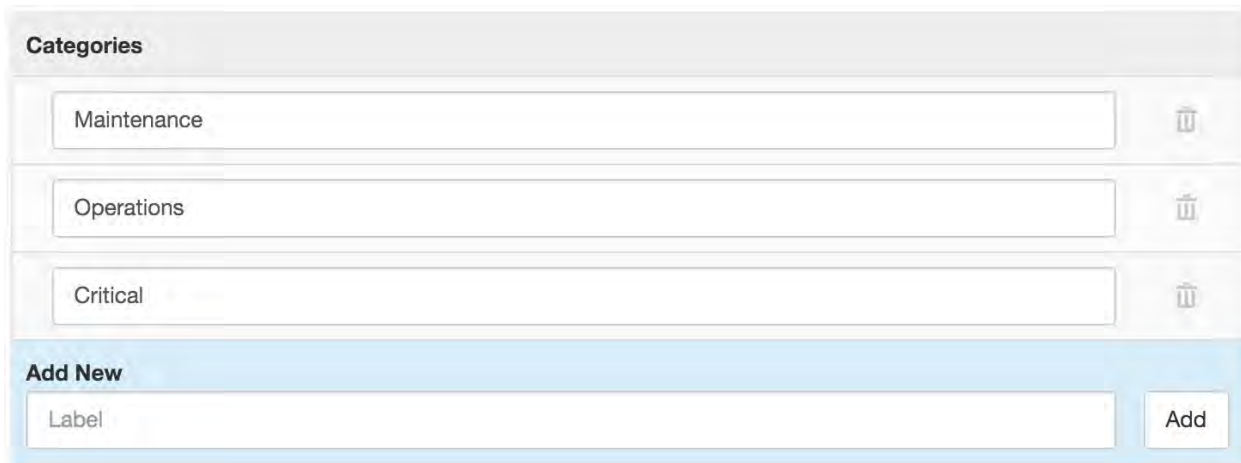
**Note:** The account owner will receive an email from the system when the total number of SMS sent within the current billing period gets close to the account limit. Another email will be sent when the limit is reached and no further notification SMS will be sent until the next billing period.

---

## Notification Categories

Notification Categories provide a mechanism for users to filter the notifications they receive from a node they have subscribed to. All Nodes, States and/or Alarms that provide a Notification option require a Category to be specified. Users must individually select the categories in each Workspace they would like to receive notifications for. See [Notification Preferences](#) (page 168) for full details.

Add, rename and remove categories as needed for your organization.



The screenshot shows a web interface for managing notification categories. At the top, there is a header bar labeled "Categories". Below this, there is a list of three existing categories: "Maintenance", "Operations", and "Critical". Each category is displayed in a light gray box with a trash icon to its right, indicating it can be deleted. Below the list, there is a section titled "Add New" with a light blue background. This section contains a text input field labeled "Label" and an "Add" button to its right.

**Note:** Removing categories will also remove associated user notification preferences. There should always be at least one notification category configured.


## 2.12.5 Billing



Please ensure your billing details are always up to date to prevent your account being suspended.


### Payment Information

Credit Card details must be kept up to date. You will be notified about any failed transactions. Please contact [support@eagle.io](mailto:support@eagle.io) if you have any issues updating your billing information.

Cardholder name	<input type="text" value="John Citizen"/>		
Card number	<input type="text" value="XXXXXXXXXXXX1111"/>		
Expiration date	<input type="text" value="01"/>	<input type="text" value="2017"/>	
CVV	<input type="text" value="XXX"/>		

### Billing History

The Billing History section displays a list of all previous account bills available for download.

Billing Period	Amount	
20th November 2013 – 20th December 2013	USD \$1115.24	

### 2.12.6 Customization



Create a personalised appearance for your company. Upload your own company logos, graphics and text which are displayed in the user interface and emails.

#### Product

#### Domains

You can specify a subdomain for free. If you would like to use your own custom domain name please contact your account manager or [support@eagle.io](mailto:support@eagle.io). Subdomain changes take effect immediately and logged in users will be prompted to navigate to the new domain.


Minimum length of 5 characters is required. The subdomain will be validated on save.

Domain	<input type="text" value="my-company"/>	.eagle.io
--------	-----------------------------------------	-----------


#### Company Logo

Company logos can be uploaded to customize the appearance of the site when accessed via the account domain. Logos are also used in emails sent to users.


Click on the logo images to replace them with your own. For best results use images that conform to the recommended dimensions.

**Icon**

The Icon image is displayed in the browser tab and used as the app icon when bookmarking the website on most devices. The icon is also displayed in the application navigation bar on small screen devices. Image dimensions: **144 x 144px**

**Inline**

The Inline image is displayed in the application navigation bar on large screen devices and content sections with a dark background. Max image dimensions: **400 x 40px**

**Block**

The Block image is the main company logo and is used on the Public site, login and emails. Max image dimensions: **600 x 100px**

Reset all logos to defaults


## Public Content

### Front Page

You can customize your landing page by changing the banner image and description.



**Company banner**



The banner image is displayed on the front page. Recommended image dimensions: **1140 x 290px**

**Banner heading**

Remote Data, Connected

**Banner text**

Connect, configure and control your monitoring fleet from the cloud.

Reset content to defaults

## Contact

Tick or untick the *Contact details* checkbox to display or hide your account contact details on the public *Contact Us* page.

**Contact details**

☒ Display details from this accounts Contact Information section on the public **Contact Us** page.

## 2.12.7 Integrations



Integrations provide extended functionality by connecting to apps from third party providers.

### S3 Backup

The S3 Backup integration enables data files received by eagle.io to be backed up to a specific S3 bucket in your own Amazon Web Services account.

---

**Note:** This feature is a paid addition to Standard accounts, and is included with Premium accounts. Please contact [sales@eagle.io](mailto:sales@eagle.io) to learn more about premium features.

---

To use this integration, first create a new S3 bucket in your AWS account to be used for the specific purpose of eagle.io backups. [Create an IAM user](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html) ([https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_users\\_create.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html)), and allocate an [API access key](https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html#access-keys-and-secret-access-keys) (<https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html#access-keys-and-secret-access-keys>) to this user. The access key will consist of two parts: an access key ID, and a secret access key. Finally, [assign write permissions](https://docs.aws.amazon.com/AmazonS3/latest/user-guide/set-bucket-permissions.html) (<https://docs.aws.amazon.com/AmazonS3/latest/user-guide/set-bucket-permissions.html>) for this user to the bucket.

With this information, you can then enable the integration and enter the three required settings:

1. Bucket name
2. Access key ID
3. Secret access key

You can also enter an optional file name prefix.

The screenshot shows a form titled "S3 Backup" with the subtitle "Automatically backup your data to Amazon Web Services S3". In the top right corner of the form is a blue button with a plus icon and the text "Integrate". Below the title, there are four input fields: "Bucket", "Access key", "Secret key", and "Prefix". The "Prefix" field has a small text label below it that says "Optional file name prefix".

## 2.12.8 Managed accounts



Managed accounts are new accounts you have created for end-clients or your own company departments to segregate assets and users but retain full management functionality and centralized billing.

Administrators of your account automatically inherit administrator access to all managed accounts.

---

**Note:** Trial accounts and users of managed accounts do not have access to the *managed accounts* section.

---

### Trials

Create managed trials to provide clients with a 30 day free evaluation of the service at no charge to you.

Managed trials are displayed in the list with their status initially set to *ACTIVE* and become *DISABLED* on expiry. Use the dropdown menu to *Switch to account*, *Request extension*, *Upgrade to paid* or *Close account*.

Disabled trial accounts will be automatically removed 3 months after expiry.

### Usage Limits

During the trial, users can create unlimited Sources, but the account is restricted to 10 free SMS.

Account	Status	✉ SMS	➔ Source	Created date	Expiry	Owner
Clear Water Tech	ACTIVE	10 ▼	No Limit ▼	10 Nov 2021	10 Dec 2021	info@clear-water-tech.com Never logged in [resend invite] ▼

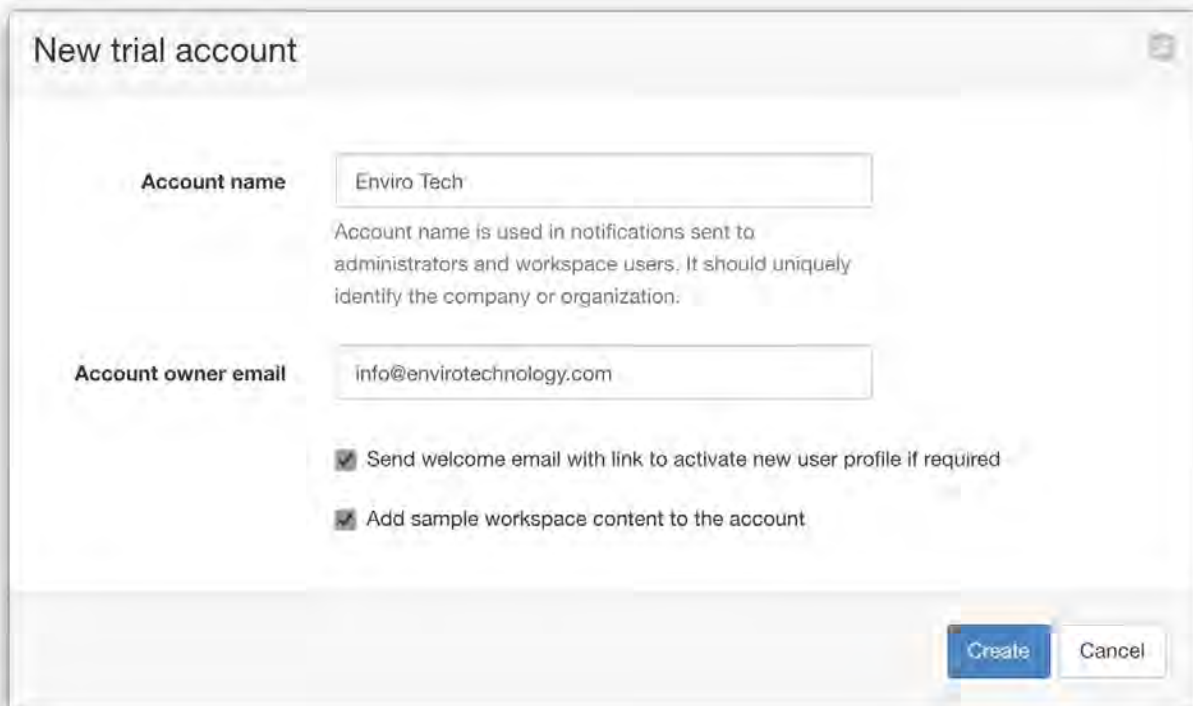
New trial

---

**Note:** When a trial account is disabled, users will not have access to workspaces, historic data or events until the account has been *extended* or *upgraded to paid*.

---

Click the *New trial* button to create a new trial account. Specify the *account name* and *account owner email* address.



**New trial account**

**Account name**

Account name is used in notifications sent to administrators and workspace users. It should uniquely identify the company or organization.

**Account owner email**

☒ Send welcome email with link to activate new user profile if required

☒ Add sample workspace content to the account

**Create** **Cancel**

You can optionally send a welcome email to the account owner. If the user has never logged-in to the system, the welcome email will contain a link to activate their user profile. You can skip the welcome email if you want to customize the account first, then use the *resend invite* option from the account list when ready (only available if the user has never logged-in).

Sample workspace content can also be added when the account is created. You should ensure you remove any demonstration datasources prior to upgrading a trial to a paid account to prevent being charged usage fees for this content.

---

**Note:** The *account owner email* address can only be assigned as the account owner of one account.

---

## Accounts

Create managed accounts to provide your clients with their own rebranded account. Source and SMS usage will contribute to your overall account usage while the account is active.

Managed accounts are displayed in the list with their status initially set to *ACTIVE*. Use the dropdown menu to *Switch to account*, *Disable*, *Enable* or *Close account*. You can disable an account and re-enable it at any time. You will not be charged usage fees for the duration the account is disabled but new data will not be collected until the account has been re-enabled.

## Usage Limits

You can optionally configure SMS and Source usage limits for individual accounts.

**SMS** limit restricts the number of SMS that can be sent from the account per monthly billing period.

**Source** limit restricts the number of Sources that can be created within the account. The usage per source may vary depending on type, size and account plan.

Applying a limit that is below *current usage* (page 265) will just prevent new SMS being sent or new Sources from being created. Limits are always capped by the primary account plan capacity.

Account	Status	✉ SMS	➔ Source	Created date	Owner	
Weather Org	DISABLED	No Limit ▾	No Limit ▾	13 Sep 2019	info@weather.org <small>Last login 3 months ago</small>	▾
Enviro Data Service	ACTIVE	0 ▾	10 ▾	2 Aug 2021	data@enviroservice.com <small>Never logged in (reset invite)</small>	▾
New account						Save Cancel

Click the *New account* button to create a new paid account. All account options are the same as the *New trial* dialog above.

---

**Note:** When an account is disabled, users will not have access to workspaces, historic data or events until the account has been *enabled* again.

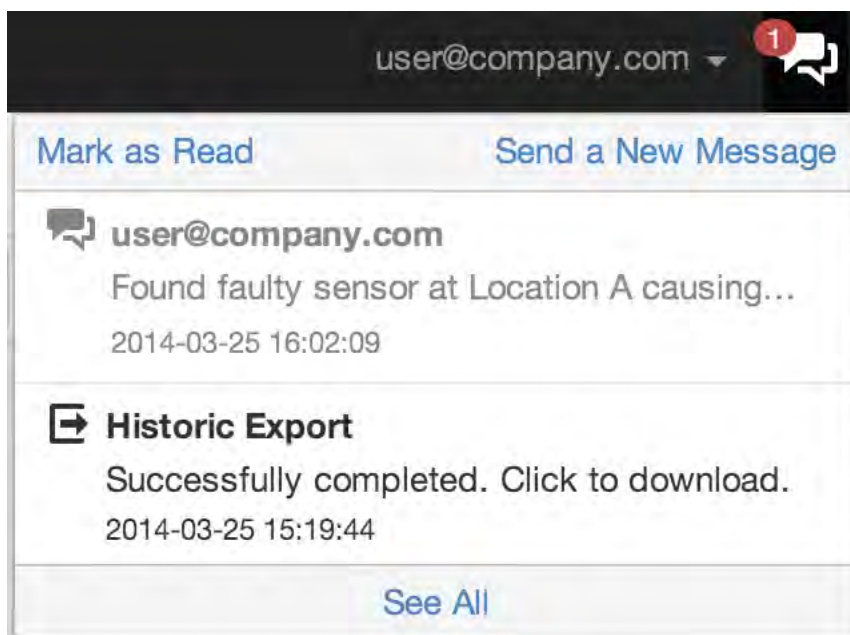
---

## 2.13 Messages



Messages provides access to items for download, general communication to users of a Workspace and important system announcements.

The Messages icon indicates the total number of *Unread* messages. Clicking the icon will reveal the Message Menu (Desktop layout only) which displays a preview of the most recent messages. Scroll down the menu to reveal older messages. Clicking on an Export message will immediately download the associated file.



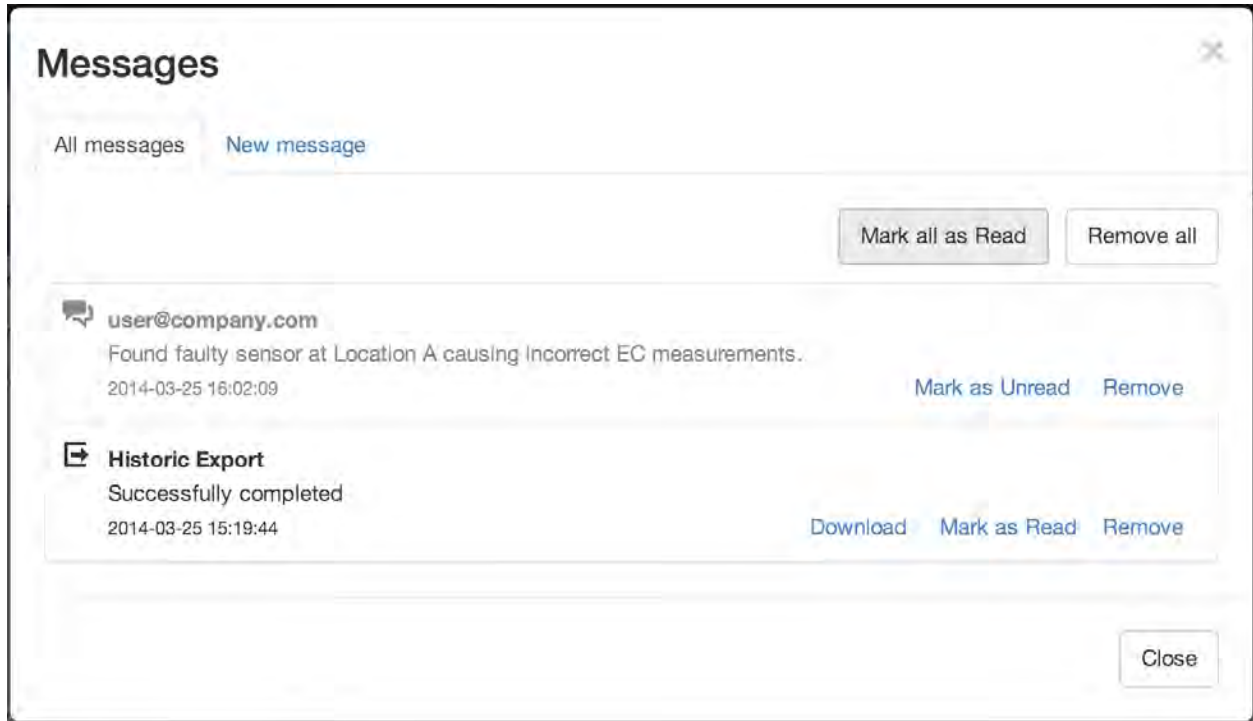
---

**Note:** Export messages are automatically removed from the system after 12 hours.

---

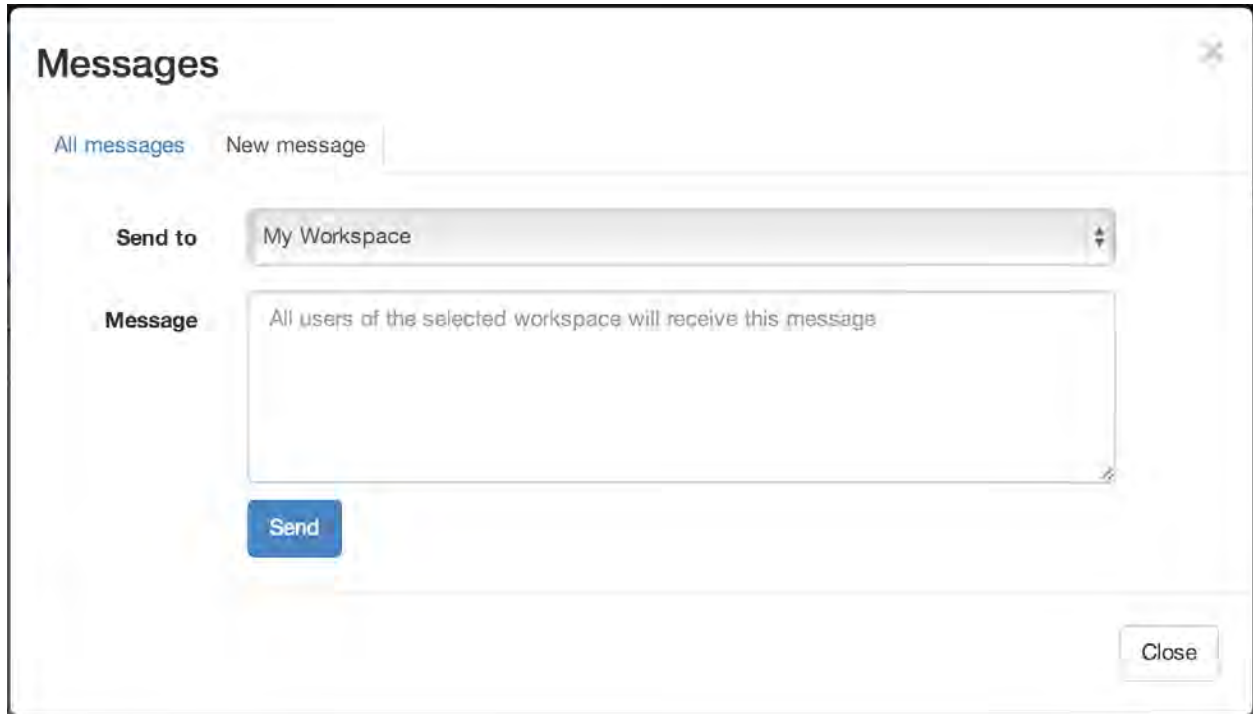
### 2.13.1 All messages

The Messages dialog displays the full message content and provides management functions for marking messages as read/unread, removing messages and downloading historic exports.



### 2.13.2 New message

Send a new message to all users of a Workspace including the associated account Administrators and Owner. You will also receive a copy of the message. Select the Workspace from the drop down, enter your message and click Send.



**Messages**

All messages | New message

**Send to** My Workspace

**Message** All users of the selected workspace will receive this message

Send

Close

---

**Note:** You must have *Send Messages* permission for the Workspace in order to send messages to its users.

---



## 2.14 Device Configuration

The following topics will assist when configuring your device for [eagle.io](https://eagle.io) (<https://eagle.io>)

### 2.14.1 Campbell Scientific

[Eagle.io](https://eagle.io) (<https://eagle.io>) provides full support for [Campbell Scientific](http://www.campbellsci.com/) (<http://www.campbellsci.com/>) PakBus data loggers including the **CR200**, **CR300**, **CR800**, **CR1000**, **CR3000**, **CR9000**, **CRVW** and **CR6** series.



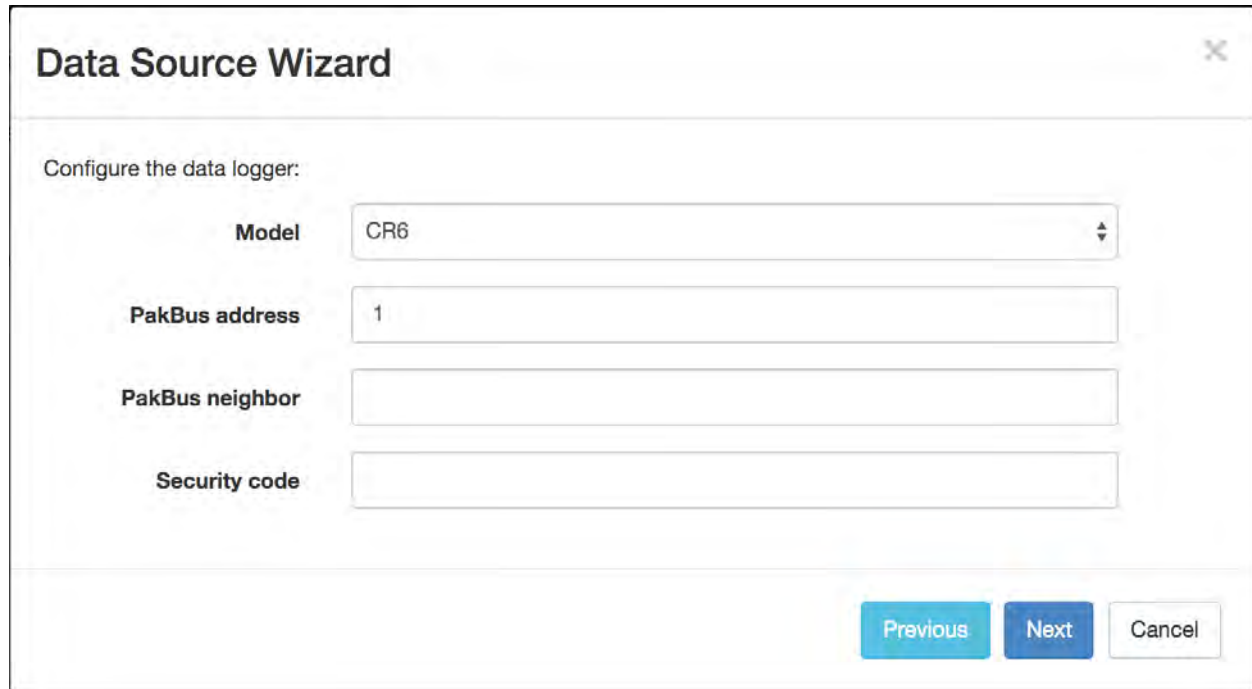
This section provides instructions for connecting your PakBus data logger to [eagle.io](https://eagle.io). Refer to *Operate and Control* (page 182) for advanced usage instructions.

### Source configuration

1. Create a *Data Source* (page 134) and select *Data logger* as the Source Type. Choose *Campbell Scientific [PakBus]* from the list and click Next.

 A screenshot of the 'Data Source Wizard' dialog box. The title bar says 'Data Source Wizard' with a close button (X) on the right. Below the title bar, the text 'Select source type:' is followed by two radio buttons: 'File' (unselected) and 'Data logger' (selected). To the right of the radio buttons is a list box containing two items: 'Campbell Scientific [PakBus]' (which is highlighted) and 'WaterLog'. At the bottom right of the dialog are two buttons: 'Next' (in blue) and 'Cancel' (in white with a grey border).

2. Select the data logger *Model* and *PakBus address* (default is 1). Optionally specify a *PakBus neighbor* and logger *Security code*. Click Next.



The image shows a 'Data Source Wizard' dialog box with a close button (X) in the top right corner. The main heading is 'Data Source Wizard'. Below it, the instruction 'Configure the data logger:' is displayed. The form contains four fields: 'Model' with a dropdown menu showing 'CR6', 'PakBus address' with a text input containing '1', 'PakBus neighbor' with an empty text input, and 'Security code' with an empty text input. At the bottom right, there are three buttons: 'Previous' (light blue), 'Next' (dark blue), and 'Cancel' (light gray).

3. Select a *Transport type* to use for [connecting to your logger](#) (page 136). This example uses the *TCP Client* Transport type to connect directly to a logger accessible via the Internet. Select the *Collection interval* and *Timezone* of the data logger. Refer to [Data Source configuration](#) (page 134) for a full list of available options.

## Data Source Wizard

Configure the connection:

### Connection

**Transport type** TCP Client

**Internet address** mydevice.dyndns.org

**TCP port** 4002

### Collection

**Collection interval** 1 hour

**Initial collection** Collect all data

☐ Assign as Location coordinate source

### Time

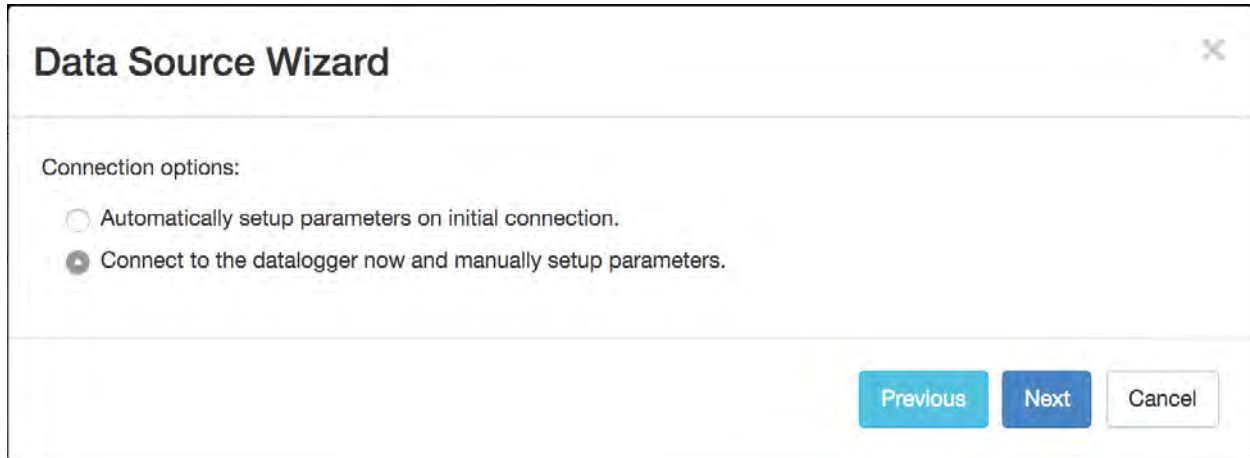
**Timezone** (GMT+10:00) Eastern Time - Melbourne, Sydney

☒ Automatically adjust clock for Daylight Savings Time

☒ Automatically update datalogger with correct time

[Previous](#) [Next](#) [Cancel](#)

4. You can choose to *Connect to the data logger now and manually setup parameters* if the logger is online. Alternatively choose the *Automatically setup parameters on initial connection* option and click Next to complete the setup.



The image shows a 'Data Source Wizard' dialog box. It has a title bar with the text 'Data Source Wizard' and a close button (X) in the top right corner. Below the title bar, there is a section labeled 'Connection options:'. This section contains two radio button options: 'Automatically setup parameters on initial connection.' and 'Connect to the datalogger now and manually setup parameters.'. The second option is selected, indicated by a filled radio button. At the bottom right of the dialog box, there are three buttons: 'Previous' (light blue), 'Next' (dark blue), and 'Cancel' (white with a grey border).

5. Table definitions are retrieved from the logger upon successful connection. Select the series from each table that you would like to create as parameters. You can create [Control](#) (page 182) Parameters from any Public table variables. Specify the parameter names (and optionally units) and click Finish to complete the setup.

## Data Source Wizard

Configure series to be acquired

Location coordinates

⊗ None

ALERT

Chk\_Info

Main

<input type="checkbox"/> Series	Parameter assignment			Units
<input checked="" type="checkbox"/> WaterLevel	New	Existing	Water Level	m
<input checked="" type="checkbox"/> Batt_Volt	New	Existing	Battery	VDC
<input type="checkbox"/> Total_Rain	New	Existing	Main - Total Rain	

Public

Rain

Status

SvcMode

Previous

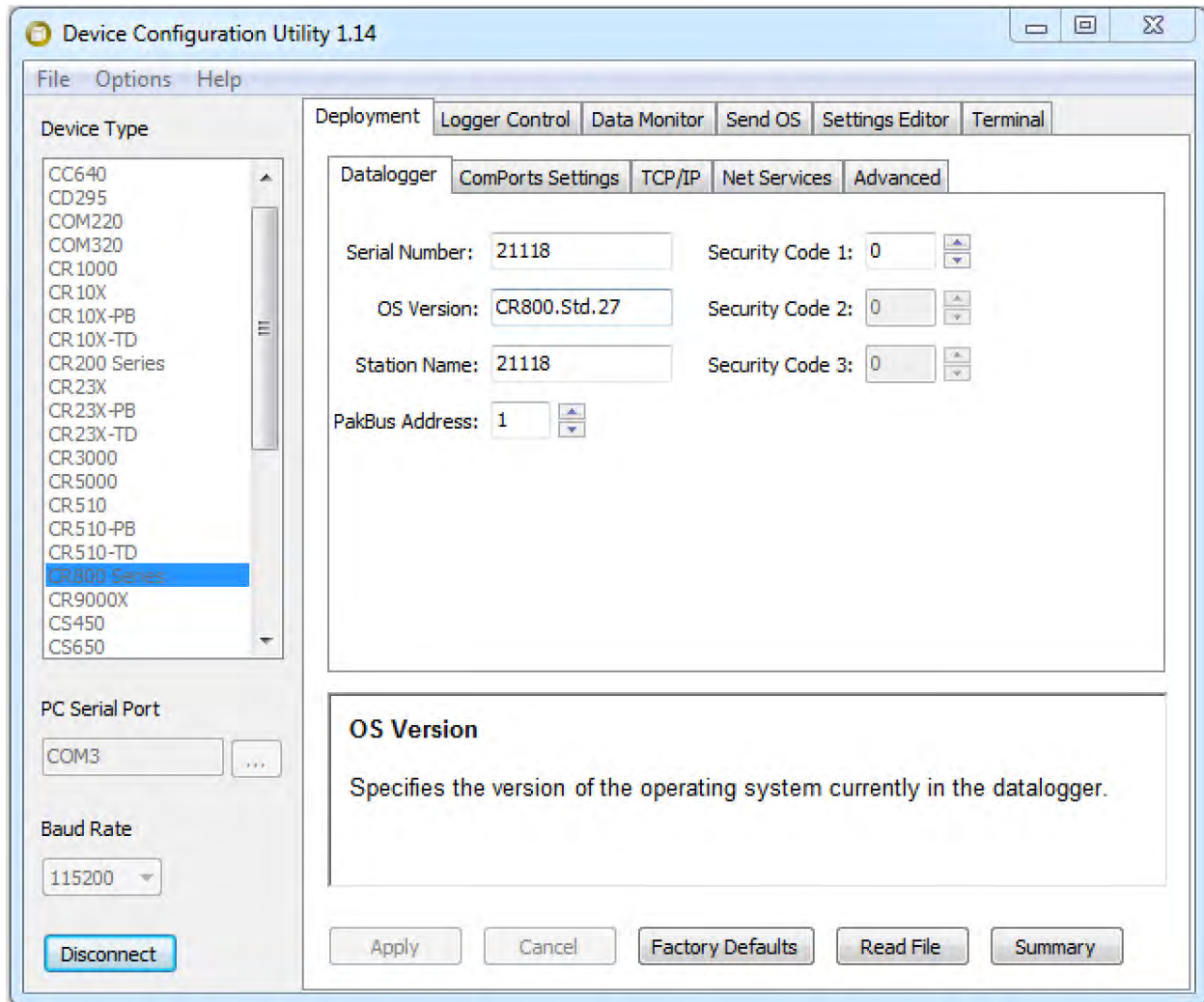
Finish

Cancel

**Note:** You can reconfigure the series Parameter assignment from the [Data Source](#) (page 134) properties dialog after the setup is complete.

## Device settings

Use the Campbell Scientific Device Configuration Utility to configure your data logger for communication with eagle.io. Refer to your [Campbell Scientific](http://www.campbellsci.com/) (<http://www.campbellsci.com/>) PakBus logger manual for connection instructions.



The following Server settings should be assigned as a minimum:

**PakBus Address** Use the PakBus address you assigned during Data Source creation. Default is 1.

**Security Code** Use the Security code you assigned during Data Source creation or use 0 for none.

You will also need to modify the TCP/IP settings as required based on your connection method.

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history

and diagnosing configuration issues.

### 2.14.2 CellVisor

The [E State Automation](http://www.estateautomation.com.au) (<http://www.estateautomation.com.au>) **CellVisor** range of data loggers can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.

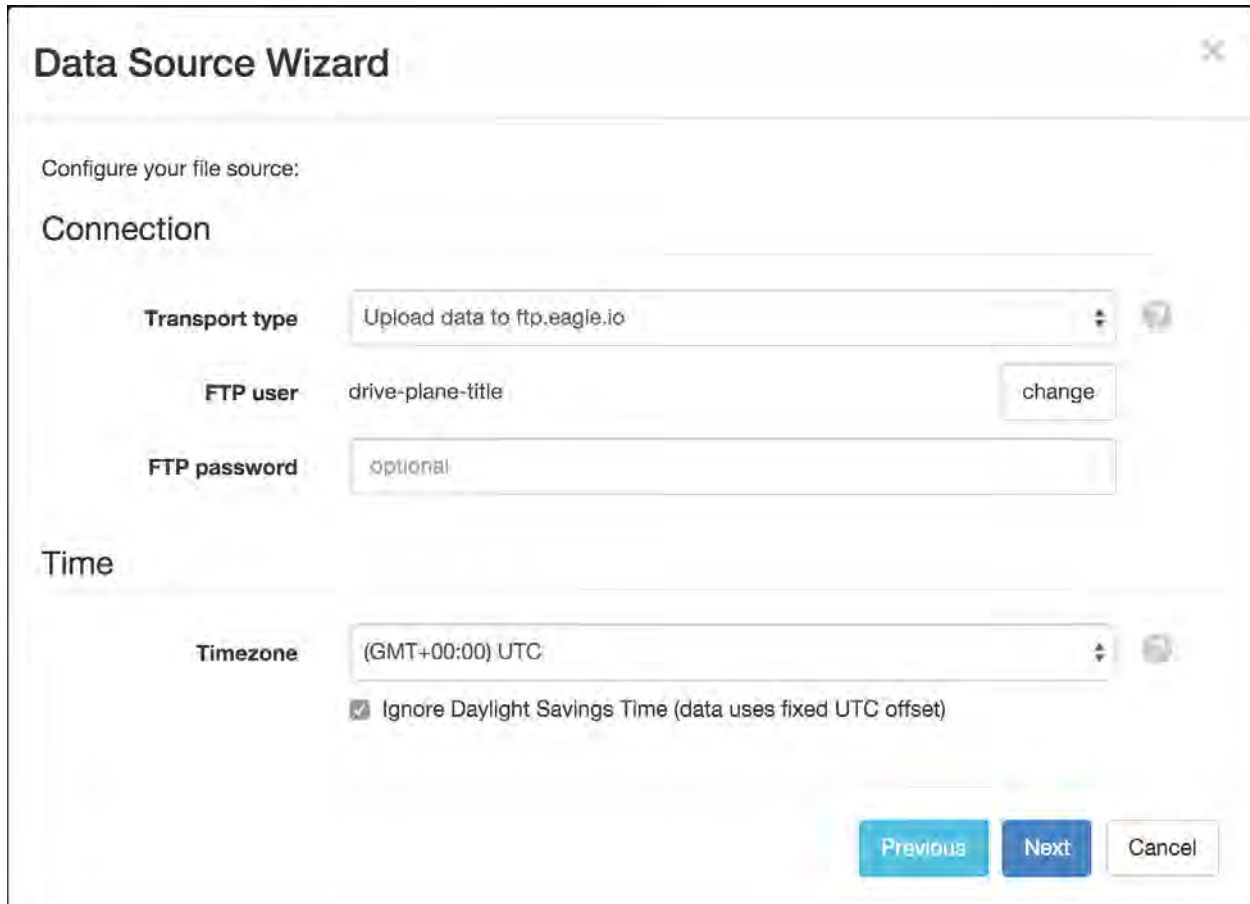


#### Source configuration

1. Create a [Data Source](#) (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.

2. Select *Upload data to ftp.eagle.io* as the transport type. A unique *Ftp user* account will be automatically generated per Data Source. You can optionally specify a password to be used by the logger for

additional security. Select *UTC* as the *Timezone* and click Next.



The image shows a 'Data Source Wizard' window with a close button in the top right. The main heading is 'Data Source Wizard'. Below it, the instruction 'Configure your file source:' is followed by the 'Connection' section. This section contains three fields: 'Transport type' with a dropdown menu showing 'Upload data to ftp.eagle.io', 'FTP user' with a text field containing 'drive-plane-title' and a 'change' button, and 'FTP password' with a text field containing 'optional'. Below the 'Connection' section is the 'Time' section, which includes a 'Timezone' dropdown menu showing '(GMT+00:00) UTC' and a checked checkbox labeled 'Ignore Daylight Savings Time (data uses fixed UTC offset)'. At the bottom right of the window are three buttons: 'Previous' (blue), 'Next' (dark blue), and 'Cancel' (white).

3. Specify *csv* as the *File name match*. This will match all incoming files for this Source containing 'csv' in the filename. To continue the setup you will need a sample copy of the logger's output data file(s).



**Add files** [Close]

**File name match**   [Help]

☐ Match using regular expression

**Sample file**   [Help]

☐ Skip or Reset sample file and retrieve on next acquisition

**Note:** If you do not have a sample available, click the *Skip* checkbox and click finish on the next screen. A sample will be saved automatically on the initial upload from the data logger and you will need to open the [Data Source](#) (page 134) properties dialog and [configure the series](#) (page 119) to create parameters.

4. The Parser configuration screen allows the user to describe the structure and format of the data file so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details.

**Parser configuration** csv

Column delimiter: , Text qualifier: " Labels row: 1 Text encoding: UTF-8 (default)

#	SiteName	IMEI	UTCDate	UTCTime	TimeZone
1	SiteName	IMEI	UTCDate	UTCTime	TimeZone
2	EIO_Test	355310036795957	2014-11-19	03:48:04	11.0
3	EIO_Test	355310036795957	2014-11-19	03:51:04	11.0
4	EIO_Test	355310036795957	2014-11-19	03:54:04	11.0
5	EIO_Test	355310036795957	2014-11-19	03:57:04	11.0
6	EIO_Test	355310036795957	2014-11-19	04:00:04	11.0

Get help with parser configuration

Apply Cancel

If the file contains a header row, set the *Labels row* control to the corresponding row number which will assign default labels for parameter creation.

Ensure the data in each column matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*.

You will need to use the drop-down menu above the columns to assign the *Record Time* data type to the first column containing part of the record timestamp. Click the configure button to change the *Format* field to use the *time format* (page 583) **YYYY-MM-DD**.

The record timestamp spans multiple columns by default. Add any additional columns containing part of the timestamp and specify the format. By default you will need to add the *UTCTime* column to the list of Joins and change the *Format* field to use the *time format* (page 583) **HH:mm:ss**. Click Apply and Next.

Parser configuration csv

Column delimiter ,
Text qualifier "
Labels row 1
Text encoding UTF-8 (default)

UTCDate + UTCTime

#
Record Time

1	UTCDate UTCTime
2	2014-11-19 03:48:04
3	2014-11-19 03:51:04
4	2014-11-19 03:54:04
5	2014-11-19 03:57:04
6	2014-11-19 04:00:04

UTCDate Joins and Parser Configuration

Joins	Field	Format
UTCDate	Time	YYYY-MM-DD
UTCTime	Time	HH:mm:ss

Add New

UTCDate
Quality

Add

Close

Get help with parser configuration

Apply
Cancel

- Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.

## Data Source Wizard

Configure series to be acquired

Location coordinates

csv

Series	Parameter assignment		Units
<input type="checkbox"/> IMEI	New	Existing	IMEI
<input type="checkbox"/> TimeZone	New	Existing	TimeZone
<input checked="" type="checkbox"/> Dir	New	Existing	Wind direction
<input checked="" type="checkbox"/> WSpd	New	Existing	Wind speed
<input checked="" type="checkbox"/> Baro	New	Existing	Barometric pressure
<input checked="" type="checkbox"/> RH	New	Existing	Relative Humidity
<input checked="" type="checkbox"/> Temp	New	Existing	Temperature

Previous

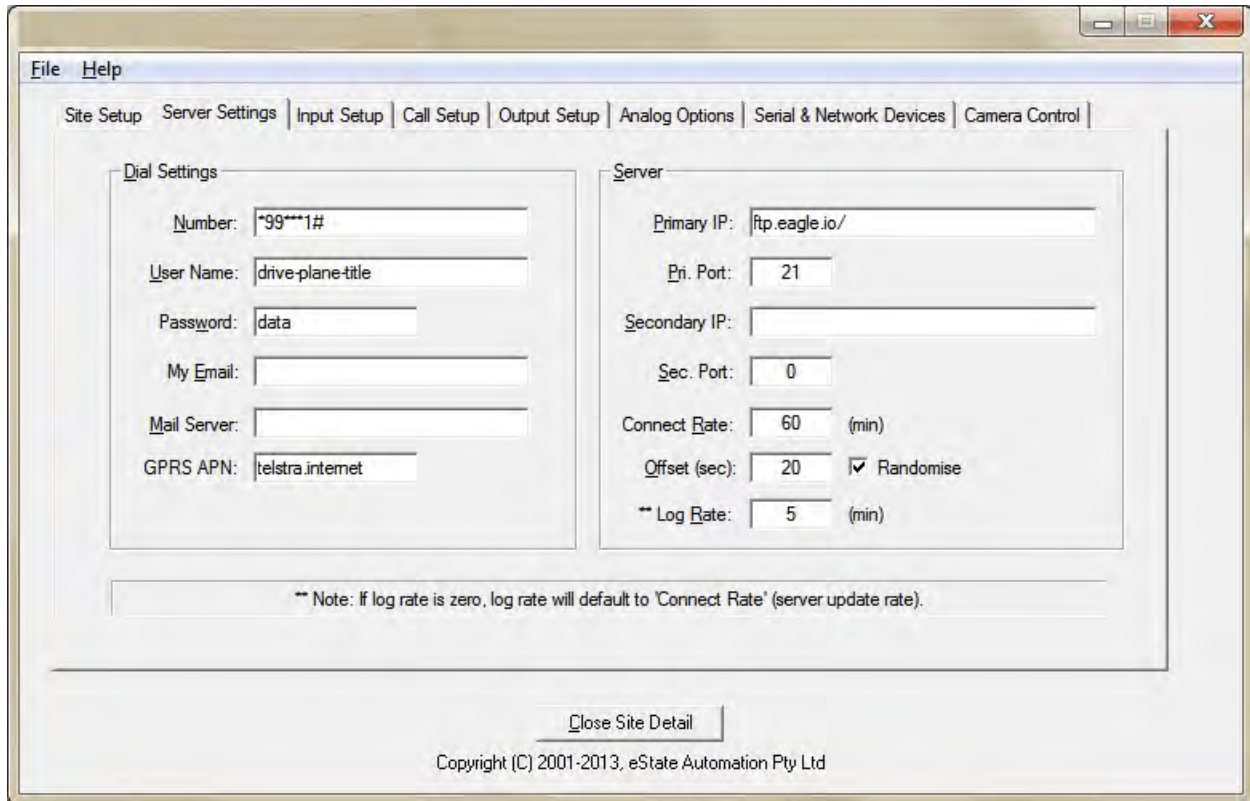
Finish

Cancel

**Note:** You can reconfigure the file name match/sample file, parser configuration and parameter assignment from the [Data Source](#) (page 114) properties dialog after creation.

## Device settings

Use the CellVisor software to configure the logger for communication with eagle.io. Refer to your CellVisor [user manual](http://estateautomation.com.au) (<http://estateautomation.com.au>) for connection instructions.



The screenshot shows a software configuration window with a menu bar (File, Help) and a tabbed interface. The 'Server Settings' tab is selected. It contains two main sections: 'Dial Settings' and 'Server'. The 'Dial Settings' section includes fields for Number (\*99\*\*\*1#), User Name (drive-plane-title), Password (data), My Email, Mail Server, and GPRS APN (telstra.internet). The 'Server' section includes fields for Primary IP (ftp.eagle.io/), Pri. Port (21), Secondary IP, Sec. Port (0), Connect Rate (60 min), Offset (sec) (20), Log Rate (5 min), and a checkbox for Randomise. A note at the bottom states: 'Note: If log rate is zero, log rate will default to 'Connect Rate' (server update rate).'. A 'Close Site Detail' button is at the bottom center, and the copyright notice 'Copyright (C) 2001-2013, eState Automation Pty Ltd' is at the bottom right.

The following Server settings should be assigned:

**Primary IP** ftp.eagle.io/

**Pri. Port** 21

**User Name** Use the unique *FTP user* generated during Data Source creation.

**Password** Use the password you assigned during Data Source creation (or leave blank for none).

---

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

---

### 2.14.3 Dataforth MAQ®20

The [Dataforth MAQ®20](https://www.dataforth.com/maq20.aspx) (<https://www.dataforth.com/maq20.aspx>) Industrial Data Acquisition and Control System can be connected with [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.



## Source configuration

1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.

## Data Source Wizard

1 Source type

2 Configure data logger

3 Configure source

4 Configure parameters

☐ Data logger

☒ File

☐ Processor

Delimited Text

JSON Time Series

Please refer to our [documentation](#) for help configuring your specific device.

Next

Cancel

2. Select *Download from FTP site* as the transport type. Configure the settings as defined in your FTP settings for the MAQ20 system. *FTP Host* must contain a valid public hostname or public IP that can be used to access the FTP server. Set Collection interval, Initial collection and Timezone to desired values, then click Next.

---

**Note:** If desired, please reference the [MA1040 MAQ20-COM Module Hardware User Manual](#) (<https://www.dataforth.com/FileDownload.ashx?docid=51>) for more information about FTP configuration.

---

## Data Source Wizard



1 Source type

2 **Configure source**

3 Add files

4 Configure parameters

## Connection

Transport type

Download from FTP site



FTP host

eg. ftp.company.com or 10.0.0.1



Port

21



Mode

FTP



Remote path

eg. /home



User

Password

Read mode

Read new records

☐ Delete files after acquire

## Collection

Collection interval

1 hour



Initial collection

Collect all data

☐ Assign as Location coordinate source

## Time

Timezone

(GMT+00:00) UTC

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous

Next

Cancel



3. If FTP settings are configured correctly, you will see a directory of files that exist in the system's SD card. Choose the filename that corresponds to the file you have configured for the SD card logger for the system, then click Apply.

Sample data

File name match

Reset

?

☐ Match using regular expression

SYSTEM~1

WSTUC\_\_5.CSV [3.81 KB]

Refresh

Apply

Cancel

4. The Parser configuration screen allows the user to describe the structure and format of the sample data file so it can be correctly interpreted by the system during subsequent data collection. Refer to [Text Parser Configuration](#) (page 123) for full details. First you will need to change the labels row to be 1.

Parser configuration WSTUC\_5.CSV

Column delimiter: Column series Labels row: None Units row: None Text qualifier: UTF-8 (default) Match: Smart Reset to defaults

Column 1	Column 2	Column 4	Column 5	Column 6	Column 7	
#	Record Time	Number	Number	Number	Number	Number
1	Timestamp	3000	3004	7000	9000	9106
2	11/14/18 21:15:43.000	2084	2184	2681	11	41
3	11/14/18 21:15:53.000	2084	2221	2953	18	41
4	11/14/18 21:16:03.000	2083	2277	2748	24	41
5	11/14/18 21:16:13.000	2083	2168	2800	22	41
6	11/14/18 21:16:23.000	2084	2257	3013	19	41
7	11/14/18 21:16:33.000	2087	2260	2953	12	41
8	11/14/18 21:16:43.000	2085	2243	2570	11	41
9	11/14/18 21:16:53.000	2086	2294	2870	21	41
10	11/14/18 21:17:03.000	2086	2261	2678	22	41
11	11/14/18 21:17:13.000	2086	2257	2886	15	41

Get help with parser configuration Apply Cancel

You will need to set the correct *time format* (page 583) by clicking the gear icon in Column 1 (Record time).

Parser configuration WSTUC\_5.CSV

Column delimiter: Column series Labels row: 1 Units row: None Text qualifier: UTF-8 (default) Match: Smart Reset to defaults

Timestamp	3000	3002	3004	7000	9000	9106	
#	Record Time	Number	Number	Number	Number	Number	
1	Timestamp	3000	3002	3004	7000	9000	9106
2	11/14/18 21:15:43.000	2084	919	2184	2681	11	41
3	11/14/18 21:15:53.000	2084	918	2221	2953	18	41
4	11/14/18 21:16:03.000	2083	912	2277	2748	24	41
5	11/14/18 21:16:13.000	2083	904	2168	2800	22	41
6	11/14/18 21:16:23.000	2084	901	2257	3013	19	41
7	11/14/18 21:16:33.000	2087	900	2260	2953	12	41
8	11/14/18 21:16:43.000	2085	898	2243	2570	11	41
9	11/14/18 21:16:53.000	2086	897	2294	2870	21	41
10	11/14/18 21:17:03.000	2086	897	2261	2678	22	41
11	11/14/18 21:17:13.000	2086	895	2257	2886	15	41

Get help with parser configuration Apply Cancel

Set the time format to be **MM/DD/YY HH:mm:ss.SSS** then click Close, Apply, and Next.

Parser configuration WSTUC\_5.CSV

Column delimiter	Format	Labels row	Units row	Text qualifier	Text encoding	Match	Reset to defaults
.	Column series	1	None	"	UTF-8 (default)	Smart	

Timestamp

#

Record Time

Labels	Timestamp
2	11/14/18 21:15:43.000
3	11/14/18 21:15:53.000
4	11/14/18 21:16:03.000
5	11/14/18 21:16:13.000
6	11/14/18 21:16:23.000
7	11/14/18 21:16:33.000
8	11/14/18 21:16:43.000
9	11/14/18 21:16:53.000
10	11/14/18 21:17:03.000
11	11/14/18 21:17:13.000

Column joins and configuration (Timestamp)

Column joins	Field	Format	Format filter
Timestamp	Time	MM/DD/YY HH:mm:ss.SSS	

Timestamp Quality Format Include record Add join

Get help with parser configuration

Apply Cancel

- Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.

---

## 2.14.4 dataTaker

The [dataTaker](http://datataker.com) (<http://datataker.com>) **DT80** range of data loggers can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.



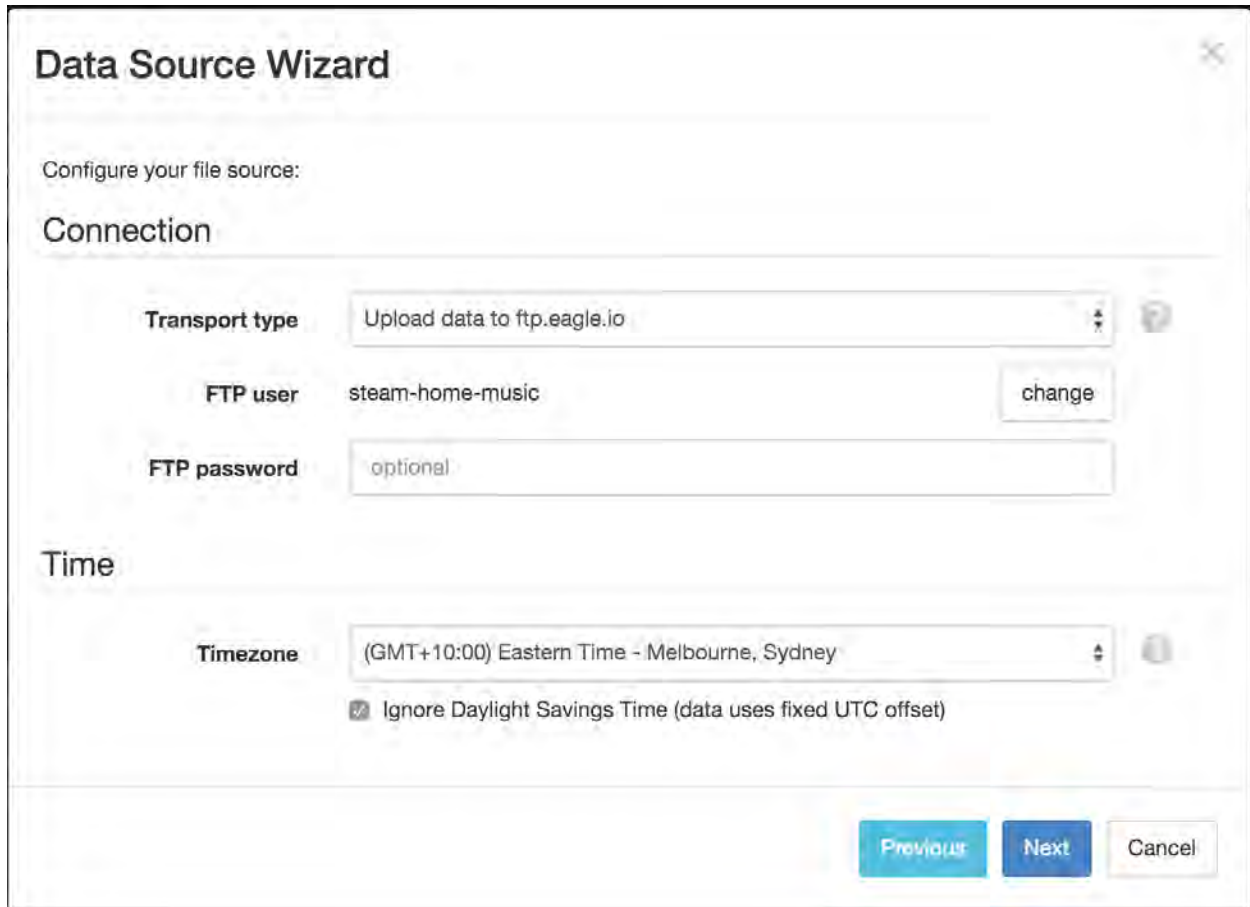
### Source configuration

1. Create a [Data Source](#) (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.

A screenshot of the 'Data Source Wizard' dialog box. The title bar says 'Data Source Wizard'. Below the title, it says 'Select source type:'. There are two radio buttons: 'File' (which is selected) and 'Data logger'. To the right of the radio buttons is a list box containing 'Delimited Text'. At the bottom right of the dialog are two buttons: 'Next' (highlighted in blue) and 'Cancel'.

2. Select *Upload data to ftp.eagle.io* as the transport type. A unique *Ftp user* account will be automatically generated per Data Source. You can optionally specify a password to be used by the logger for

additional security. Select the timezone of the logger's clock and click Next.



The image shows a 'Data Source Wizard' dialog box with a close button in the top right corner. It is divided into two main sections: 'Connection' and 'Time'. The 'Connection' section has a title bar and contains three fields: 'Transport type' with a dropdown menu set to 'Upload data to ftp.eagle.io', 'FTP user' with the text 'steam-home-music' and a 'change' button, and 'FTP password' with the text 'optional'. The 'Time' section also has a title bar and contains a 'Timezone' dropdown menu set to '(GMT+10:00) Eastern Time - Melbourne, Sydney' and a checked checkbox labeled 'Ignore Daylight Savings Time (data uses fixed UTC offset)'. At the bottom right, there are three buttons: 'Previous' (disabled), 'Next' (active), and 'Cancel'.

**Data Source Wizard**

Configure your file source:

**Connection**

**Transport type** Upload data to ftp.eagle.io

**FTP user** steam-home-music **change**

**FTP password** optional

**Time**

**Timezone** (GMT+10:00) Eastern Time - Melbourne, Sydney

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

**Previous** **Next** **Cancel**

3. Specify *CSV* as the *File name match*. This will match all incoming files for this Source containing 'CSV' in the filename. To continue the setup you will need a sample copy of the logger's output data file(s).

**Add files**

File name match: CSV [Reset]

☐ Match using regular expression

Sample file: Browse Sample.CSV

☐ Skip or Reset sample file and retrieve on next acquisition

[Apply] [Cancel]

**Note:** If you do not have a sample available, click the *Skip* checkbox and click finish on the next screen. A sample will be saved automatically on the initial upload from the data logger and you will need to open the [Data Source](#) (page 134) properties dialog and [configure the series](#) (page 119) to create parameters.

4. The Parser configuration screen allows the user to describe the structure and format of the data file so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details.



**Parser configuration** CSV

Column delimiter: , Text qualifier: " Labels row: 1 Text encoding: UTF-8 (default)

#	Timestamp	TZ	Current_3 (mm/s)	Voltage_4	Calculation_3
1	Record Time	Disabled	Number	Number	Number
2	2014/11/13 09:46:15.000	n	-0.001472	-0.158204	
3	2014/11/13 09:46:20.008	n	-0.001475	-0.152497	
4	2014/11/13 09:46:25.002	n	-0.001477	-0.156358	
5	2014/11/13 09:46:30.000	n	-0.001474	-0.156358	
6	2014/11/13 09:46:35.004	n	-0.00147	-0.158204	
7	2014/11/13 09:46:40.000	n	-0.00147	-0.156358	
8	2014/11/13 09:46:45.000	n	-0.00147	-0.160134	
9	2014/11/13 09:46:50.000	n	-0.001454	-0.156358	
10	2014/11/13 09:46:55.010	n	-0.001453	-0.161981	

Get help with parser configuration

Apply Cancel

If the file contains a header row, set the *Labels row* control to the corresponding row number which will assign default labels for parameter creation.

Ensure the data in each column matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*.

You will need to change the default time format for the *Record Time* column. Click the configure button above the column and change the *Format* field to use the *time format* (page 583) **YYYY/MM/DD HH:mm:ss.SSS**. Click Apply and Next.



**Parser configuration** CSV

Column delimiter: , Text qualifier: + Labels row: 1 Text encoding: UTF-8 (default)

**Timestamp**

Record Time

1	Timestamp
2	2014/11/13 09:46:15.000
3	2014/11/13 09:46:20.008
4	2014/11/13 09:46:25.002
5	2014/11/13 09:46:30.000
6	2014/11/13 09:46:35.004
7	2014/11/13 09:46:40.000
8	2014/11/13 09:46:45.000
9	2014/11/13 09:46:50.000
10	2014/11/13 09:46:55.010

**Timestamp** Joins and Parser Configuration

Joins	Field	Format
Timestamp	Time	YYYY/MM/DD HH:mm:ss.SSS

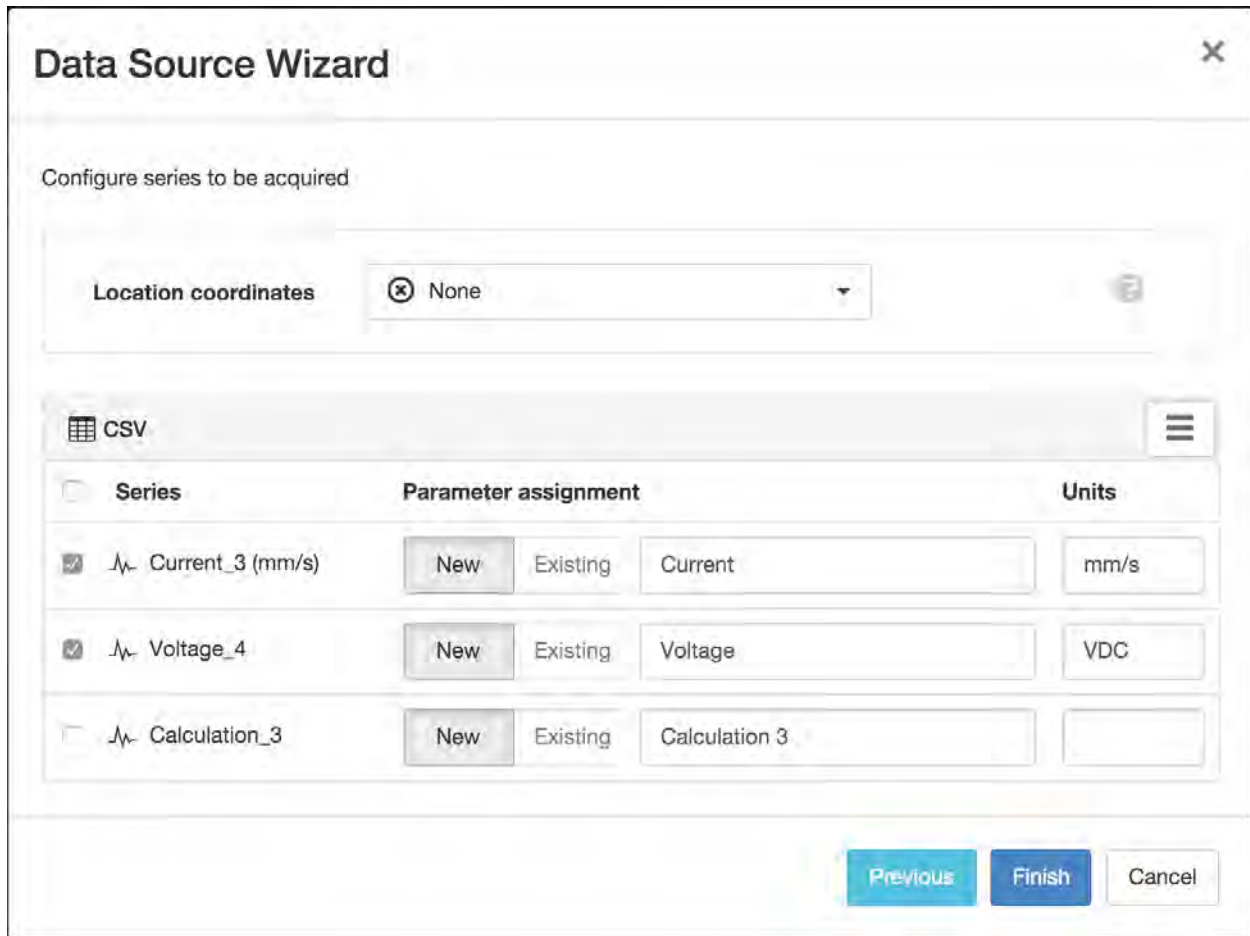
**Add New**

Timestamp Quality Add

Get help with parser configuration

Apply Cancel

- Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.



The Data Source Wizard dialog box is titled "Data Source Wizard" with a close button (X) in the top right corner. Below the title bar, the instruction "Configure series to be acquired" is displayed. A section for "Location coordinates" features a dropdown menu currently set to "None". Below this, a "CSV" tab is selected, indicated by a grid icon and a hamburger menu icon. The main area contains a table with three columns: "Series", "Parameter assignment", and "Units". The table lists three series: "Current\_3 (mm/s)", "Voltage\_4", and "Calculation\_3". Each series has a checkbox, a "New" button, an "Existing" button, a text field for the parameter name, and a text field for the units. The "Current\_3" series is checked and has "Current" assigned and "mm/s" as units. The "Voltage\_4" series is checked and has "Voltage" assigned and "VDC" as units. The "Calculation\_3" series is unchecked and has "Calculation 3" assigned and an empty units field. At the bottom right, there are three buttons: "Previous" (light blue), "Finish" (dark blue), and "Cancel" (light gray).

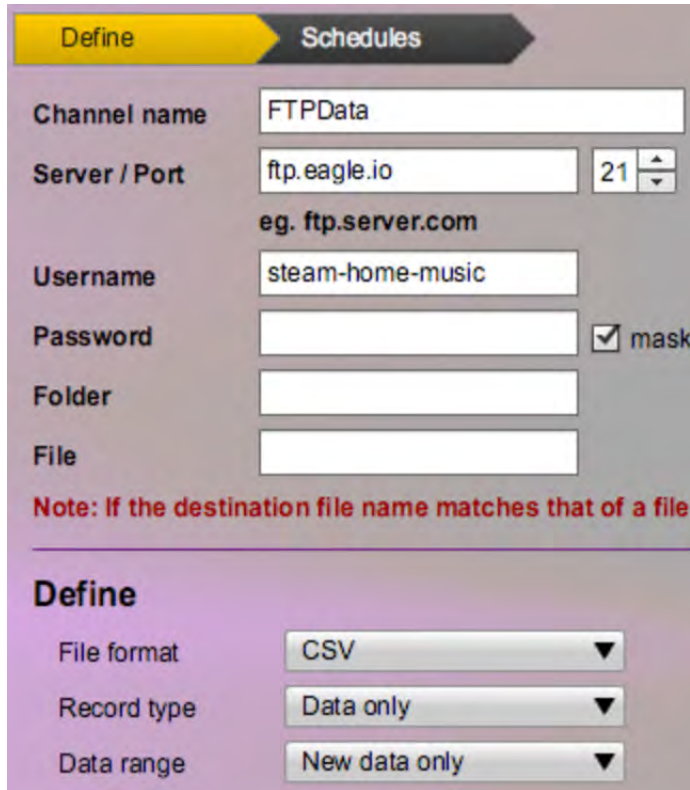
Series	Parameter assignment	Units
<input checked="" type="checkbox"/> Current_3 (mm/s)	New Existing Current	mm/s
<input checked="" type="checkbox"/> Voltage_4	New Existing Voltage	VDC
<input type="checkbox"/> Calculation_3	New Existing Calculation 3	

**Note:** You can reconfigure the file name match/sample file, parser configuration and parameter assignment from the [Data Source](#) (page 114) properties dialog after creation.

## Device settings

Use the built-in web interface of the dataTaker to configure the logger for communication with eagle.io. Refer to your dataTaker [user manual](http://www.datataker.com) (<http://www.datataker.com>) for connection instructions.

Create a new *FTP logged data* action as part of a schedule.



The screenshot shows the 'Define' tab of the eagle.io configuration interface. The 'Channel name' is set to 'FTPData'. The 'Server / Port' is set to 'ftp.eagle.io' with a port of '21'. The 'Username' is 'steam-home-music'. The 'Password' field is masked with a checkmark and the word 'mask'. The 'Folder' and 'File' fields are empty. A red note states: 'Note: If the destination file name matches that of a file'. Below this, the 'Define' section has three dropdown menus: 'File format' set to 'CSV', 'Record type' set to 'Data only', and 'Data range' set to 'New data only'.

The following settings should be assigned to the action:

**Server** ftp.eagle.io

**Port** 21

**Username** Use the unique *FTP user* generated during Data Source creation.

**Password** Use the password you assigned during Data Source creation (or leave blank for none).

**Folder & File** Leave blank

**File format** CSV

**Record type** Data only

**Data range** New data only

**Schedules** Select all schedules that should be included in the upload.

---

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

---

### 2.14.5 Spider

The [Halytech](http://halytech.com.au) (<http://halytech.com.au>) **Spider SMS** and **microSpider** range of data loggers can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.



## Source configuration

1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.

A screenshot of the 'Data Source Wizard' dialog box. The title bar says 'Data Source Wizard'. Below the title, it says 'Select source type:'. There are two radio buttons: 'File' (selected) and 'Data logger'. To the right of the radio buttons is a list box containing 'Delimited Text'. At the bottom right, there are two buttons: 'Next' (blue) and 'Cancel' (white).

2. Select *Upload data to ftp.eagle.io* as the transport type. A unique *Ftp user* account will be automatically generated per Data Source. You can optionally specify a password to be used by the logger for additional security. Select the timezone of the logger's clock and click Next.

### Data Source Wizard

Configure your file source:

#### Connection

**Transport type** Upload data to ftp.eagle.io

**FTP user** steam-home-music change

**FTP password** optional

#### Time

**Timezone** (GMT+10:00) Eastern Time - Melbourne, Sydney

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous Next Cancel

3. Specify *csv* as the *File name match*. This will match all incoming files for this Source containing 'csv' in the filename. To continue the setup you will need a sample copy of the logger's output data file(s).

### Add files

**File name match** csv Reset

☐ Match using regular expression

**Sample file** Browse sample.csv

☐ Skip or Reset sample file and retrieve on next acquisition

Apply Cancel

**Note:** If you do not have a sample available, click the *Skip* checkbox and click finish on the next screen. A sample will be saved automatically on the initial upload from the data logger and you will need to open the [Data Source](#) (page 134) properties dialog and [configure the series](#) (page 119) to create parameters.

- The Parser configuration screen allows the user to describe the structure and format of the data file so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details.

**Parser configuration** csv

Column delimiter: , Text qualifier: " Labels row: 1 Text encoding: UTF-8 (default)

	Date/Time	Stream Level	Cab Temp	Tank Pressure	Temperature
#	⌚ Record Time ⚙️	📏 Number ⚙️	📏 Number ⚙️	⊗ Disabled ⚙️	📏 Number ⚙️
1	Date/Time	Stream Level	Cab Temp	Tank Pressure	Temperature
2	23-Aug-14 06:15:00	-0.1	17.9	-0.01	17.39
3	23-Aug-14 06:30:00	-0.099	17.8	0	17.37
4	23-Aug-14 06:45:00	-0.1	17.8	-0.01	17.62

Get help with parser configuration Apply Cancel

If the file contains a header row, set the *Labels row* control to the corresponding row number which will assign default labels for parameter creation.

Ensure the data in each column matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*.

You will need to change the default time format for the *Record Time* column. Click the configure button above the column and change the *Format* field to match the [time format](#) (page 583) of the data. By default the Spider SMS requires **DD-MMM-YY HH:mm:ss** and the microSpider requires **DD-MMM-YY H:mm:ss**. Click Apply and Next.

Parser configuration

csv

Column delimiter

,

Text qualifier

"

Labels row

1

Text encoding

UTF-8 (default)

Date/Time

#

Record Time

1

Date/Time

2

23-Aug-14 06:15:00

3

23-Aug-14 06:30:00

4

23-Aug-14 06:45:00

Date/Time

Joins and Parser Configuration

Close

Joins

Field

Format

Date/Time

Time

DD-MMM-YY HH:mm:ss

Add New

Date/T

Quality

Add

Get help with parser configuration

Apply

Cancel

5. Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.



## Data Source Wizard

Configure series to be acquired

Location coordinates

None

csv

Series	Parameter assignment			Units
<input checked="" type="checkbox"/> Stream Level	New	Existing	Level	m
<input checked="" type="checkbox"/> Cab Temp	New	Existing	Cabinet Temperature	°C
<input checked="" type="checkbox"/> Temperature	New	Existing	Temperature	°C
<input type="checkbox"/> SpCond	New	Existing	SpCond	
<input checked="" type="checkbox"/> Turbidity	New	Existing	Turbidity	NTU
<input checked="" type="checkbox"/> DO	New	Existing	Dissolved Oxygen	ppm
<input type="checkbox"/> DO	New	Existing	DO (2)	

Previous

Finish

Cancel

**Note:** You can reconfigure the file name match/sample file, parser configuration and parameter assignment from the [Data Source](#) (page 114) properties dialog after creation.

## Device settings

Use the built-in web interface of the Spider SMS or microSpider to configure the logger for communication with eagle.io. Refer to your [Halytech](http://halytech.com.au) (<http://halytech.com.au>) Spider user manual for connection instructions.



The following settings should be assigned:

**Server** ftp.eagle.io

**Port** 21

**Username** Use the unique *FTP user* generated during Data Source creation.

**Password** You **must set an FTP password** or the logger will not attempt the connection. Use the password you assigned during Data Source creation or use *data* for none.

---

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

---

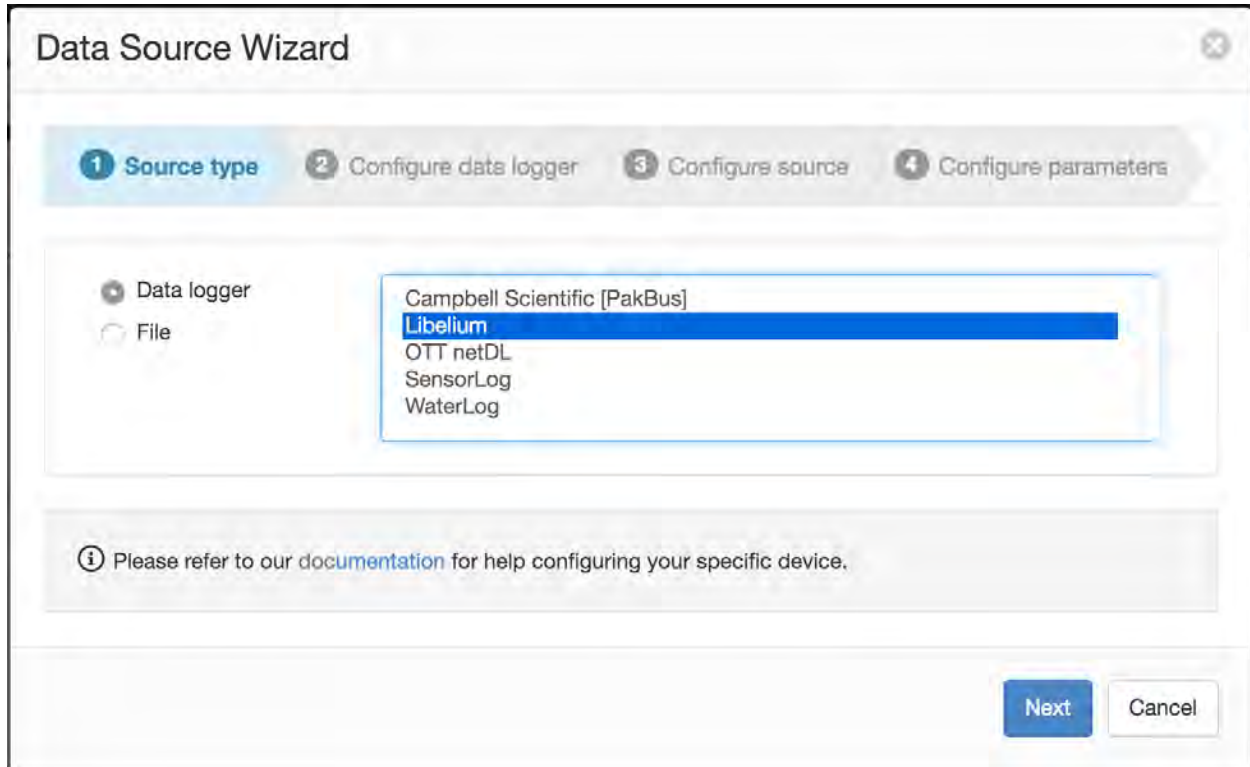
### 2.14.6 Libelium Waspote

[Libelium](http://www.libelium.com/) (<http://www.libelium.com/>) **Waspote - Plug & Sense!** loggers can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps. Waspotes can communicate directly or use a Meshlium to connect large sensor networks.



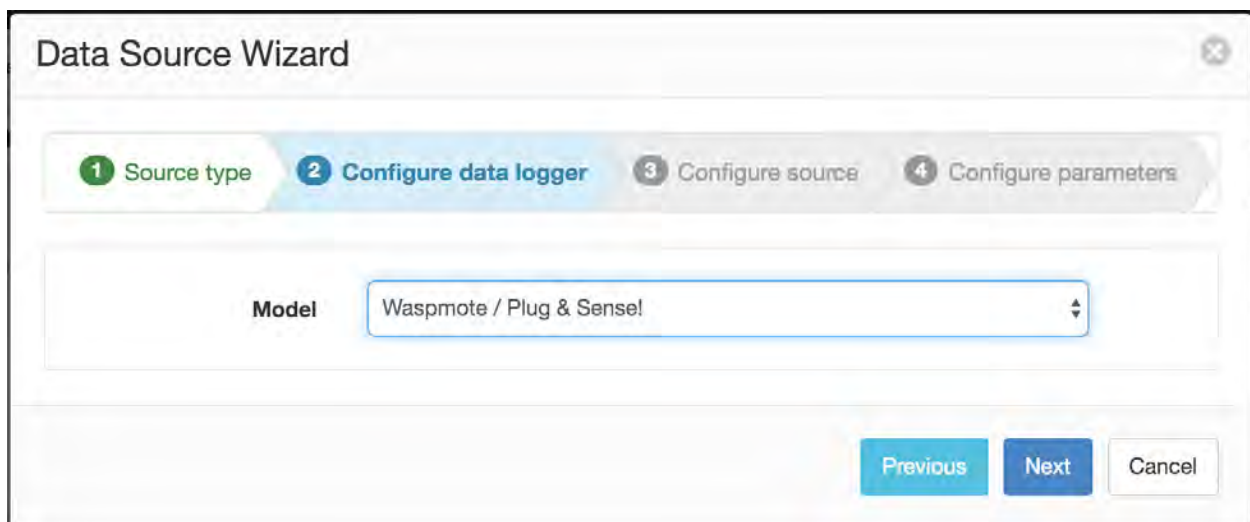
#### Source configuration

1. Create a [Data Source](#) (page 134) and select *Data logger* as the Source Type. Choose *Libelium* from the list and click Next.



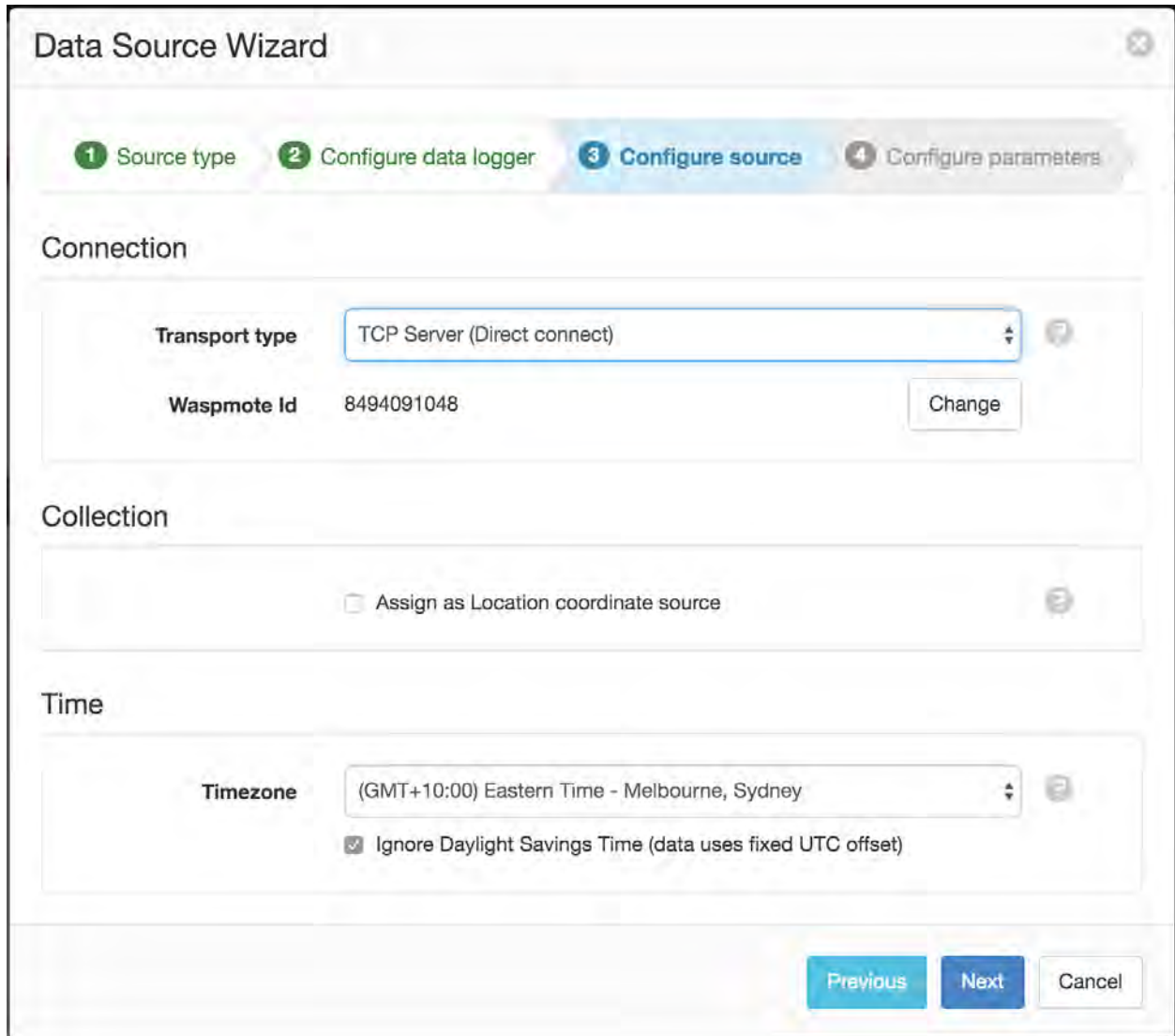
The screenshot shows the 'Data Source Wizard' window. At the top, there are four steps: 1. Source type (highlighted in blue), 2. Configure data logger, 3. Configure source, and 4. Configure parameters. Below the steps, there are two radio buttons: 'Data logger' (selected) and 'File'. To the right of the 'Data logger' radio button is a list box containing the following options: 'Campbell Scientific [PakBus]', 'Libellium' (highlighted in blue), 'OTT netDL', 'SensorLog', and 'WaterLog'. Below the list box, there is a note: 'Please refer to our [documentation](#) for help configuring your specific device.' At the bottom right, there are two buttons: 'Next' (blue) and 'Cancel' (white).

2. Select *Waspnote / Plug & Sense!* as the *Model*. Click Next.



The screenshot shows the 'Data Source Wizard' window, now at step 2: 'Configure data logger'. The steps at the top are: 1. Source type, 2. Configure data logger (highlighted in blue), 3. Configure source, and 4. Configure parameters. Below the steps, there is a label 'Model' followed by a dropdown menu. The dropdown menu is open, showing the option 'Waspnote / Plug & Sense!'. At the bottom right, there are three buttons: 'Previous' (blue), 'Next' (blue), and 'Cancel' (white).

3. Select *TCP Server* as the transport type if the Waspnote will connect directly to eagle.io (alternatively jump to Step 4 for Meshlium connection). A *Waspnote Id* will be automatically generated. Select the timezone of the logger's clock and click Next.



The image shows a 'Data Source Wizard' window with a progress bar at the top indicating four steps: 1. Source type, 2. Configure data logger, 3. Configure source (highlighted), and 4. Configure parameters. The 'Connection' section contains a 'Transport type' dropdown menu set to 'TCP Server (Direct connect)' and a 'Wasmote Id' field with the value '8494091048' and a 'Change' button. The 'Collection' section has an unchecked checkbox labeled 'Assign as Location coordinate source'. The 'Time' section features a 'Timezone' dropdown menu set to '(GMT+10:00) Eastern Time - Melbourne, Sydney' and a checked checkbox labeled 'Ignore Daylight Savings Time (data uses fixed UTC offset)'. At the bottom right, there are three buttons: 'Previous' (blue), 'Next' (blue), and 'Cancel' (white).

**Data Source Wizard**

1 Source type 2 Configure data logger 3 **Configure source** 4 Configure parameters

**Connection**

Transport type TCP Server (Direct connect)

Wasmote Id 8494091048 Change

**Collection**

☐ Assign as Location coordinate source

**Time**

Timezone (GMT+10:00) Eastern Time - Melbourne, Sydney

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous Next Cancel

4. Select *Meshlium* if the Wasmote will send data via a Meshlium (alternatively jump to Step 6). Enter the *Meshlium Id* and optional *Auth key* if configured in the eagle.io application on the device. Click Verify.

**Data Source Wizard**

1 Source type   2 Configure data logger   **3 Configure source**   4 Configure parameters

**Connection**

Transport type: Meshlium

Meshlium Id: 31324213418649

Auth key: secret key configured in application (optional)

Verify

**Collection**

☐ Assign as Location coordinate source

**Time**

Timezone: (GMT+10:00) Eastern Time - Melbourne, Sydney

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous   Next   Cancel

**Note:** The eagle.io application in the Meshlium device needs to be started and initial connection established with eagle.io before it can be used as a transport.

5. Select the *Waspnote* you would like to acquire data from and click Next.

Connection

Transport type: Meshlium

Meshlium: meshlium [31324213418649] Meshlium last connected 2 month ago

Change

Waspote: wasp-01 [403469691]

- Parameters will be automatically created on initial data acquisition. Click Finish to complete the setup.

Data Source Wizard

1 Source type 2 Configure data logger 3 Configure source 4 Configure parameters

Sensors

Series will be available on initial data acquisition
------------------------------------------------------

Previous Finish Cancel

**Note:** You can disable/enable parameters from the [Data Source](#) (page 134) properties dialog after initial creation.

## Device settings

Use the settings below for *Direct Connection* or *Meshlium Connection* with your Waspote.

## Direct Connection

Refer to our example Wasmote operating programs for use with 3G (download) and 4G modems (download) to establish a direct connection to eagle.io.

Apply the following settings to the variables at the top of the operating program:

**wasmoteId** Use the unique *Wasmote Id* generated during Data Source creation. eg. “8494091048”

**host** “device.eagle.io”

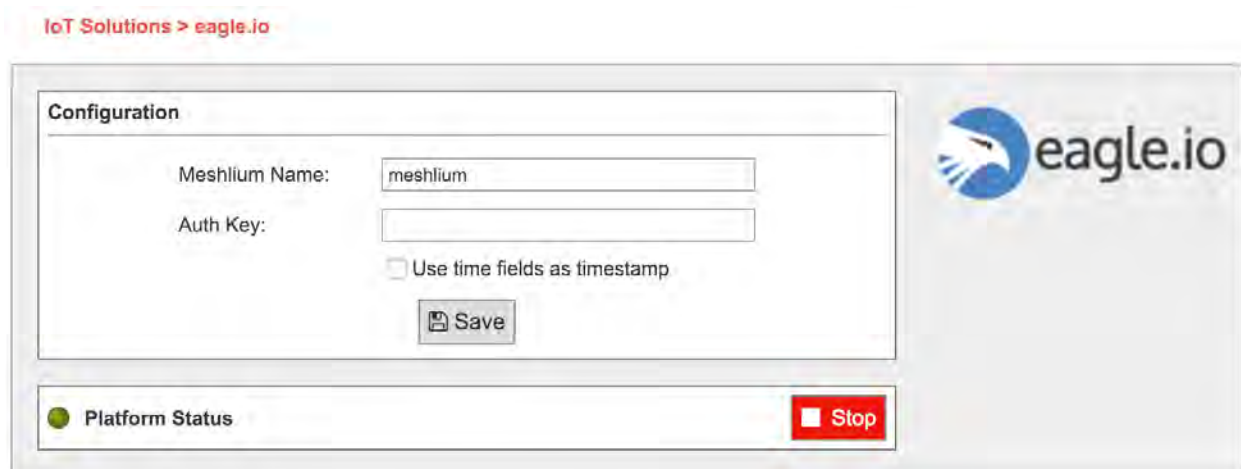
**port** “5566”

## Meshlium Connection

Connect to the web interface on the Meshlium device, open the eagle.io Cloud Connector configuration and *Save* the following settings prior to clicking *Start* to run the application:

**Meshlium Name** Enter a name to help identify this device

**Auth Key** Optional secret key required to create a Data Source that uses this device in eagle.io (or leave blank for none).



The screenshot shows the 'IoT Solutions > eagle.io' breadcrumb at the top. The main configuration area is titled 'Configuration' and contains the following fields and controls:

- Meshlium Name:** A text input field containing the value 'meshlium'.
- Auth Key:** An empty text input field.
- ☐ Use time fields as timestamp
- Save:** A button with a floppy disk icon.

At the bottom of the configuration area, there is a 'Platform Status' section with a green circular indicator and a red 'Stop' button.

---

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

---

### 2.14.7 Worldsensing

[Loadsensing](https://www.worldsensing.com/loadsensing/) (<https://www.worldsensing.com/loadsensing/>) is the IoT remote monitoring solution by [Worldsensing](https://www.worldsensing.com/) (<https://www.worldsensing.com/>), a global industrial IoT leader and expert in monitoring the

geotechnical, geospatial and structural performance of civil infrastructures. The monitoring solution can be connected with [eagle.io](https://eagle.io) (<https://eagle.io>) through the Worldsensing software called **Connectivity Management Tool (CMT)** (<https://www.worldsensing.com/connectivity-connectivitymanagement/>) in a few easy steps.



## Introduction

**CMT (Connectivity Management Tool)** (<https://www.worldsensing.com/connectivity-connectivitymanagement/>) is part of the Worldsensing connectivity portfolio. It allows integrating data collected through Worldsensing monitoring networks with 3rd party systems. The CMT software features FTP push of data to 3rd party systems, such as [eagle.io](https://eagle.io).

## Setting up CMT with eagle.io

Follow the steps below to configure CMT to send data to an eagle.io data source. Use a similar process to connect CMT Edge to eagle.io.

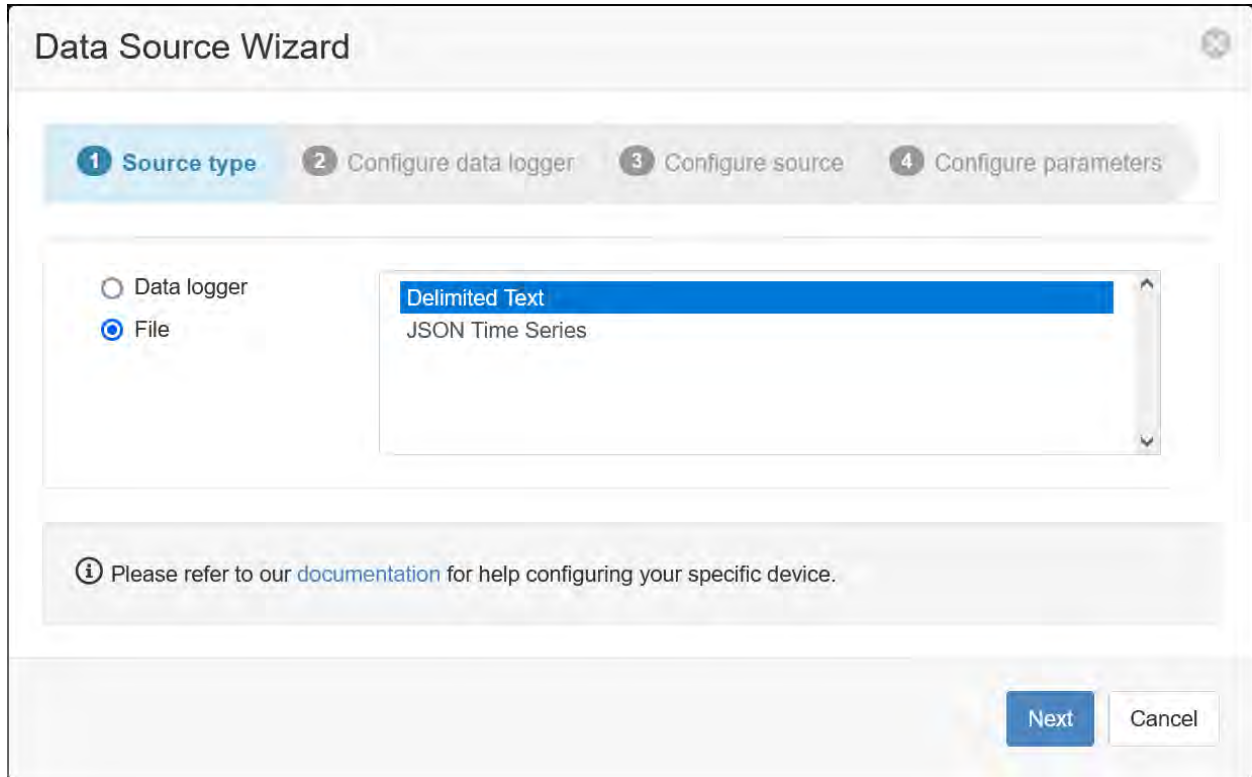
### 3 Steps for an easy setup:

1. Configure a new eagle.io data source with the **Upload data to ftp.eagle.io** transport
2. Configure CMT with FTP client details
3. Use the first file sent as sample data to finalize configuring the data source parser, and create parameters

### 1. Configure an eagle.io data source

1. Create a *data source* (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click **Next**.





The image shows a 'Data Source Wizard' dialog box with a close button in the top right corner. It features a progress bar at the top with four steps: 1. Source type (highlighted in blue), 2. Configure data logger, 3. Configure source, and 4. Configure parameters. Below the progress bar, there are two radio buttons: 'Data logger' and 'File'. The 'File' radio button is selected. To the right of the radio buttons is a list box containing two options: 'Delimited Text' (which is highlighted in blue) and 'JSON Time Series'. At the bottom of the dialog, there is an information icon followed by the text 'Please refer to our [documentation](#) for help configuring your specific device.' and two buttons: 'Next' and 'Cancel'.

2. Select *Upload to ftp.eagle.io* as the transport type. A unique *Username* will be automatically generated per data source; for this example, it is *spoon-table-node*. A password should also be entered for additional security. Select the desired timezone (in this example it is **UTC** to match the timezone of the data file) and click **Next**.

The screenshot shows the 'Data Source Wizard' window with four steps: 1. Source type, 2. Configure source (active), 3. Add files, and 4. Configure parameters. The 'Configure source' section is divided into three parts: Connection, Collection, and Time.

**Connection**

- Transport type:** A dropdown menu set to 'Upload to ftp.eagle.io'.
- Username:** A text field containing 'spoon-table-node'.
- Password:** A text field with masked characters (dots) and a toggle icon.
- IP whitelist:** A list box with a search bar containing 'eg. 192.168.7.52/32' and 'Add'/'Remove' buttons.

**Collection**

- ☐ Assign as Location coordinate source

**Time**

- Timezone:** A dropdown menu set to '(GMT+00:00) UTC'.
- ☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

At the bottom right, there are three buttons: 'Previous' (blue), 'Next' (blue), and 'Cancel' (white).

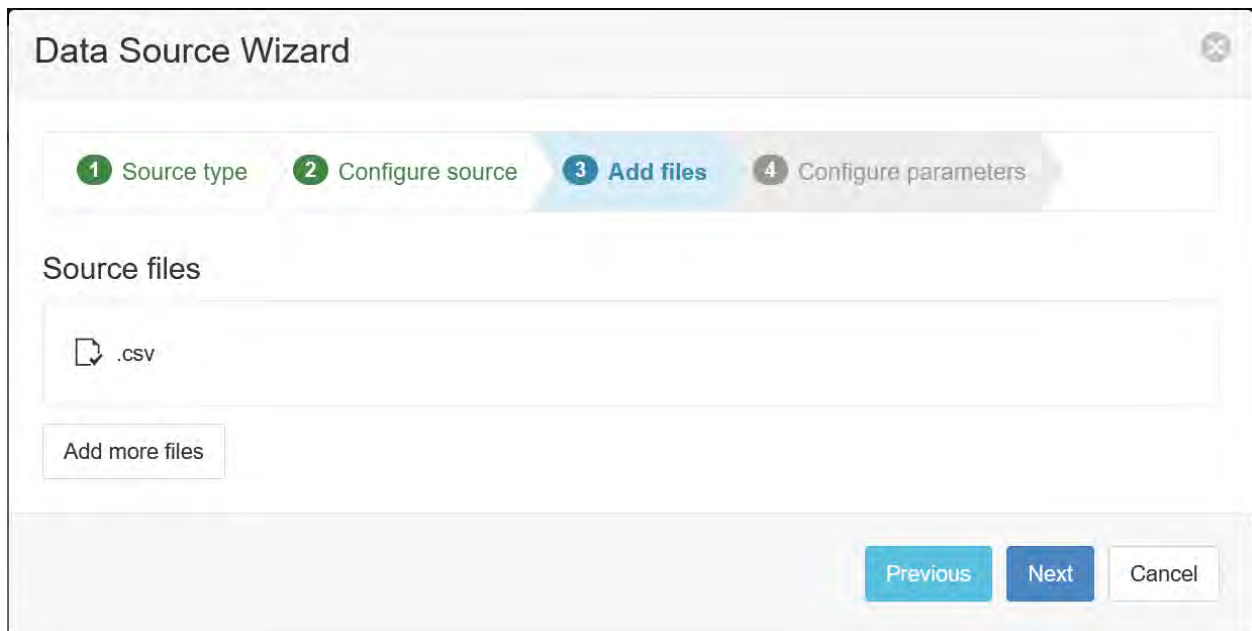
3. Check the box labeled **Skip or Reset sample file and retrieve on next acquisition**. Enter a file match of `.csv` and click **Apply**.



The 'Sample data' dialog box contains the following elements:


- Sample file:** A 'Browse' button and a text field with the placeholder 'Select a sample file'.
- ☒ Skip or Reset sample file and retrieve on next acquisition
- File name match:** A text field containing '.csv' and a 'Reset' button.
- ☐ Match using regular expression
- Buttons:** 'Apply' and 'Cancel' at the bottom right.

3. Click **Next**.



The 'Data Source Wizard' dialog box features a progress bar with four steps: 1 Source type, 2 Configure source, 3 Add files (highlighted), and 4 Configure parameters.

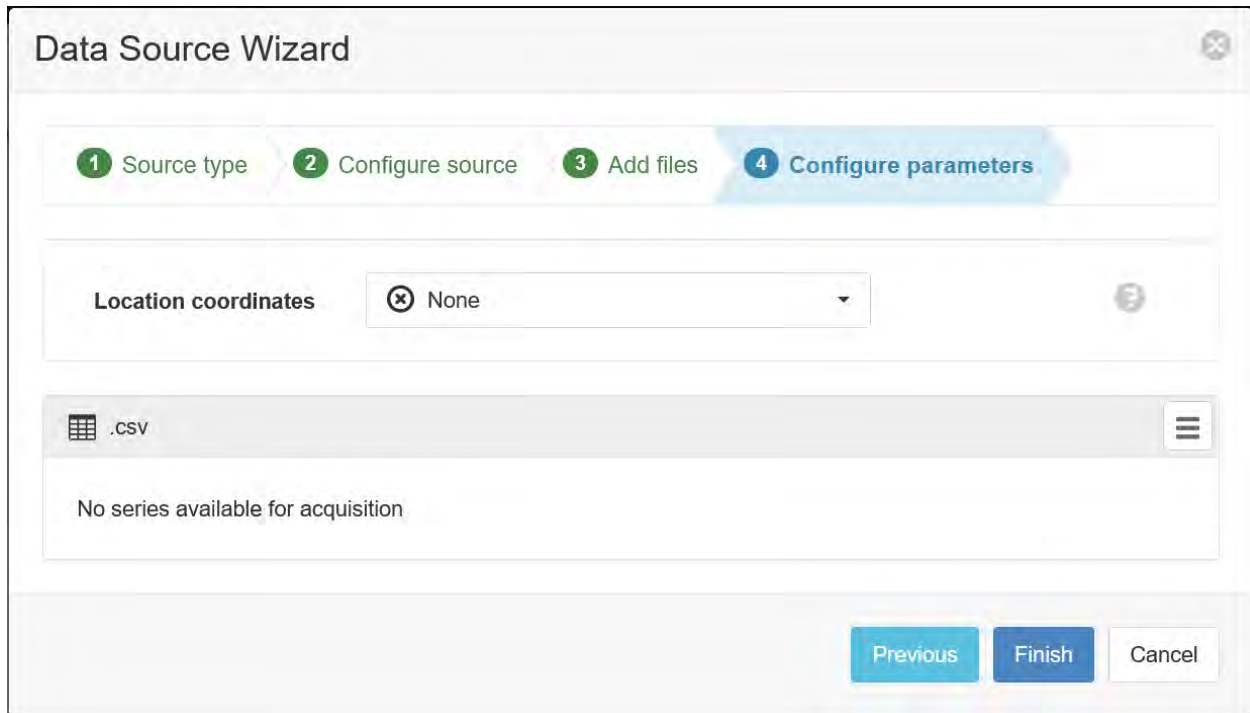
**Source files**

A list box shows a single file:  .csv

**Add more files**

**Buttons:** 'Previous', 'Next', and 'Cancel' at the bottom right.

4. Click **Finish**.



The image shows a 'Data Source Wizard' dialog box with a close button in the top right corner. It features a progress bar with four steps: 1 Source type, 2 Configure source, 3 Add files, and 4 Configure parameters. Step 4 is currently selected and highlighted in blue. Below the progress bar, there is a 'Location coordinates' section with a dropdown menu set to 'None' and a help icon. Underneath is a list of data sources, showing a single entry for '.CSV' with a table icon and a menu icon. Below the list, it states 'No series available for acquisition'. At the bottom right, there are three buttons: 'Previous' (disabled), 'Finish' (active), and 'Cancel'.

Data Source Wizard

1 Source type 2 Configure source 3 Add files 4 Configure parameters

Location coordinates

.CSV

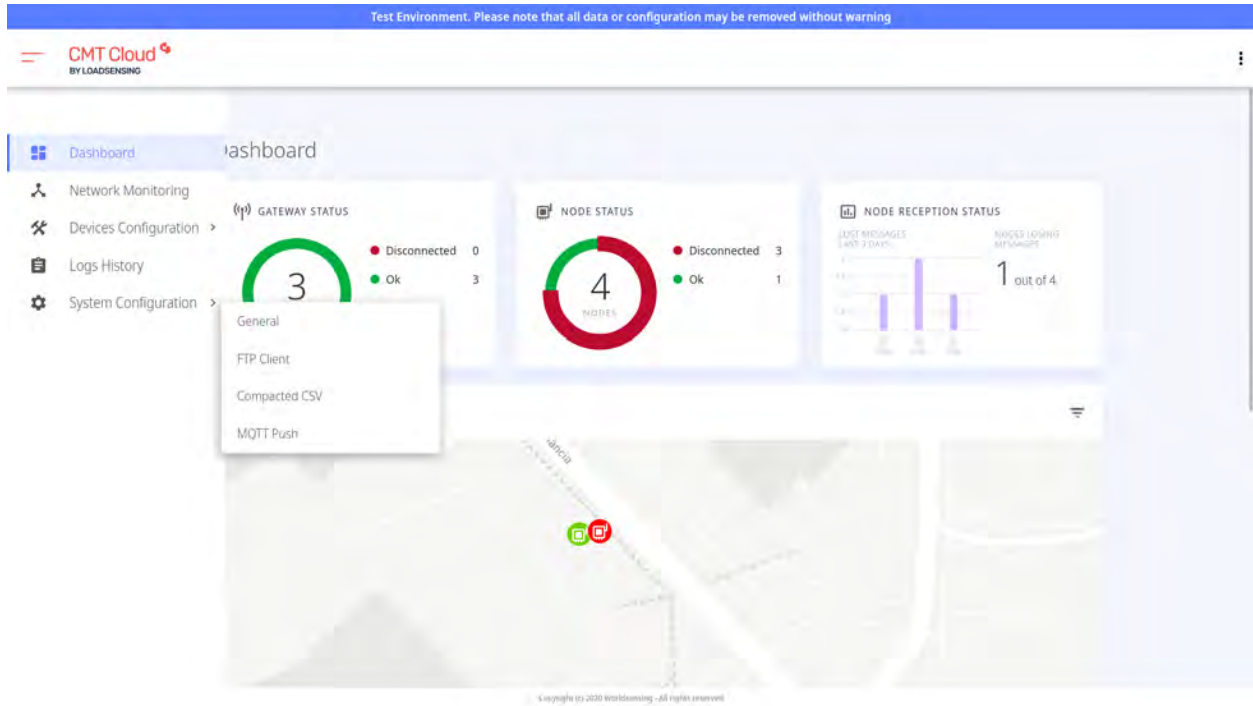
No series available for acquisition

Previous Finish Cancel

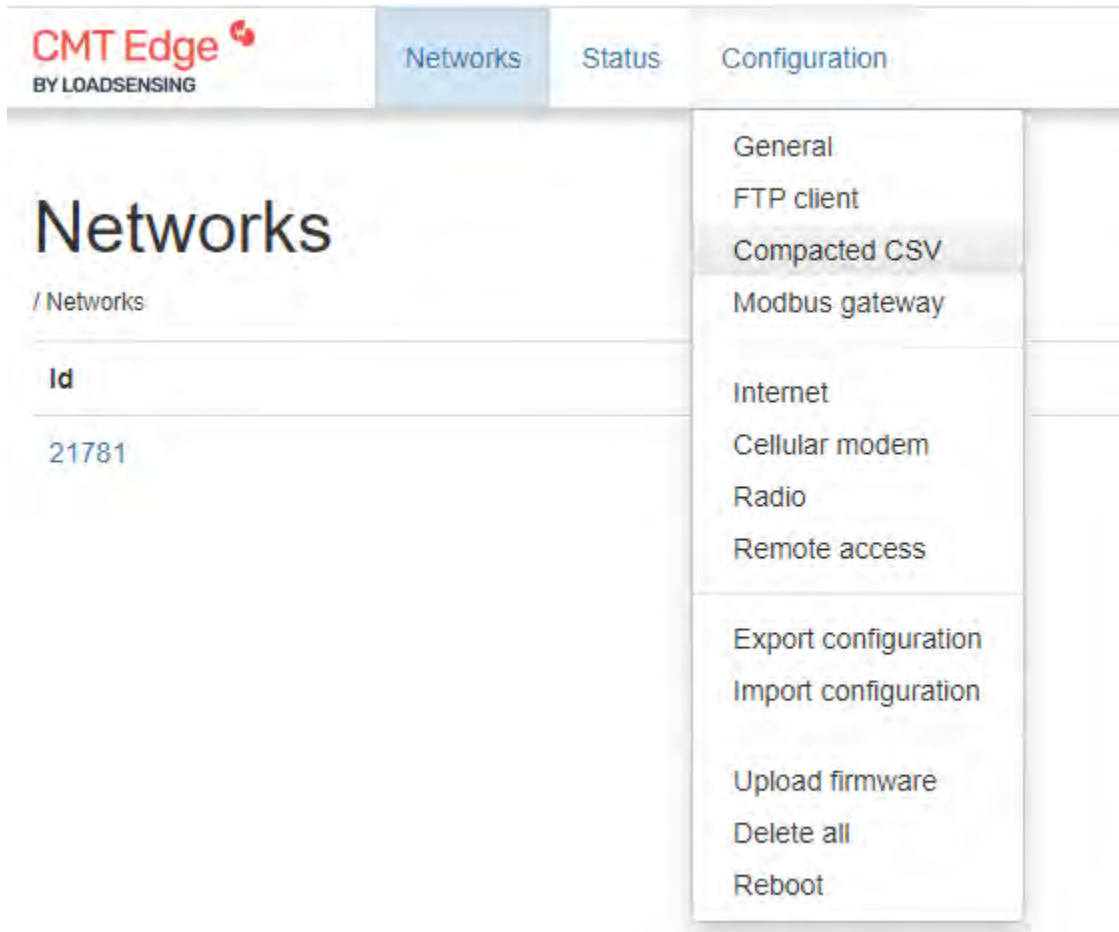
## 2. Configure CMT

1. After logging into the CMT, select *System Configuration/Configuration*, then *FTP Client*.

CMT Cloud:



CMT Edge:



2. Check the **Enable FTP** box and enter the following FTP client details:

**Hostname** `ftp.eagle.io`

**Port number** `21`

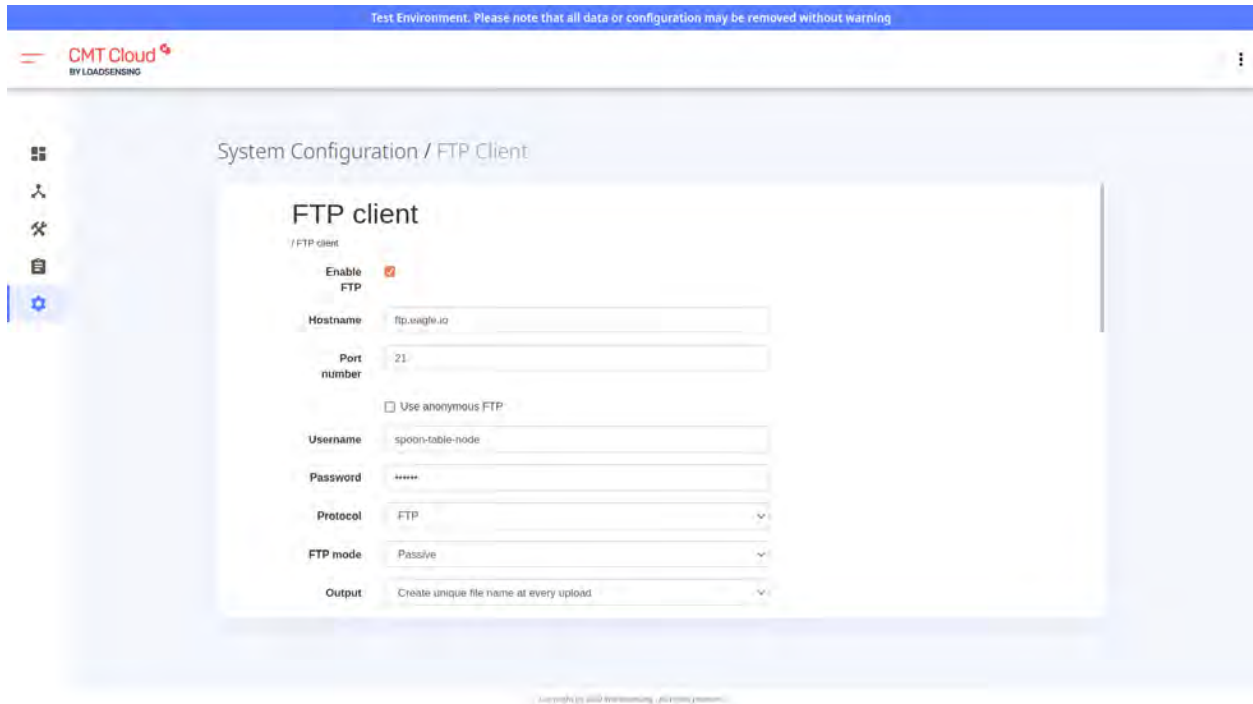
**Username** Use the unique FTP username generated during creation of the data source. In this example, it is `spoon-table-node`. Every data source will have a different unique username.

**Password** Use the password you entered during creation of the data source.

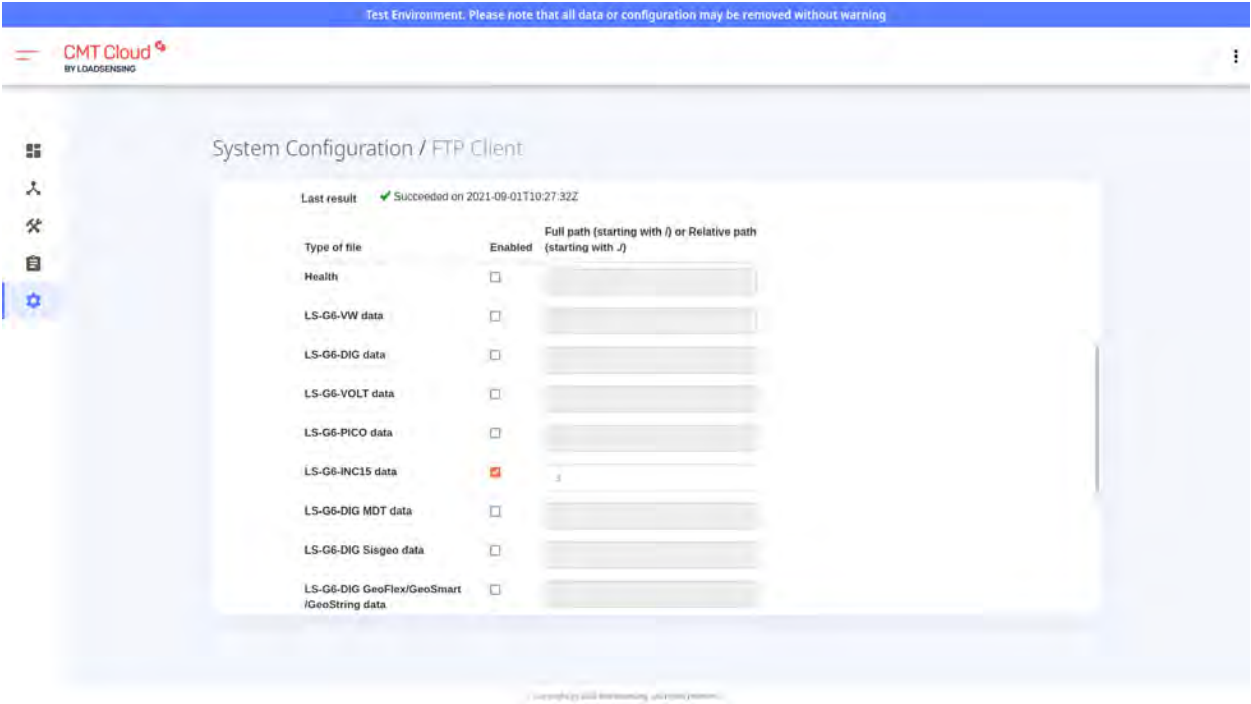
**Protocol** `FTP`

**FTP mode** `Passive`

**Output** `Create unique file name at every upload`

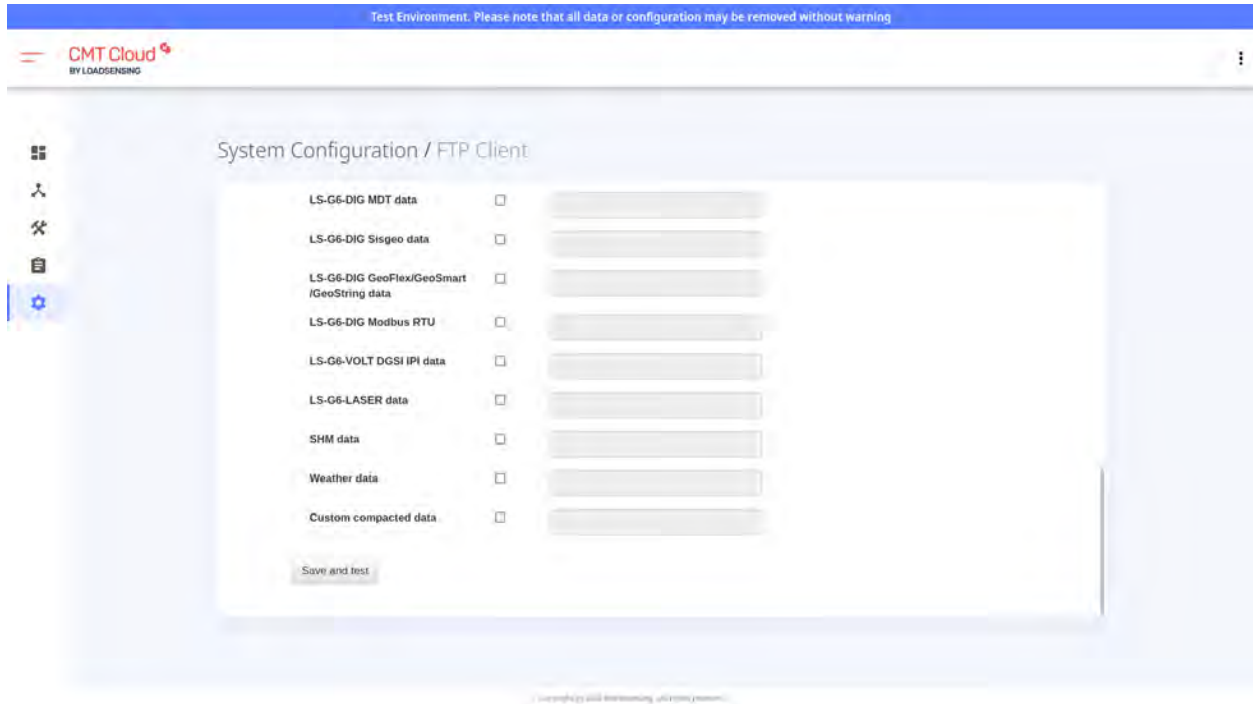


3. Click a single box corresponding to the data you want to send, for example **LS-G6-INC15 data**, and enter a relative path of `./`:



4. Click **Save and test**

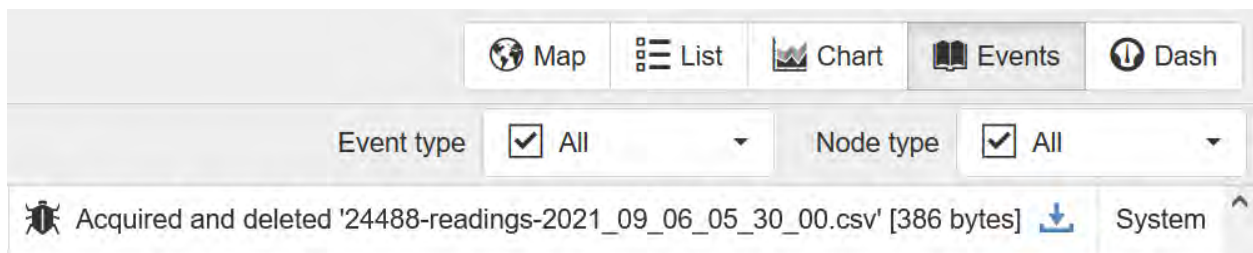




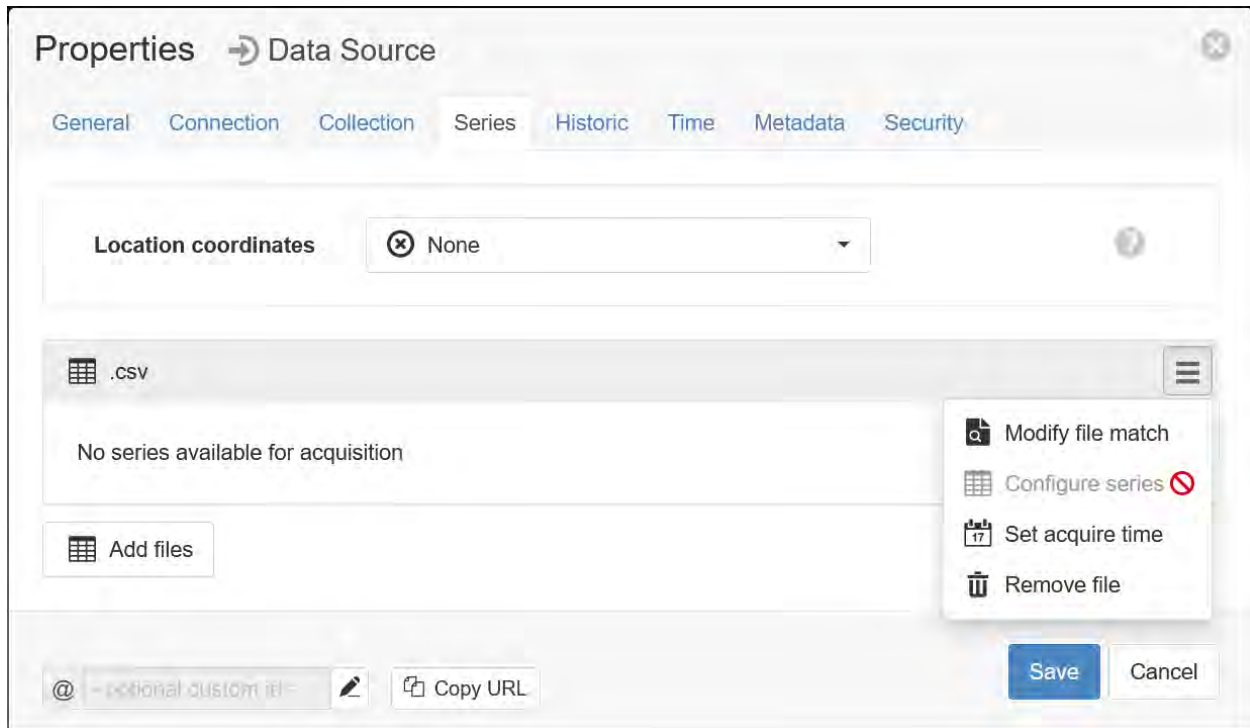
### 3. Finalize the configuration of the data source parser

Once a file has been sent from the CMT software to eagle.io via FTP, this file can be used to finalize the configuration of the data source that was created previously.

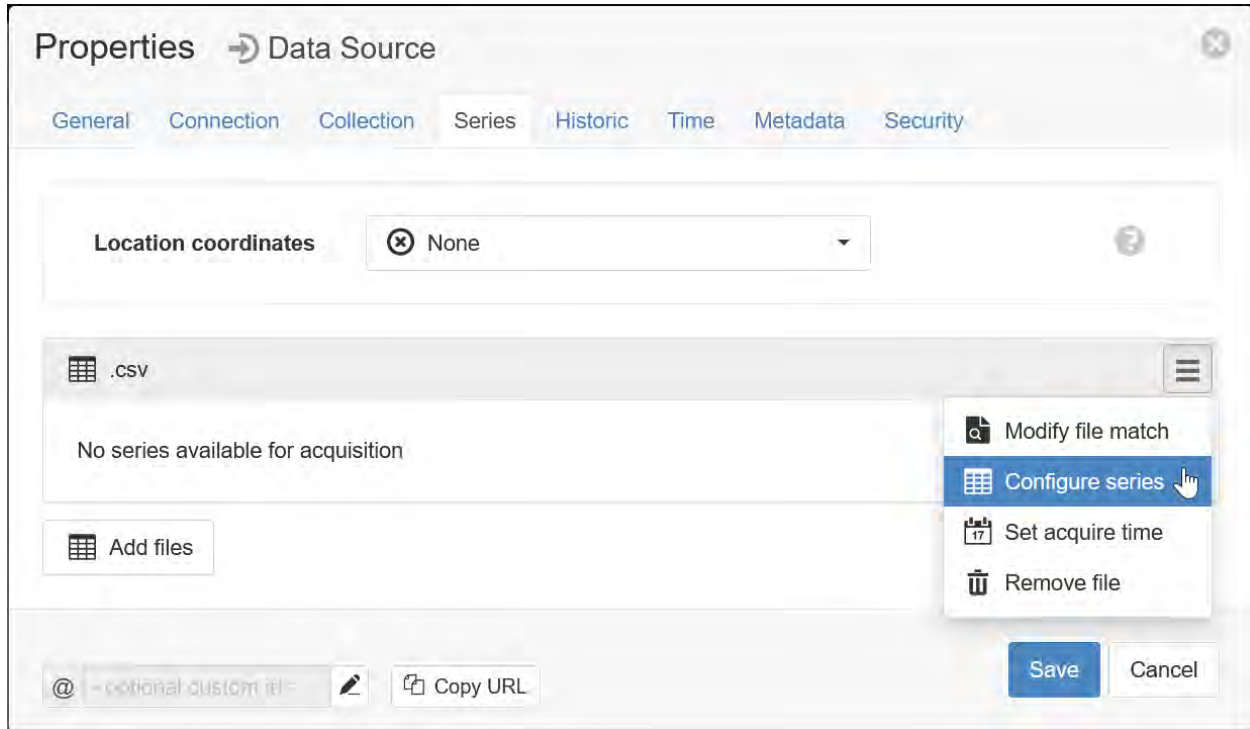
1. In eagle.io, check the debug [Events](#) (page 43) view of the data source to ensure that at least one file has been received. The event will look like this:



2. In the *data source* (page 114) properties dialog, go to the **Series** tab and choose the **Configure series** option for the **.csv** data file. If this option is greyed-out, that means a file has not yet been received (refer to the previous step):



3. Once a file has been received, the **Configure series** option for the **.csv** data file will be available:



4. The Parser configuration screen allows the user to describe the structure and format of the data file so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details. The data file contains a header row, so set the *Labels row* control to row 10 which will assign default labels for parameter creation (that is the purple line in the example below). Ensure the data in each column matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*. The default time format for the *Record Time* column is already set to **YYYY-MM-DD HH:mm:ss** and should not need to change. After appropriate configuration, the parser should look now look like this; click **Apply** to continue:

Parser configuration .csv

Column delimiter	Format	Labels row	Units row	Decimal separator	Text qualifier	Text encoding	Match	Success required
Column series	Column series	10	None	Period	"	UTF-8 (default)	Smart	<input type="checkbox"/> Success required <input type="button" value="Reset to defaults"/>

	Date-and-time	Temp-24488-Ch1	Aaxis-24488-Ch1	Baxis-24488-Ch1	AaxisVariation-24488-Ch1	BaxisVariation-24488-Ch1	Column 7
	<input type="checkbox"/> Record Time	<input type="checkbox"/> Number	<input type="checkbox"/> Number	<input type="checkbox"/> Number	<input type="checkbox"/> Number	<input type="checkbox"/> Number	<input checked="" type="checkbox"/> Disabled
1	Node ID	24488					
2	Gateway ID	810394605443143331					
3	Model	LS-G6-INC15					
4	Hw version						
5	Fw version						
6	Location Lat						
7	Location Lon						
8	Created time	2021-09-01 00:00:10					
9	Timezone	UTC					
LABELS	Date-and-time	Temp-24488-Ch1	Aaxis-24488-Ch1	Baxis-24488-Ch1	AaxisVariation-24488-Ch1	BaxisVariation-24488-Ch1	
11	2021-09-06 11:30:00	25.3	-0.8443	-0.1823	-11111.8443	-2222.1823	

- Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click **Save** to complete the configuration and create the parameters.

Properties

→ Data Source

General

Connection

Collection

Series

Historic

Time

Metadata

Security

Location coordinates

⊗ None

.CSV

<input checked="" type="checkbox"/> Series	Parameter assignment			Units
<input checked="" type="checkbox"/> Temp-24488-Ch1	New	Existing	Temp-24488-Ch1	
<input checked="" type="checkbox"/> Aaxis-24488-Ch1	New	Existing	Aaxis-24488-Ch1	
<input checked="" type="checkbox"/> Baxis-24488-Ch1	New	Existing	Baxis-24488-Ch1	
<input checked="" type="checkbox"/> AaxisVariation-24488-Ch1	New	Existing	AaxisVariation-24488-Ch1	
<input checked="" type="checkbox"/> BaxisVariation-24488-Ch1	New	Existing	BaxisVariation-24488-Ch1	

Add files

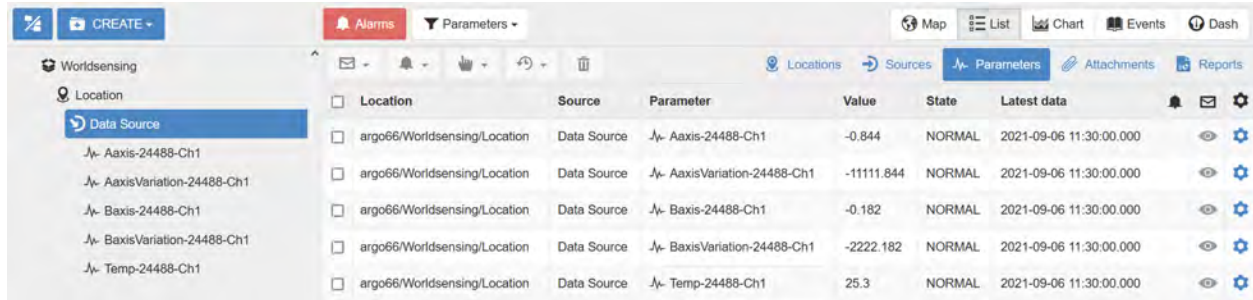
@ - optional custom id -

Copy URL

Save

Cancel

- The parameters have now been created; when data is next sent via FTP, the values will appear in the Parameters view:



Location	Source	Parameter	Value	State	Latest data	
<input type="checkbox"/> argo66/Worldsensing/Location	Data Source	J\ Axis-24488-Ch1	-0.844	NORMAL	2021-09-06 11:30:00.000	
<input type="checkbox"/> argo66/Worldsensing/Location	Data Source	J\ AxisVariation-24488-Ch1	-11111.844	NORMAL	2021-09-06 11:30:00.000	
<input type="checkbox"/> argo66/Worldsensing/Location	Data Source	J\ Baxis-24488-Ch1	-0.182	NORMAL	2021-09-06 11:30:00.000	
<input type="checkbox"/> argo66/Worldsensing/Location	Data Source	J\ BaxisVariation-24488-Ch1	-2222.182	NORMAL	2021-09-06 11:30:00.000	
<input type="checkbox"/> argo66/Worldsensing/Location	Data Source	J\ Temp-24488-Ch1	25.3	NORMAL	2021-09-06 11:30:00.000	

**Note:** You can reconfigure the file name match/sample file, parser configuration and parameter assignment from the [data source](#) (page 114) properties dialog after creation.

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

### 2.14.8 Maxon Intelimax+

The [Maxon Intelimax+ 3G SERIAL MA-2020](http://www.maxon.com.au/products/modems-routers/INTELIMAX+_3G_SERIAL_MA-2020.html) ([http://www.maxon.com.au/products/modems-routers/INTELIMAX+\\_3G\\_SERIAL\\_MA-2020.html](http://www.maxon.com.au/products/modems-routers/INTELIMAX+_3G_SERIAL_MA-2020.html)) modem can be quickly configured to connect your data logger to [eagle.io](https://eagle.io) (<https://eagle.io>). The *Intelimax+* takes any standard internet ready SIM Card without the need for fixed IP addresses and offers exceptional performance in locations with marginal signal strength.



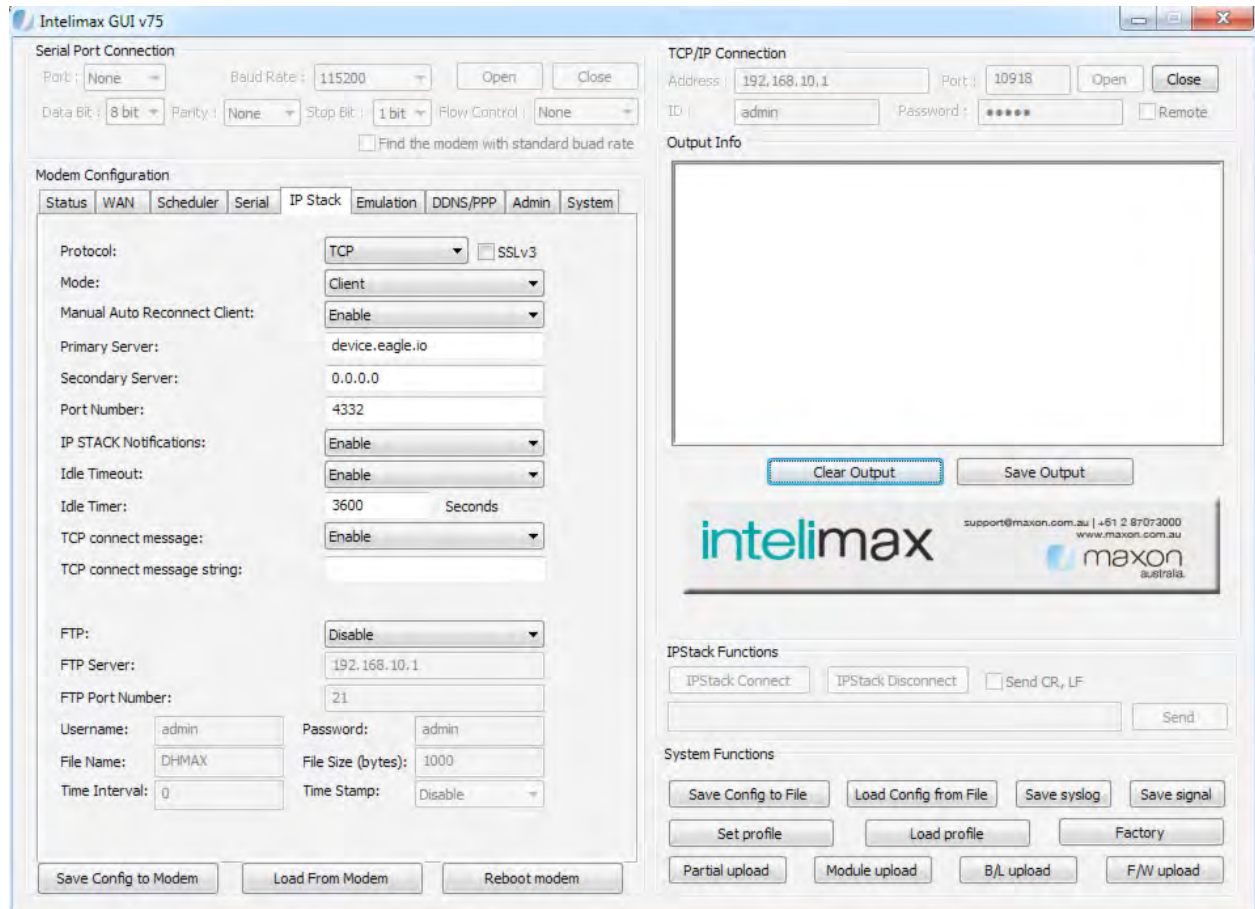
### Source configuration

You need to configure your Data Source to use the **TCP Callback** transport. Click the *use existing* button and enter in the IMEI number printed on the back of the modem. See [configuring a Data Source](#) (page 134) for full details.

<b>Transport type</b>	<div>TCP Callback</div> <div>?</div>
<b>Device Id</b>	356938035643809
	<div>use existing</div>

## Device settings

Use the [Intelimax GUI software](http://www.maxon.com.au/product-supports/drivers-and-manuals.html) (<http://www.maxon.com.au/product-supports/drivers-and-manuals.html>) to configure the logger for communication with eagle.io. Refer to your Intelimax+ [user manual](http://www.maxon.com.au/products/modems-routers/INTELIMAX+_3G_SERIAL_MA-2020.html) ([http://www.maxon.com.au/products/modems-routers/INTELIMAX+\\_3G\\_SERIAL\\_MA-2020.html](http://www.maxon.com.au/products/modems-routers/INTELIMAX+_3G_SERIAL_MA-2020.html)) for connection instructions.



The following settings should be assigned:

**Protocol** TCP

**Mode** Client

**Primary Server** device.eagle.io

**Port Number** 4332

**TCP connect message** Enable



### 2.14.9 Myriota Developer Toolkit

The [Myriota Network](http://myriota.com) (<http://myriota.com>) is a next generation, direct-to-orbit IoT technology platform that offers data transfer at mass scale.

The [Myriota Developer Toolkit](http://myriota.com/developers) (<http://myriota.com/developers>) gives you all the tools you need to add affordable, long battery life IoT connectivity to your product. This unique direct-to-satellite technology offers the lowest cost of data transfer and the longest battery life for applications, and can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.



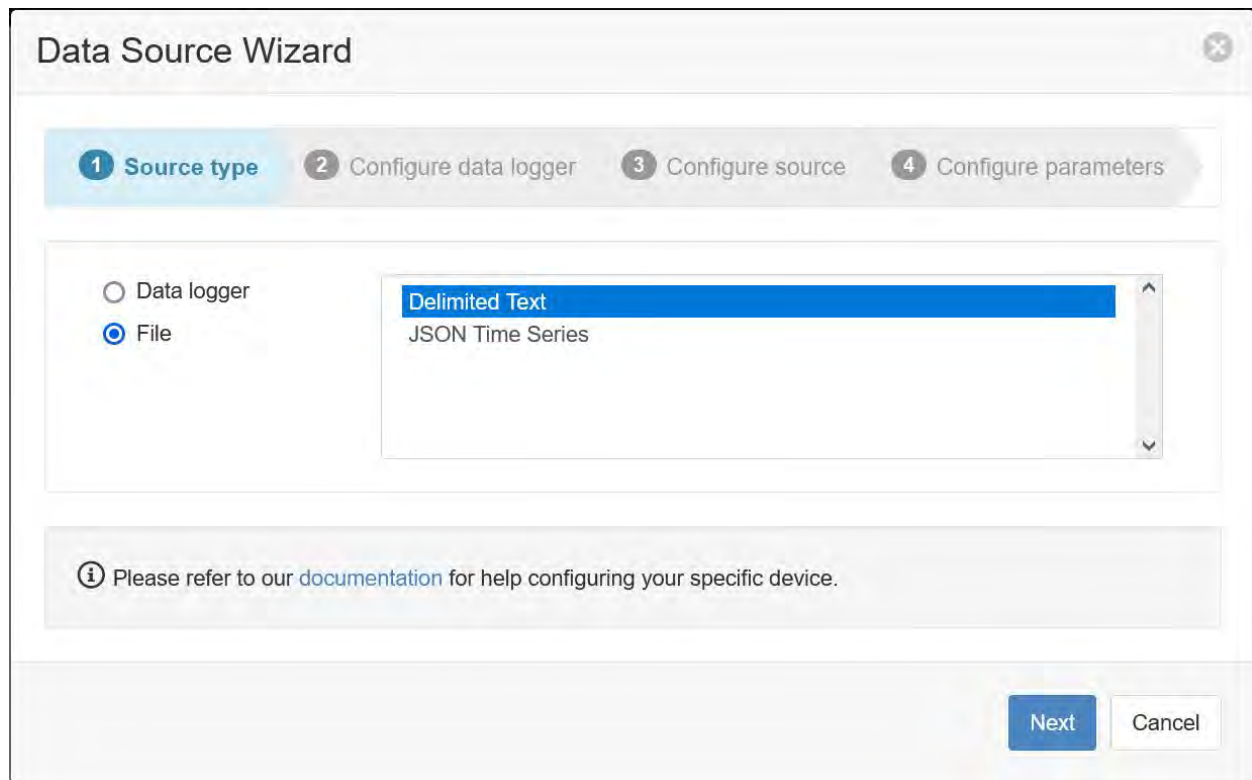
#### Introduction

There are 5 main tasks required when setting up the Myriota Developer Toolkit to work with eagle.io

1. Configure an eagle.io data source with the **Upload via HTTP** transport
2. Define Myriota message format using Ruby unpack syntax
3. Determine the destination URL (defined by the data source and the message format)
4. Send a sample message to the destination URL with the send message function
5. Use the sample message to configure the eagle.io parser, and map/name your variables

## 1. Configure an eagle.io data source

1. Create a *Data Source* (page 114) (note that a Data Source node can only be created within a *Location* (page 56) node) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.



The image shows a 'Data Source Wizard' dialog box with a close button in the top right corner. It features a progress bar at the top with four steps: 1. Source type (highlighted in blue), 2. Configure data logger, 3. Configure source, and 4. Configure parameters. Below the progress bar, there are two radio buttons: 'Data logger' and 'File' (which is selected). To the right of the radio buttons is a list box containing 'Delimited Text' (highlighted in blue) and 'JSON Time Series'. At the bottom of the dialog, there is a text box with an information icon and the text 'Please refer to our [documentation](#) for help configuring your specific device.' and two buttons: 'Next' and 'Cancel'.

2. Select *Upload via HTTP* as the transport type. A unique URL will be automatically generated per data source. Select the desired timezone and click Next.

Data Source Wizard

1 Source type

2 Configure source

3 Add files

4 Configure parameters

Connection

Transport type

Upload via HTTP

?

URL

https://upload.eagle.io/source/soap-laugh-corn

Authentication

None

?

IP whitelist

eg. 192.168.7.52/32

Add

Remove

?

Collection

☐ Assign as Location coordinate source

?

Time

Timezone

(GMT+10:00) Eastern Time - Melbourne, Sydney

?

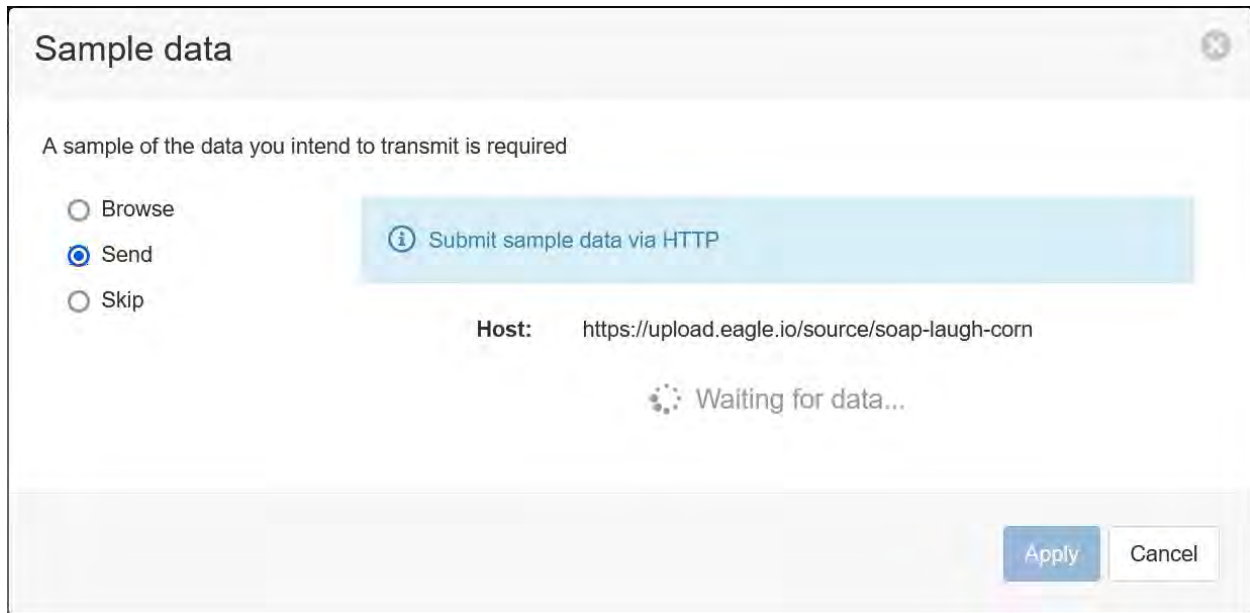
☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous

Next

Cancel

3. Choose the **Send** option:



Sample data

A sample of the data you intend to transmit is required

☐ Browse

☒ Send

☐ Skip

Submit sample data via HTTP

Host: https://upload.eagle.io/source/soap-laugh-corn

Waiting for data...

Apply Cancel

The data source is now ready to receive sample data.

## 2. Define Myriota message format

In order to send data from the Myriota device to eagle.io, the data format must first be defined. This will describe how to decode the message, which comprises a 40-character hexadecimal string. [Ruby unpack syntax](https://apidock.com/ruby/String/unpack) (<https://apidock.com/ruby/String/unpack>) is used to define the format.

This is an example of the packed Myriota string, and the data format which defines how to unpack it:

0f009ab3abef7f383d5bbc2bb460cccccccccccc

Format String char	Data type	Number of hex pairs	Value in demo expression	Decimal Conversion
v	Unsigned short int	2	0f00	15
l	Int (INT32)	4	9ab3abef	-273960038
l	Int (INT32)	4	7f383d5b	1530738815
V	Long int (INT32)	4	Bc2bb460	1622420412
V	Long int (UINT32)	4	cccccccc	3435973836

Note that each of the 5 elements packed into this particular hexadecimal string has a data type which maps to a letter code. These letter codes are an example of the [Ruby unpack syntax](https://apidock.com/ruby/String/unpack) (<https://apidock.com/ruby/String/unpack>), a shorthand way of defining data types.

The 5 elements of this example are defined by the 5 code letters, **v l l V V**. Although all messages will be 40-character hex strings, other messages may be defined by different combinations of data types, which will result in different combinations of the [unpack](https://apidock.com/ruby/String/unpack) (<https://apidock.com/ruby/String/unpack>) letter codes.

The combination of letter codes that define the data format will play an important role in the following step.

### 3. Determine the destination URL

The destination URL that Myriota will send data to takes the following format:

**`https://upload.eagle.io/source/<server key>/myriota?format=<data format>`**

Where:

- **<server key>** is the unique three-word key that is generated as part of the URL during the data source wizard in step 1
- **<data format>** is the combination of letter codes that define the specific data format used to unpack the 40-character hexadecimal message

The first portion of this URL is displayed in the data source creation wizard, which at the end of step 1 was left waiting for data.

This should be appended with `/myriota?format=<data format>` to complete the URL, for example:

**`https://upload.eagle.io/source/soap-laugh-corn/myriota?format=vllVV`**

Note that this specific example uses a server key of **soap-laugh-corn**, and a data format of **vllVV**

#### 4. Send a sample message to the destination URL

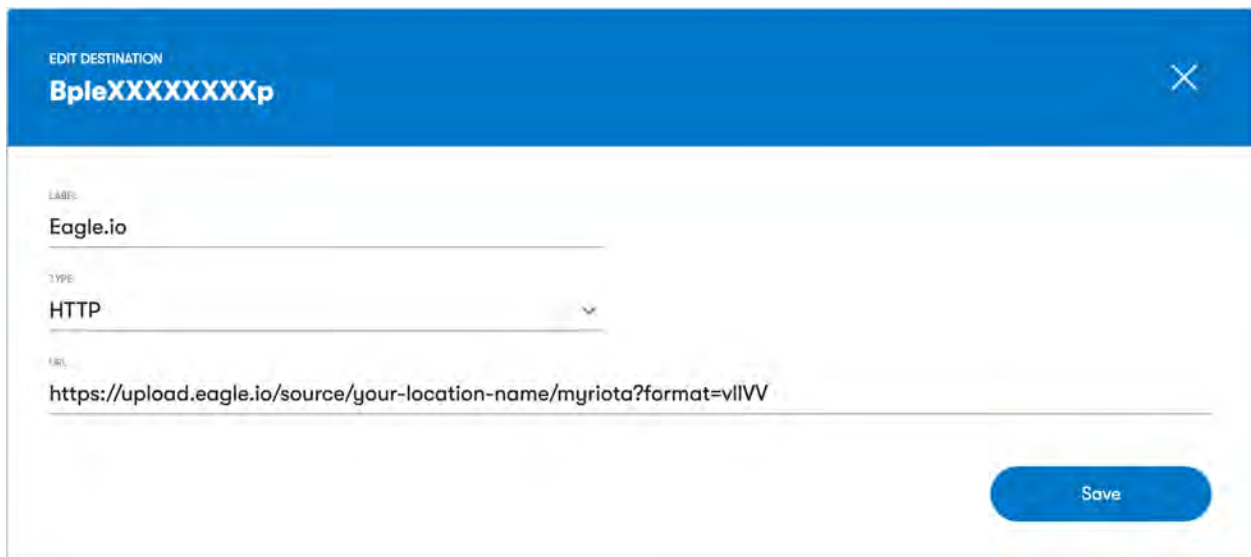
Using the URL determined in the previous step, the Myriota device can now be configured.

---

**Note:** Each eagle.io destination URL is unique to that data source. You will need to create a new data source with a new URL for each new Myriota device, along with with a corresponding new destination in the Myriota Device Manager

---

1. In the **Destinations** section of the Myriota Device Manager, click the **Create** button to create a new destination. Enter a descriptive label, choose **HTTP** as the type, and enter the destination URL from the previous step, then click **Save**:



EDIT DESTINATION

BpleXXXXXXXXp

LABEL

Eagle.io

TYPE

HTTP

URL

https://upload.eagle.io/source/your-location-name/myriota?format=vllVV

Save

2. In the **Devices** section of the Myriota Device Manager, assign a device to the new eagle.io destination:

EDIT DEVICE

00900000b5

LABEL

Myriota SnL

Destination/s

Eagle.io

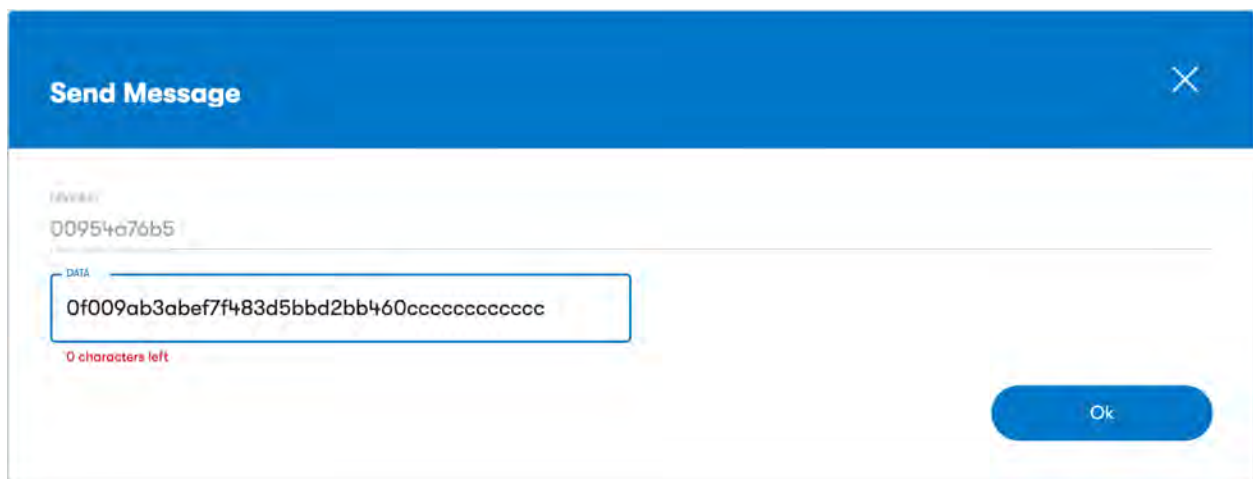
+ Add destination

Save

3. Send a sample message from Myriota by clicking the configured device, and then clicking the **Send**

**message** button.

4. Enter the example data string (a 40-character hex string formatted in [Ruby unpack format](https://apidock.com/ruby/String/unpack) (<https://apidock.com/ruby/String/unpack>)) and click **Ok** to send the data to the eagle.io data source which is waiting to receive it:



**Send Message**

MESSAGE

00954676b5

DATA

0f009ab3abef7f483d5bbd2bb460cccccccccccc

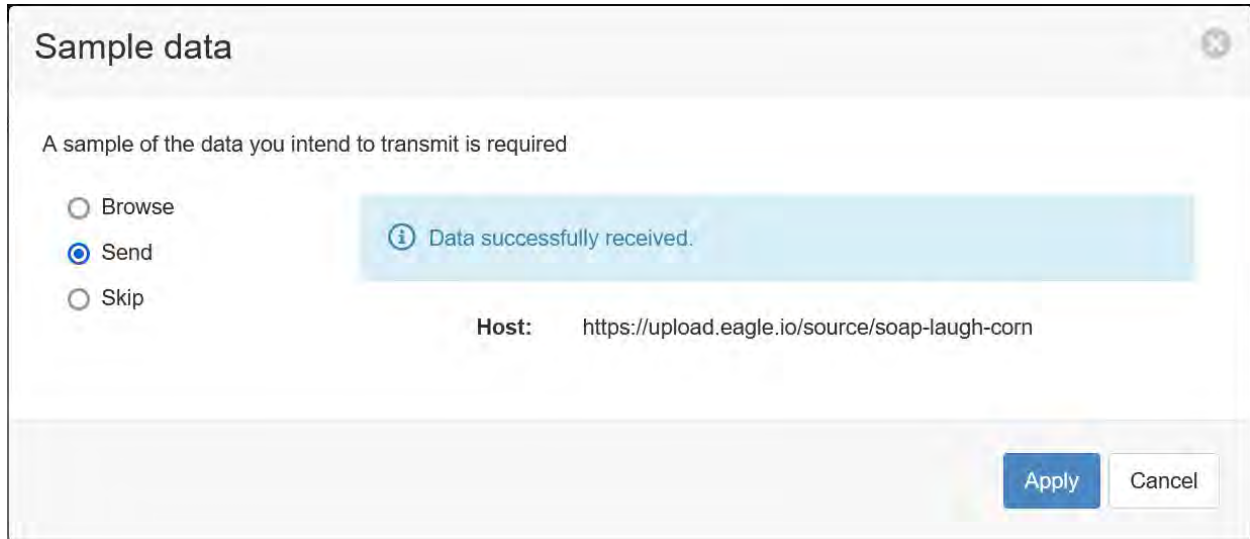
0 characters left

Ok

## 5. Use the sample message to configure the eagle.io parser

Returning to the eagle.io data source wizard, you should now see that data has been received:



A screenshot of a 'Sample data' dialog box. The title bar says 'Sample data' with a close button. Below the title, it says 'A sample of the data you intend to transmit is required'. There are three radio buttons: 'Browse', 'Send' (which is selected), and 'Skip'. To the right of the 'Send' button, there is a light blue notification box with an information icon and the text 'Data successfully received.'. Below this, the 'Host:' is set to 'https://upload.eagle.io/source/soap-laugh-corn'. At the bottom right, there are 'Apply' and 'Cancel' buttons.

Sample data

A sample of the data you intend to transmit is required

☐ Browse

☒ Send

☐ Skip

**Data successfully received.**

Host: https://upload.eagle.io/source/soap-laugh-corn

Apply Cancel

**Note:** If the data source wizard does not say **Data successfully received**, review the actions taken in step 4, and ensure the correct URL was used.

Click the **Apply** button to move to the next step of the data source wizard.

The Parser configuration screen allows the user to describe the structure and format of the message so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details.

The Myriota message will be received as a header row and a data row. The header row (which is called a **Labels row** in the parser configuration) will comprise each letter code used in the format portion of the URL. The number of columns defined will be the same as the number of letters in the format code. In the current example the format is **vllVV**, therefore 5 columns will be defined with header values of **v**, **l**, **l**, **V**, **V**

Parser configuration Sensors

Column delimiter	Format	Labels row	Units row	Decimal separator	Text qualifier
...	Column series	None	None	Period	"
Text encoding	Match	Success required			
UTF-8 (default)	Smart	Optional	Reset to defaults		

Column 1	Column 2	Column 3	Column 4	Column 5
Record Time	Number	Number	Time	Number
15	-273960038	1530738815	1622420412	3435973836

Get help with parser configuration

Apply Cancel

Configure the **Labels row** to be row 1.

The data in each column will be different when different formats are defined. The parser must be configured so that each portion of the message matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*. The specific message in this example defines a record timestamp in column 4, with all other values being numbers. Therefore in the parser configuration, the column data type of column 1 should be changed to **Number**, and the column data type of column 4 should be changed to **Record Time**. The parser configuration should now look like this:

Parser configuration Sensors

Column delimiter	Format	Labels row	Units row	Decimal separator	Text qualifier
...	Column series	1	None	Period	"
Text encoding	Match	Success required			
UTF-8 (default)	Smart	Minimum 1 col			
Reset to defaults					

	V	I	I	V	V
	Number	Number	Number	Record Time	Number
LABELS	V	I	I	V	V
	15	-273960038	1530738815	1622420412	3435973836

Get help with parser configuration

Apply Cancel

Additionally, the **Record Time** column will need to be configured to match the timestamp format. Click the cog icon next to column 4 to configure the record time format, and change the *Format* field to use the *time format* (page 583) **X**, which defines a timestamp format of seconds since UNIX epoch:

Parser configuration Sensors

Column delimiter:  Format:  Labels row:  Units row:  Decimal separator:  Text qualifier:

Text encoding:  Match:  Success required:  [Reset to defaults](#)

Column joins and configuration (V) [Close](#)

Column joins	Field	Format	Format filter
V	<input type="text" value="Time"/>	<input type="text" value="X"/>	
V	<input type="text" value="Quality"/>	<input type="text" value="X"/>	<input type="text" value="Include record"/> <a href="#">Add join</a>

V

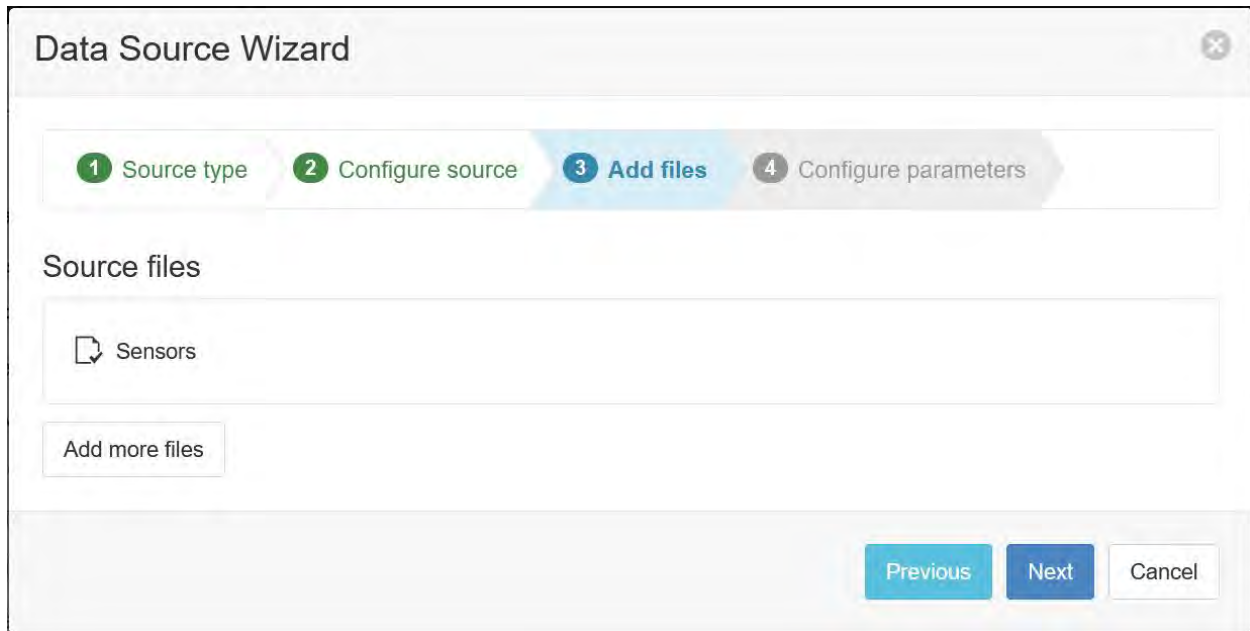
#  [Settings](#)

LABELS V

1622420412

[Get help with parser configuration](#) [Apply](#) [Cancel](#)

Click **Close** to return to the main parser, and click **Apply** to exit the parser. Click **Next** to move to step 4 in the data source wizard:



The image shows a 'Data Source Wizard' dialog box with a close button in the top right corner. It features a progress bar with four steps: 1 Source type, 2 Configure source, 3 Add files (highlighted in blue), and 4 Configure parameters. Below the progress bar, the 'Source files' section contains a text input field with a folder icon and the text 'Sensors'. Below this field is a button labeled 'Add more files'. At the bottom right of the dialog are three buttons: 'Previous' (light blue), 'Next' (dark blue), and 'Cancel' (light gray).

By default, the 4 parameter names will be based on the letters in the header row:

Data Source Wizard

1 Source type

2 Configure source

3 Add files

4 Configure parameters

Location coordinates

⊗ None

?

Sensors

Series

Parameter assignment

Units

☒

$\text{J}_\text{v}$  v

New

Existing

v

☒

$\text{J}_\text{I}$  I

New

Existing

I

☒

$\text{J}_\text{I}$  I

New

Existing

I (2)

☒

$\text{J}_\text{V}$  V

New

Existing

V

Previous

Finish

Cancel

Change the defaults to the desired the parameter names (and optionally units) for each column defined in the parser configuration. Uncheck any series you do not want to create as parameters:

358

Chapter 2. Using eagle.io

Data Source Wizard

1 Source type
2 Configure source
3 Add files
4 Configure parameters

Location coordinates
⊗ None

Sensors

Series	Parameter assignment		Units
<input checked="" type="checkbox"/> $\text{J}_\text{v}$	New	Existing	sequence
<input checked="" type="checkbox"/> $\text{J}_\text{l}$	New	Existing	lat
<input checked="" type="checkbox"/> $\text{J}_\text{l}$	New	Existing	long
<input checked="" type="checkbox"/> $\text{J}_\text{V}$	New	Existing	current

Previous
Finish
Cancel

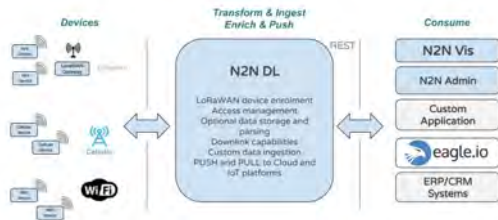
Click Finish to complete the setup.

**Note:** You can reconfigure the parser configuration and parameter assignment from the [Data Source](#) (page 114) properties dialog after creation.

**Note:** If the Myriota message includes latitude and longitude, this can be configured to update a location map marker within eagle.io by using this [process](https://help.eagle.io/hc/en-au/articles/4404653973273-Using-coordinates-from-a-data-source-in-a-location) (<https://help.eagle.io/hc/en-au/articles/4404653973273-Using-coordinates-from-a-data-source-in-a-location>).

### 2.14.10 NNNco

The **N2N DL platform** (<https://www.nnnco.io/n2n/v3/docs/>) is designed to offer programmatic access to a multitude of IoT devices from multiple networks (LoRaWAN / Cellular & Internet enabled). This platform allows **eagle.io** (<https://eagle.io>) to use the *Download via HTTP* (page 116) transport to acquire the data.



### Getting an API key for access

An NNNco API key will be required for eagle.io to access data on your NNNco devices. API keys can be obtained from NNNco by getting in touch with one of their sales reps or emailing [n2n-dl-support@nnnco.com.au](mailto:n2n-dl-support@nnnco.com.au).

Read more about [getting API access](https://www.nnnco.io/n2n/v3/docs/getting-started/#getting-access) (<https://www.nnnco.io/n2n/v3/docs/getting-started/#getting-access>).

### Getting a device ID

A device ID will be required for each data source you create in eagle.io. Read more about [getting a list of devices that belong to your accounts](https://www.nnnco.io/n2n/v3/docs/getting-started/#get-a-list-of-devices-that-belong-to-your-account) (<https://www.nnnco.io/n2n/v3/docs/getting-started/#get-a-list-of-devices-that-belong-to-your-account>).

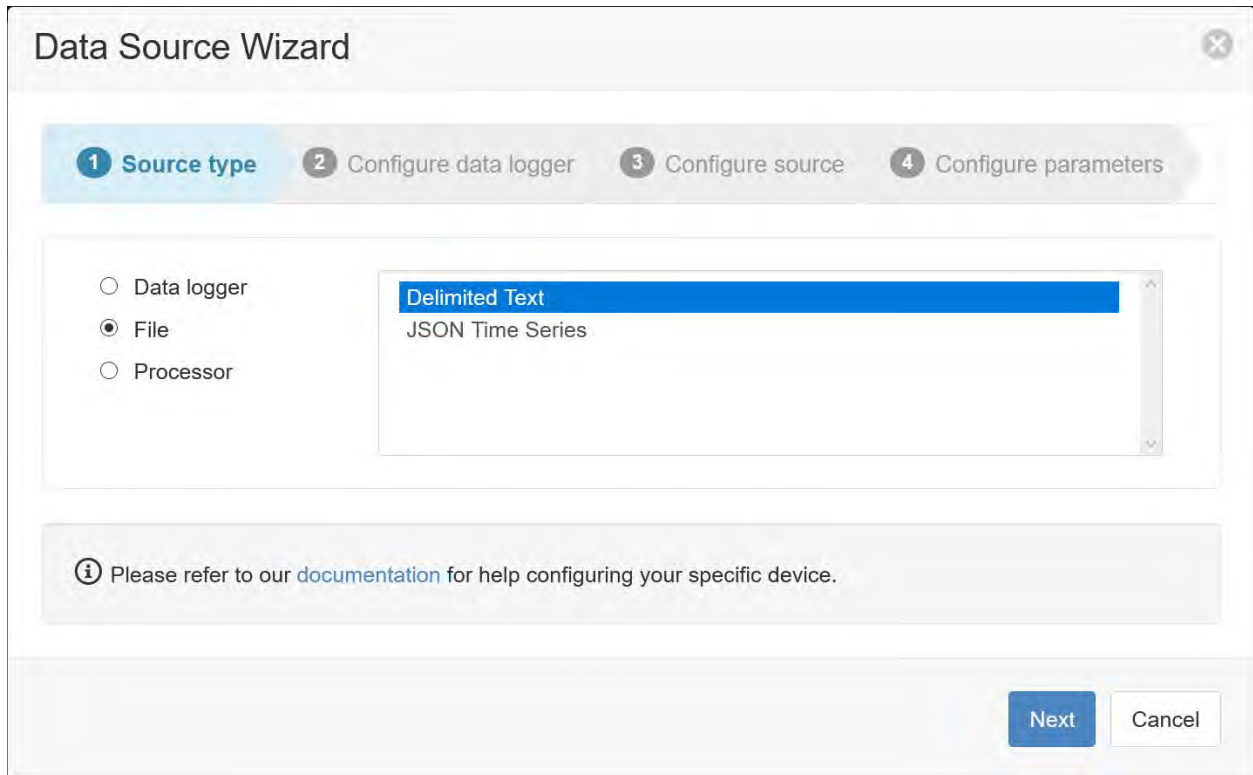
Once you have your API key and at least one device ID, you are ready to create a new data source in eagle.io.

The following documentation and images will use an example NNNco API key of `8a0892e3aedbe2f2ade4bd727d4725e0c639d021` and an example NNNco device ID of `393435376A377906`.

### Source configuration

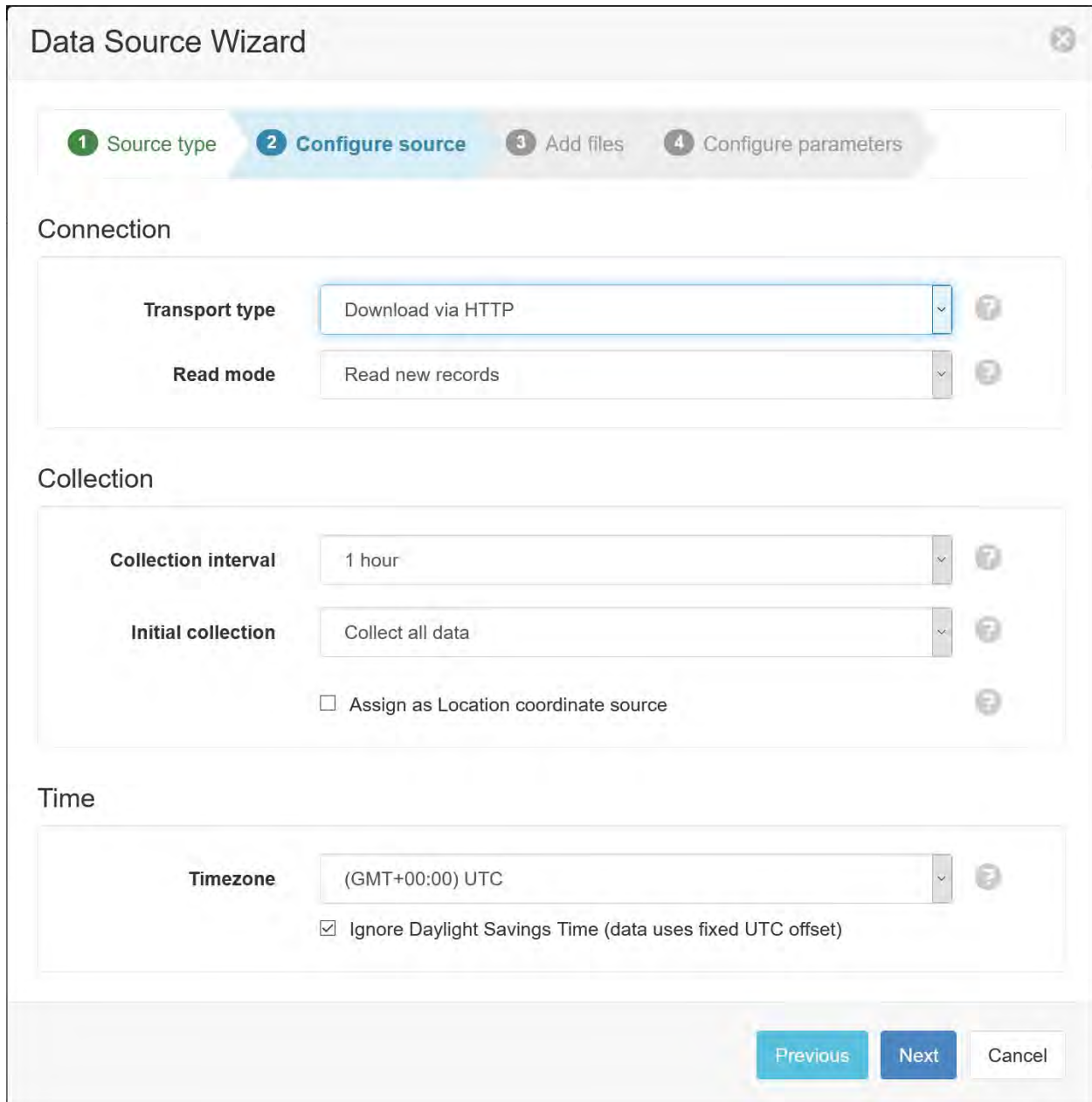
1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.





The image shows a 'Data Source Wizard' dialog box with a close button in the top right corner. It features a progress bar with four steps: 1. Source type (active), 2. Configure data logger, 3. Configure source, and 4. Configure parameters. Under step 1, there are three radio button options: 'Data logger', 'File' (which is selected), and 'Processor'. To the right of these options is a list box containing 'Delimited Text' (highlighted in blue) and 'JSON Time Series'. Below the list box is an information message: 'Please refer to our [documentation](#) for help configuring your specific device.' At the bottom right, there are 'Next' and 'Cancel' buttons.

2. Select *Download via HTTP* as the transport type. The read mode should be left as **Read new records**. Choose a desired collection interval, for example 1 hour. The timezone of the data source should be set to **(GMT+00:00) UTC**, then click Next.



**Data Source Wizard**

1 Source type   2 **Configure source**   3 Add files   4 Configure parameters

**Connection**

Transport type: Download via HTTP

Read mode: Read new records

**Collection**

Collection interval: 1 hour

Initial collection: Collect all data

☐ Assign as Location coordinate source

**Time**

Timezone: (GMT+00:00) UTC

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous   Next   Cancel

3. Enter the URL of the device in the following format, where **<key>** is your API key and **<deviceId>** is your device ID:

```
https://api:<key>@www.nnnco.io/v3/api/core/devices/<deviceId>/readings/export
```

For example, if your API key is **8a0892e3aedbe2f2ade4bd727d4725e0c639d021** and your device ID is **393435376A377906**, then the URL would be:

```
https://api:8a0892e3aedbe2f2ade4bd727d4725e0c639d021@www.nnnco.io/v3/api/core/devices/393435376A377906/readings/export
```

After entering the URL in the wizard, click Apply.

Sample data

Uri

https://api:8a0892e3aedbe2f2ade4bd727d4725e0c639d021@wv

Reset

Apply

Cancel

4. The Parser configuration screen allows the user to describe the structure and format of the sample data file so it can be correctly interpreted by the system during subsequent data collection. Refer to [Text Parser Configuration](#) (page 123) for full details. First you will need to change the labels row to be 1.

Parser configuration <https://api.8a0892e3aedbe2f2ade4bd727d4725e0c639d021@www.nnnco.io/v3/api/core/devices/393435376A377908/readings/export>

Column delimiter:  Format:  Labels row:  Units row:  Decimal separator:  Text qualifier:  Text encoding:  Match:  Success required:  [Reset to defaults](#)

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
<input type="text" value="time"/>	<input type="text" value="temp (C)"/>	<input type="text" value="humid (%)"/>	<input type="text" value="ch-1 (v)"/>	<input type="text" value="ch-2 (v)"/>	<input type="text" value="ch-3 (v)"/>	<input type="text" value="ch-4 (v)"/>	<input type="text" value="ch-5 (v)"/>
1	2021-05-17T01:51:44.272Z	19.48	66.75	300	21	5.75	3.52
2	2021-05-17T01:56:49.242Z	19.51	66.75	300	21	5.75	3.52
3	2021-05-17T02:01:52.962Z	19.53	66.75	300	21	5.75	3.52
4	2021-05-17T02:06:52.527Z	19.54	66.75	300	21	5.75	3.52
5	2021-05-17T02:11:51.112Z	19.56	66.75	300	21	5.75	3.52
6	2021-05-17T02:16:50.679Z	19.58	67.24	300	20	63	3.51
7	2021-05-17T02:21:56.315Z	19.6	67.24	300	20	63	3.52
8	2021-05-17T02:26:54.270Z	19.62	67.24	300	20	63	3.62
9	2021-05-17T02:31:59.324Z	19.63	67.24	300	20	63	3.52
10	2021-05-17T02:37:02.530Z	19.68	67.24	300	20	63	3.52
11	2021-05-17T02:42:06.180Z	19.7	67.24	300	20	63	3.52

You will need to set the correct *time format* (page 583) by clicking the gear icon in Column 1 (Record time).

Parser configuration <https://api.8a0892e3aedbe2f2ade4bd727d4725e0c639d021@www.nnnco.io/v3/api/core/devices/393435376A377906/readings/export>

Column delimiter: Columns series Labels row: 1 Units row: None Decimal separator: Period Text qualifier: Text encoding: UTF-8 (default) Match: Smart Success required: ☐ Reset to defaults

time	status digital	ch-1 temperature(°C)	ch-2 humidity(%)	ch-3 time(s)	ch-4 lora-rssi	ch-5 lora-snr	battery voltage(V)
Record Time	Number	Number	Number	Number	Number	Number	Number
2021-03-17T01:51:44.272Z	1	19.49	66.75	300	-21	5.75	3.52
2021-03-17T01:59:45.341Z	1	19.51	66.75	300	-21	5.75	3.52
2021-03-17T02:01:52.461Z	1	19.53	66.75	300	-21	5.75	3.52
2021-03-17T02:08:52.527Z	1	19.54	66.75	300	-21	5.75	3.52
2021-03-17T02:11:51.111Z	1	19.56	66.75	300	-21	5.75	3.52
2021-03-17T02:16:50.679Z	1	19.58	67.24	300	-20	63	3.51
2021-03-17T02:21:56.915Z	1	19.6	67.24	300	-20	63	3.52
2021-03-17T02:26:54.279Z	1	19.62	67.24	300	-20	63	3.52
2021-03-17T02:31:59.924Z	1	19.63	67.24	300	-20	63	3.52
2021-03-17T02:37:02.530Z	1	19.68	67.24	300	-20	63	3.52
2021-03-17T02:42:06.186Z	1	19.7	67.24	300	-20	63	3.52

Get help with parser configuration Apply Cancel

Set the time format to be **YYYY-MM-DDTHH:mm:ss.SSSZ** then click Close twice.

Parser configuration <https://api.8a0892e3aedbe2f2ade4bd727d4725e0c639d021@www.nnnco.io/v3/api/core/devices/393435376A377906/readings/export>

Column delimiter: Columns series Labels row: 1 Units row: None Decimal separator: Period Text qualifier: Text encoding: UTF-8 (default) Match: Smart Success required: ☐ Reset to defaults

time

Record Time

time

2021-03-17T01:51:44.272Z

2021-03-17T01:56:45.241Z

2021-03-17T02:01:52.461Z

2021-03-17T02:06:52.527Z

2021-03-17T02:11:51.111Z

2021-03-17T02:16:50.679Z

2021-03-17T02:21:56.915Z

2021-03-17T02:26:54.279Z

2021-03-17T02:31:59.924Z

2021-03-17T02:37:02.530Z

2021-03-17T02:42:06.186Z

Column joins and configuration

Column joins

Field

Format

Format filter

time

Time

YYYY-MM-DDTHH:mm:ss.SSSZ

time

Quality

Include record

Add join

Get help with parser configuration Apply Cancel

Ensure the data in each column matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*. In the following example, the labels row has now been set to 1 (row background is purple), the record time has been set to the correct format (column background is not red), and the 7 data columns are all of type Number. Click Apply to close the parser and continue.

Parser configuration <https://api.8a0892e3aedbe2f2ade4bd727d4725e0c639d021@www.nnnco.io/v3/api/core/devices/393435376A377906/readings/export>

Column delimiter:  Format:  Labels row:  Units row:  Decimal separator:  Text qualifier:  Text encoding:  Match:  Success required:  [Reset to defaults](#)

	time	status digital	ch-1 temperature(°C)	ch-2 humidity(%)	ch-3 time(s)	ch-4 lora-rssi	ch-5 lora-snr	battery voltage(V)
	Record Time	Number	Number	Number	Number	Number	Number	Number
1	2021-03-17T01:51:44.272Z	1	19.48	66.75	300	-21	5.75	3.52
2	2021-03-17T01:56:45.241Z	1	19.51	66.75	300	-21	5.75	3.52
3	2021-03-17T02:01:52.461Z	1	19.53	66.75	300	-21	5.75	3.52
4	2021-03-17T02:06:52.327Z	1	19.54	66.75	300	-21	5.75	3.52
5	2021-03-17T02:11:51.114Z	1	19.56	66.75	300	-21	5.75	3.52
6	2021-03-17T02:16:50.679Z	1	19.59	67.24	300	-20	5.75	3.51
7	2021-03-17T02:21:56.015Z	1	19.6	67.24	300	-20	5.75	3.52
8	2021-03-17T02:26:54.279Z	1	19.62	67.24	300	-20	5.75	3.52
9	2021-03-17T02:31:59.824Z	1	19.63	67.24	300	-20	5.75	3.52
10	2021-03-17T02:37:02.539Z	1	19.66	67.24	300	-20	5.75	3.52
11	2021-03-17T02:42:06.196Z	1	19.7	67.24	300	-20	5.75	3.52


[Get JSON with parser configuration](#) [Apply](#) [Cancel](#)

5. Click Next to continue.

## Data Source Wizard

1 Source type 2 Configure source 3 Add files 4 Configure parameters

### Source files

 <https://api.8a0892e3aedbe2f2ade4bd727d4725e0c639d021@www.nnnco.io/v3/api/core/devices/393435376A377906/readings/export>

[Add more files](#)

[Previous](#) [Next](#) [Cancel](#)








6. Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.

Data Source Wizard

1 Source type
2 Configure source
3 Add files
4 **Configure parameters**

Location coordinates
⊗ None

https://api:8a0892e3aedbe2f2ade4bd727d4725e0c639d021@www.nnnco.io/v3/api/core/devices/39343537f

Series	Parameter assignment			Units
<input checked="" type="checkbox"/>  status digital	New	Existing	status digital	
<input checked="" type="checkbox"/>  ch-1 temperature(°C)	New	Existing	ch-1 temperature(°C)	
<input checked="" type="checkbox"/>  ch-2 humidity(%)	New	Existing	ch-2 humidity(%)	
<input checked="" type="checkbox"/>  ch-3 time(s)	New	Existing	ch-3 time(s)	
<input checked="" type="checkbox"/>  ch-4 lora-rssi	New	Existing	ch-4 lora-rssi	
<input checked="" type="checkbox"/>  ch-5 lora-snr	New	Existing	ch-5 lora-snr	
<input checked="" type="checkbox"/>  battery voltage(V)	New	Existing	battery voltage(V)	

Previous
Finish
Cancel

**Note:** You can reconfigure the device URL, parser configuration and parameter assignment from the [Data Source](#) (page 114) properties dialog after creation.

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.



### 2.14.11 OTT netDL

The [OTT](http://ott.com) (<http://ott.com>) **netDL 500** and **netDL 1000** data loggers can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.



#### Source configuration

1. Create a [Data Source](#) (page 114) and select *Data logger* as the Source Type. Choose *OTT netDL* from the list and click Next.

### Data Source Wizard

1 Source type

2 Configure data logger

3 Configure source

4 Configure parameters

☒ Data logger  
☐ File

Campbell Scientific [PakBus]  
**OTT netDL**  
SensorLog  
WaterLog

Please refer to our [documentation](#) for help configuring your specific device.

Next

Cancel

2. Select the data logger Model and specify the number of channels/sensors configured in the logger (this can be changed later). Click Next.

The screenshot shows a 'Data Source Wizard' window with a progress bar at the top indicating four steps: 1. Source type, 2. Configure data logger (current step), 3. Configure source, and 4. Configure parameters. The main area contains two sections. The first section has two dropdown menus: 'Model' set to 'netDL 500/1000' and 'Number of channels' set to '3'. The second section contains four rows of settings: 'Data format' is 'CSV', 'Compression' is 'None', 'Field separator' is 'Comma', and 'Decimal separator' is 'Period (.)'. At the bottom right, there are three buttons: 'Previous' (disabled), 'Next' (active), and 'Cancel'.

Step	Label
1	Source type
2	Configure data logger
3	Configure source
4	Configure parameters

Model	netDL 500/1000
Number of channels	3

Data format	CSV
Compression	None
Field separator	Comma
Decimal separator	Period (.)

PreviousNextCancel

2. Select *Upload data to ftp.eagle.io* as the transport type. A unique *Ftp user* account will be automatically generated per Data Source. You can optionally specify a password to be used by the logger for additional security. Select the timezone of the logger's clock and click Next.



**Data Source Wizard**

1 Source type   2 Configure data logger   **3 Configure source**   4 Configure parameters

**Connection**

Transport type: Upload to ftp.eagle.io

FTP user: steam-home-music   [Change](#)

FTP password: optional   ...

**Time**

Timezone: (GMT+10:00) Eastern Time - Melbourne, Sydney

[Previous](#)   [Next](#)   [Cancel](#)

4. Specify the channel Id and name of each parameter (with optional units) you would like to collect from the logger. Uncheck any series you do not want to create as parameters. In the example below, the logger will have channel '0810' configured as 'U Supply'. Click Finish to complete the setup.

**Data Source Wizard**

1 Source type 2 Configure data logger 3 Configure source 4 **Configure parameters**

Location coordinates: ⊗ None

**Sensors**

<input checked="" type="checkbox"/>	Series	Parameter assignment		Units	
<input checked="" type="checkbox"/>	0810	New	Existing	U Supply	V
<input checked="" type="checkbox"/>	0002	New	Existing	Hydrolab Temperature	°C
<input checked="" type="checkbox"/>	0003	New	Existing	pH	

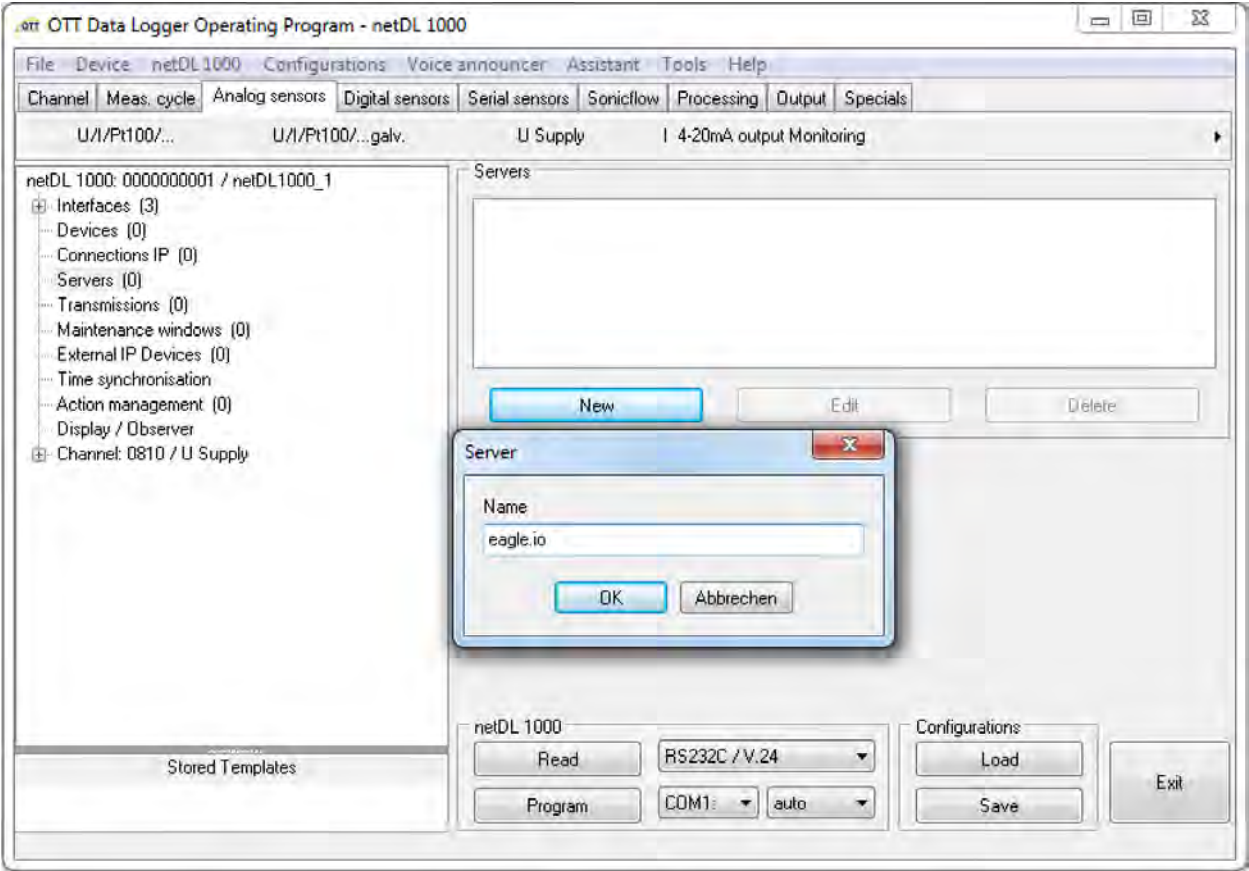
Previous Finish Cancel

**Note:** You can reconfigure the series and parameter assignment from the *Data Source* (page 134) properties dialog after creation.

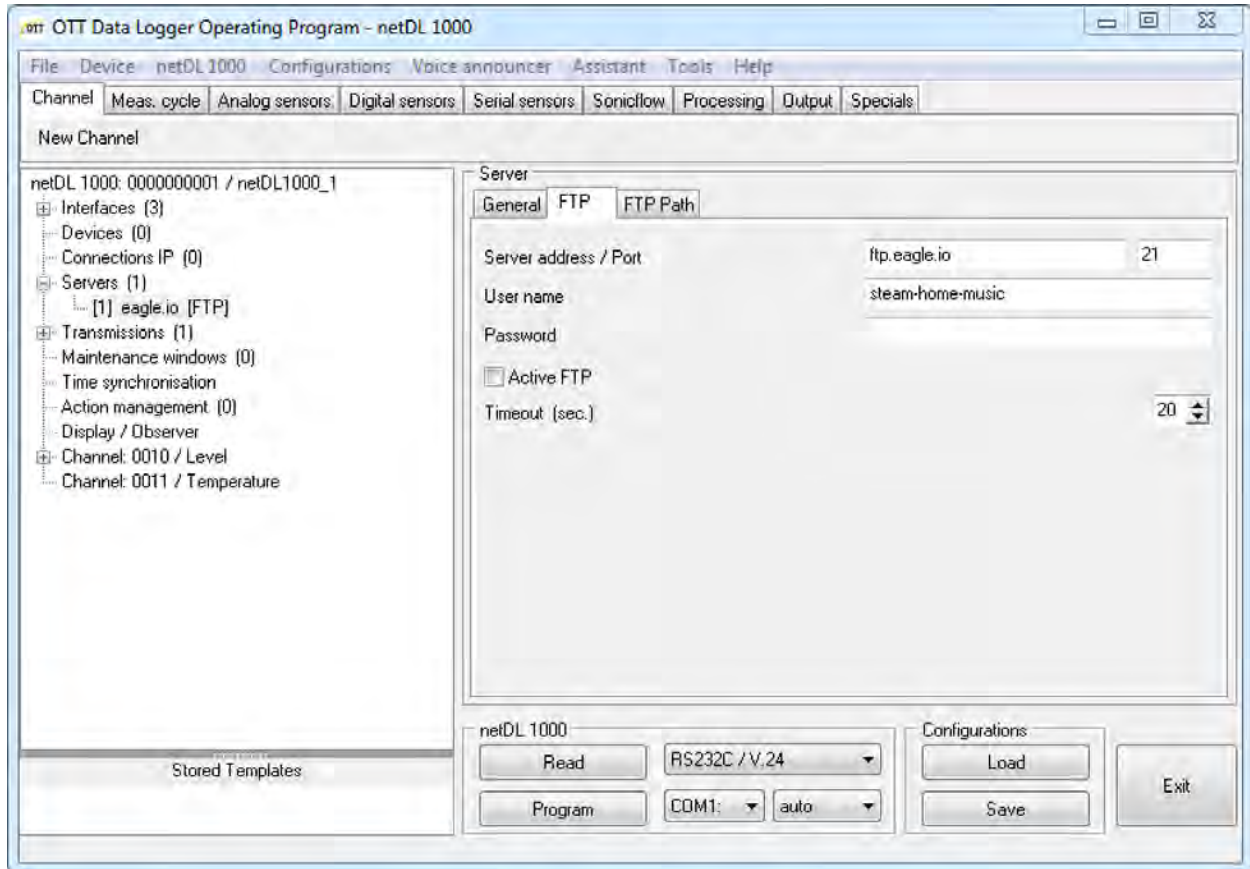
## Device settings

Use the OTT Data Logger Operating Program to configure the netDL logger for communication with eagle.io. Refer to your OTT netDL [user manual](http://www.ott.com) (<http://www.ott.com>) for connection instructions.

1. Create a new *Server* and specify the *Type* as FTP.



2. Change to the FTP tab.



The following FTP settings should be assigned:

**Server address** ftp.eagle.io

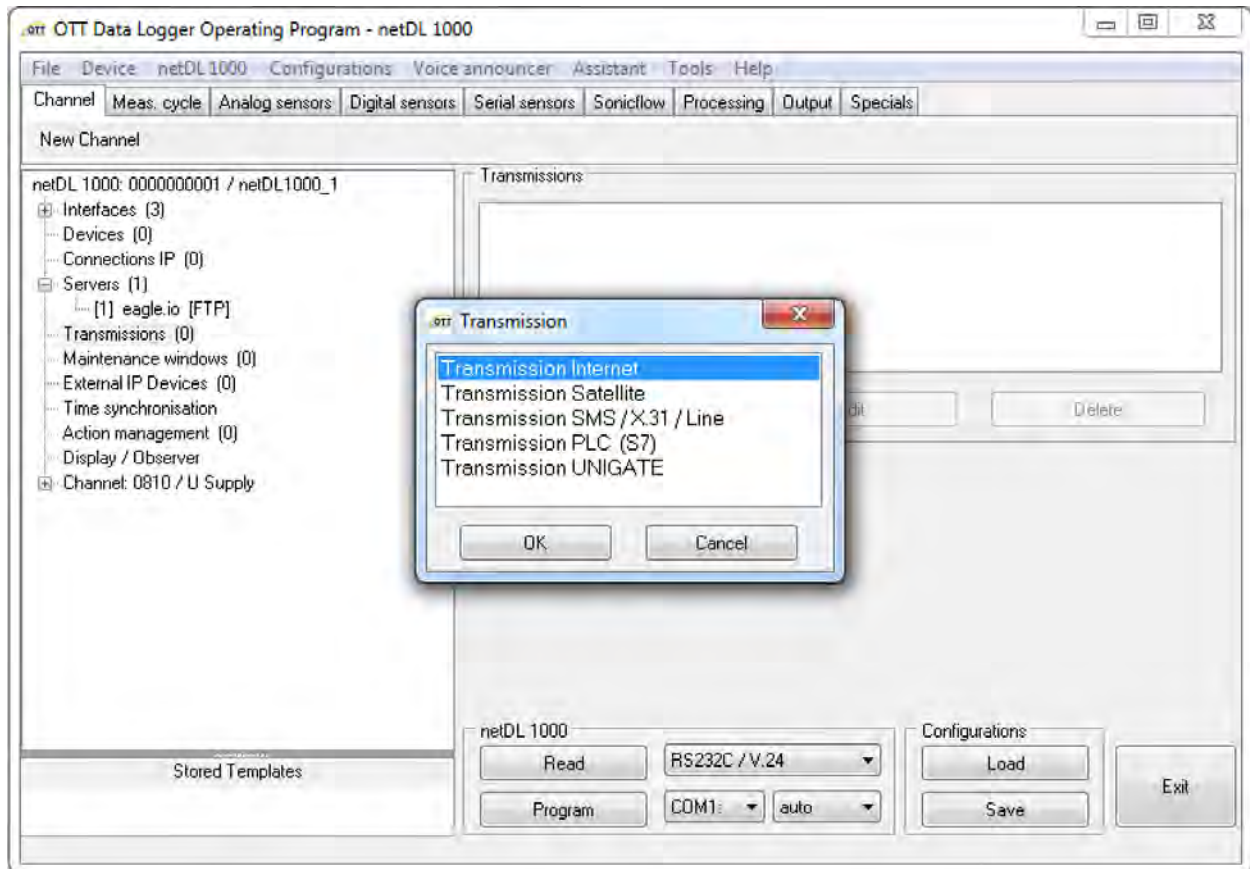
**Port** 21

**User name** Use the unique *FTP user* generated during Data Source creation.

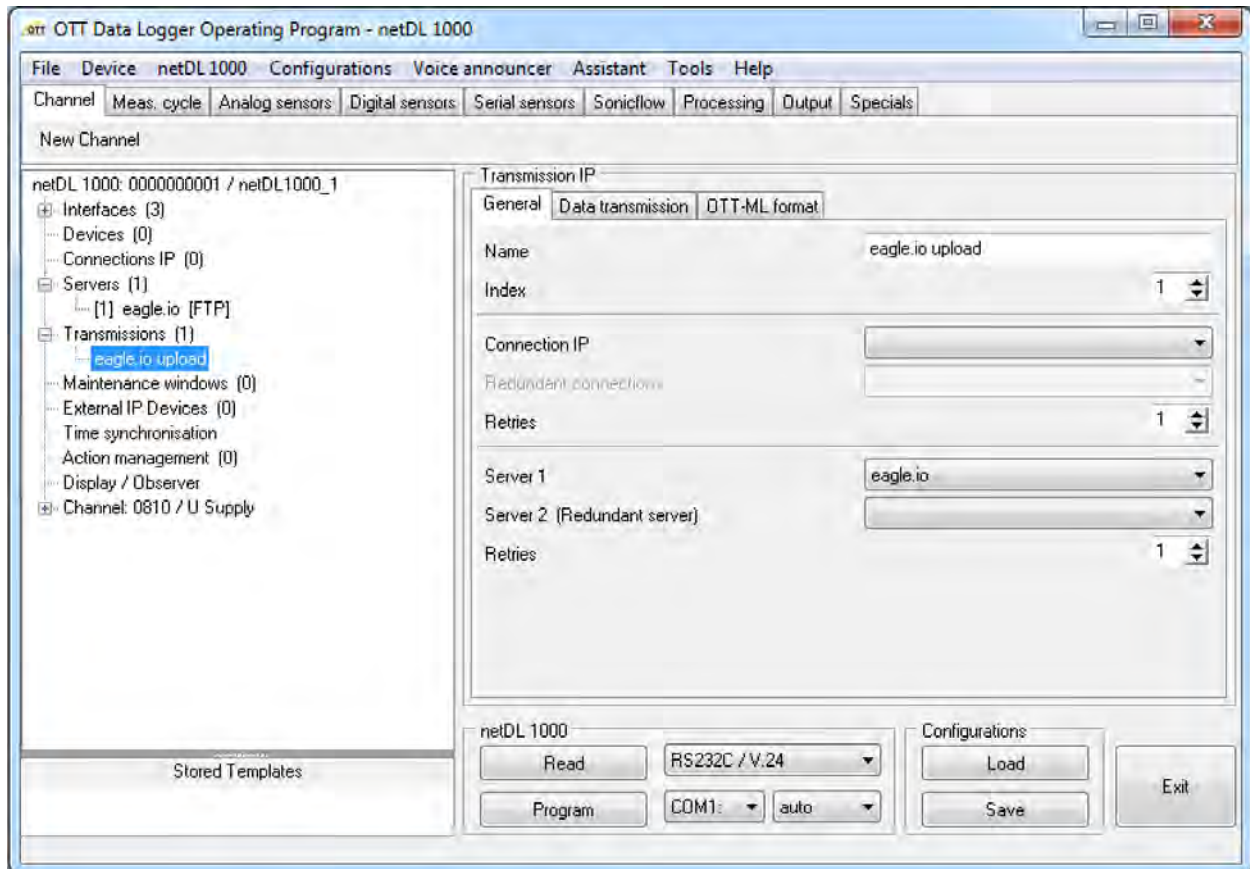
**Password** Use the password you assigned during Data Source creation (or leave blank for none).

**FTP Path** Leave blank

3. Create a new *Transmission* and select *Transmission Internet* as the type.

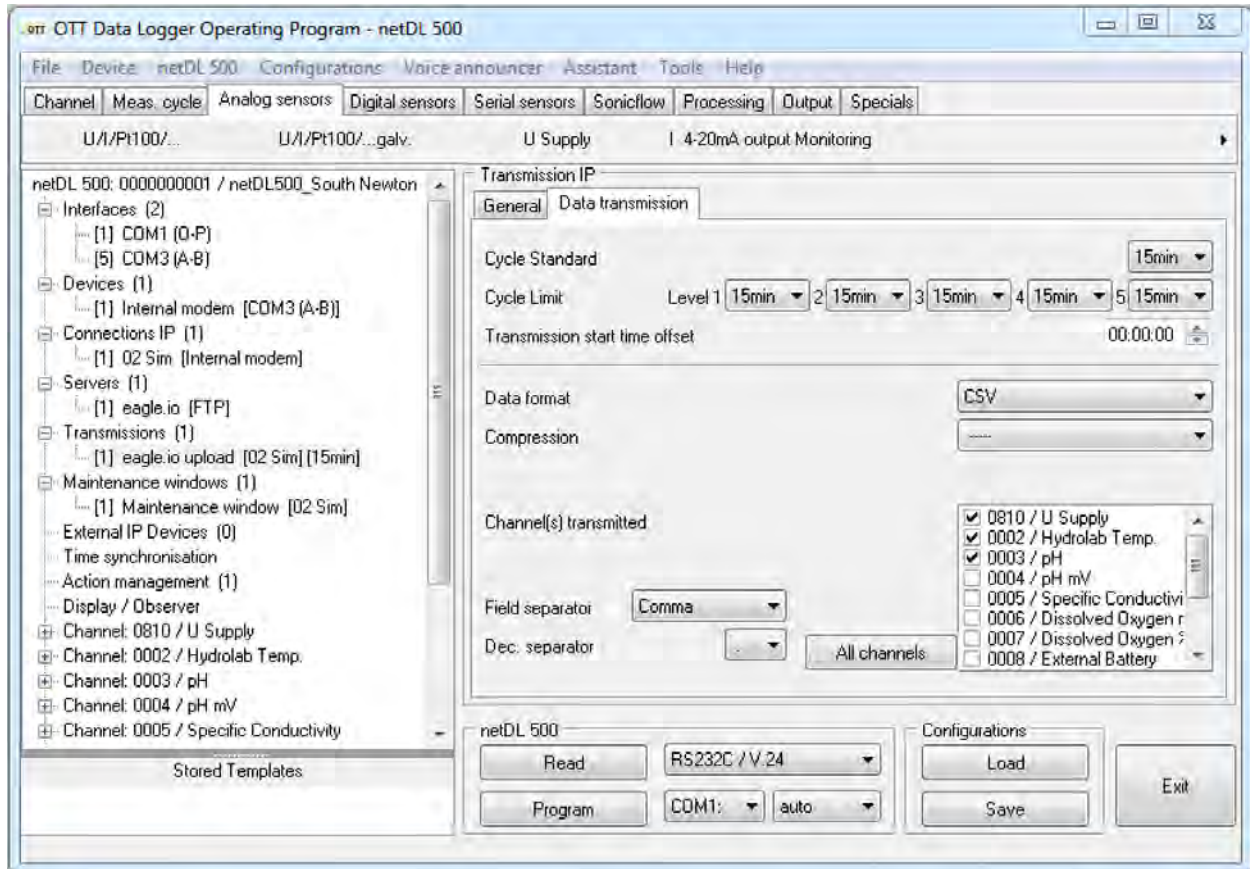


4. Configure the *Server 1* option to use the newly created Server. ie. eagle.io



5. Change to the *Data Transmission* tab and specify the *Cycle Standard* which sets the frequency of uploads (eg. 15min). Set the *Data format* to *CSV* with no compression. Select *Comma* as the *Field separator* and *Period (.)* as the *Decimal separator*.





6. Tick all the channels you would like to transmit to eagle.io (or select the *All channels* option) and save the configuration to the data logger.

**Note:** Use the *Events* (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

### 2.14.12 Point Orange 3G

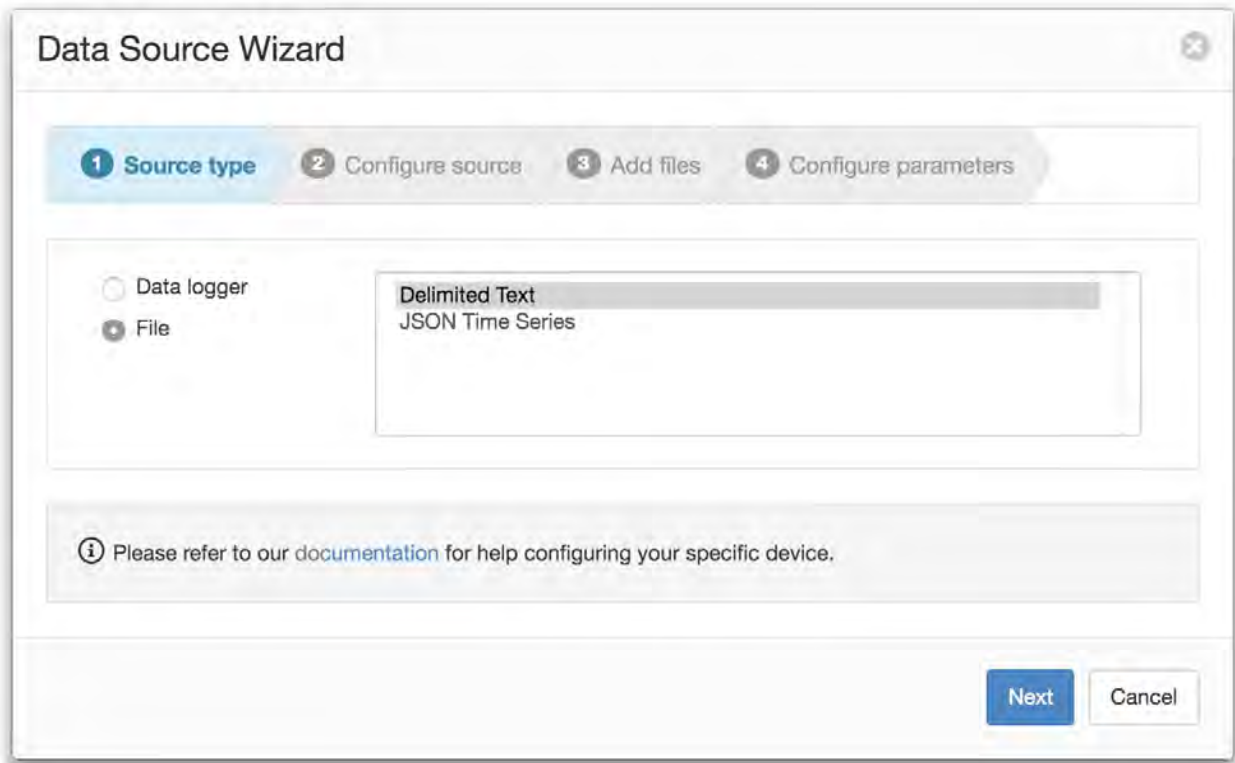
The *Metasphere Point Orange 3G* (<https://www.metasphere.co.uk/products/battery-powered/point-orange-3g-comms/>) RTU outstation can be configured to send data to *eagle.io* (<https://eagle.io>) in a few easy steps.



### Source configuration

1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.





The image shows a 'Data Source Wizard' dialog box with a close button in the top right corner. It features a progress bar with four steps: 1. Source type (highlighted in blue), 2. Configure source, 3. Add files, and 4. Configure parameters. Under the 'Source type' step, there are two radio buttons: 'Data logger' (unselected) and 'File' (selected). To the right of these buttons is a list box containing 'Delimited Text' (highlighted) and 'JSON Time Series'. Below the list box is a light gray informational box with an 'i' icon and the text: 'Please refer to our [documentation](#) for help configuring your specific device.' At the bottom right, there are 'Next' and 'Cancel' buttons.

2. Select *Upload to ftp.eagle.io* as the transport type. A unique *Ftp user* account will be automatically generated per Data Source. You can optionally specify a password to be used by the logger for additional security. Select the timezone of the logger's clock and click Next.

The image shows a 'Data Source Wizard' window with four steps: 1. Source type, 2. Configure source (active), 3. Add files, and 4. Configure parameters. The 'Configure source' step is divided into three sections: Connection, Collection, and Time. The Connection section has fields for Transport type (Upload to ftp.eagle.io), FTP user (leaf-yak-smash), and FTP password (masked with dots). The Collection section has a checkbox for 'Assign as Location coordinate source'. The Time section has a Timezone dropdown (GMT+10:00 Eastern Time - Melbourne, Sydney) and a checked checkbox for 'Ignore Daylight Savings Time (data uses fixed UTC offset)'. At the bottom are buttons for Previous, Next, and Cancel.

**Data Source Wizard**

1 Source type 2 **Configure source** 3 Add files 4 Configure parameters

**Connection**

Transport type Upload to ftp.eagle.io

FTP user leaf-yak-smash Change

FTP password .....

**Collection**

☐ Assign as Location coordinate source

**Time**

Timezone (GMT+10:00) Eastern Time - Melbourne, Sydney

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous Next Cancel

3. Specify `.csv` as the *File name match*. This will match all incoming files for this Source containing `.csv` in the filename. To continue the setup you will need a sample copy of the logger's output data file(s).

**Add files**

**Sample file**  Orange\_T\_20161012.csv ?

☐ Skip or Reset sample file and retrieve on next acquisition

**File name match** .csv Reset ?

☐ Match using regular expression

**Note:** If you do not have a sample available, click the *Skip* checkbox and click finish on the next screen. A sample will be saved automatically on the initial upload from the data logger and you will need to open the [Data Source](#) (page 134) properties dialog and [configure the series](#) (page 119) to create parameters.

4. The Parser configuration screen allows the user to describe the structure and format of the data file so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details.

Parser configuration .csv

Column delimiter: | Labels row: 1 Units row: None Text qualifier: " Text encoding: UTF-8 (default) Match: Smart Reset to defaults

# Date & Time AI05 (C) AI06 (v) AI20 Column 5

Record Time % Number % Number % Number % Disabled

LABELS	Date & Time	AI05 (C)	AI06 (v)	AI20	Column 5
2	2016/10/12 05:30:00	27.369995	6.746336	3.022017	
3	2016/10/12 05:40:00	27.353882	6.773622	3.028189	
4	2016/10/12 05:50:00	27.350133	6.779901	3.031801	

Get help with parser configuration Apply Cancel

If the file contains a header row, set the *Labels row* control to the corresponding row number which will assign default labels for parameter creation.

Ensure the data in each column matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*.

You will need to change the default time format for the *Record Time* column. Click the configure button above the column and change the *Format* field to use the *time format* (page 583) **YYYY/MM/DD HH:mm:ss**. Click Apply and Next.

Parser configuration .csv

Column delimiter: , Labels row: 1 Units row: None Text qualifier: " Text encoding: UTF-8 (default) Match: Smart Reset to defaults

**Date & Time**

Record Time

**Column joins and configuration (Data & Time)**

Column joins	Field	Format	Format filter
Date & Time	Time	YYYY/MM/DD HH:mm:ss	
Date & Time	★ Quality	format	Include record Add join

Get help with parser configuration

Apply Cancel

- Click next to proceed to parameter assignment.

Data Source Wizard

1 Source type 2 Configure source 3 Add files 4 Configure parameters

Source files

.csv

Add more files

Previous Next Cancel

- Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.

**Data Source Wizard**

1 Source type 2 Configure source 3 Add files 4 **Configure parameters**

Location coordinates: (X) None

.csv

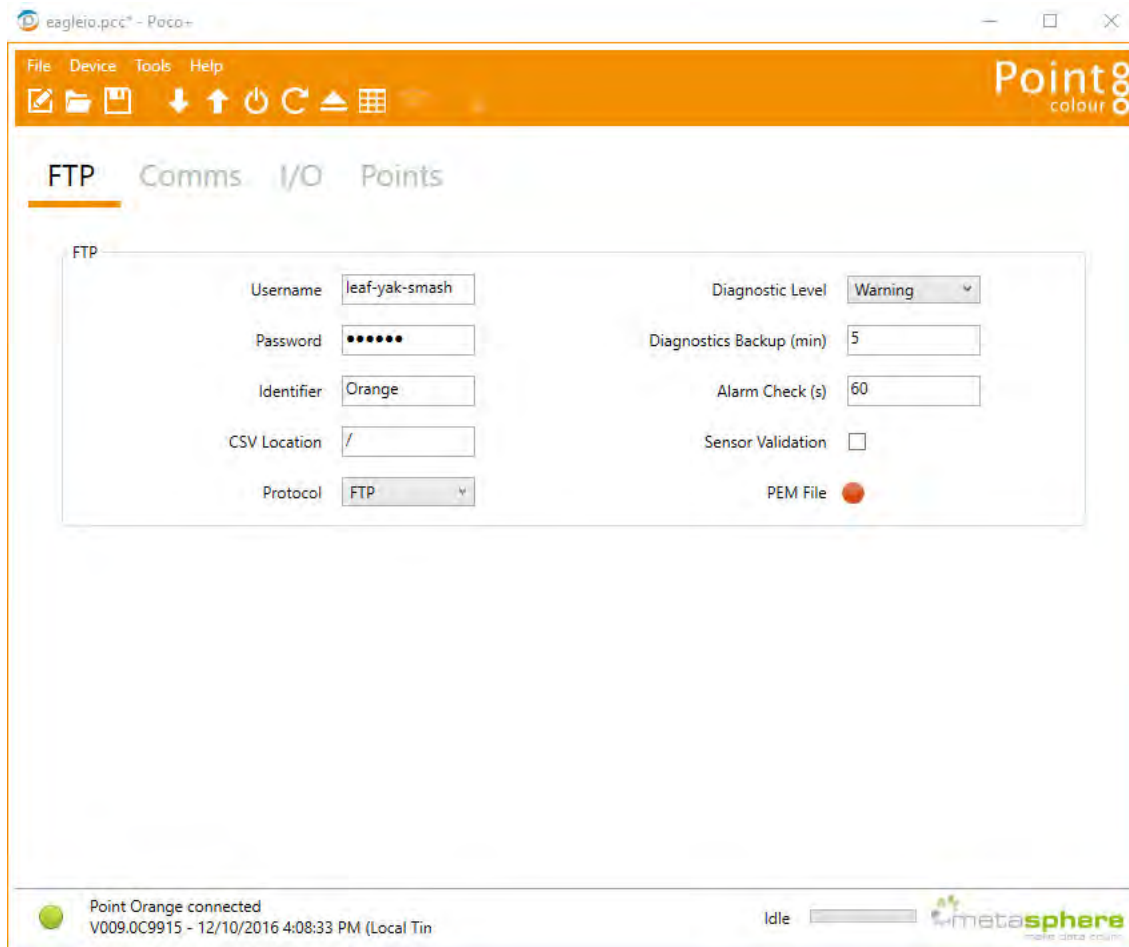
Series	Parameter assignment		Units	
<input checked="" type="checkbox"/> AI05 (C)	New	Existing	Temperature	°C
<input checked="" type="checkbox"/> AI06 (v)	New	Existing	Battery	V
<input checked="" type="checkbox"/> AI20	New	Existing	Level	m

Previous Finish Cancel

**Note:** You can reconfigure the file name match/sample file, parser configuration and parameter assignment from the *Data Source* (page 114) properties dialog after creation.

## Device settings

Use the Metasphere Poco+ application software to configure the logger for communication with eagle.io. Refer to your Point Orange 3G [user manual](https://www.metasphere.co.uk) (<https://www.metasphere.co.uk>) for connection instructions.



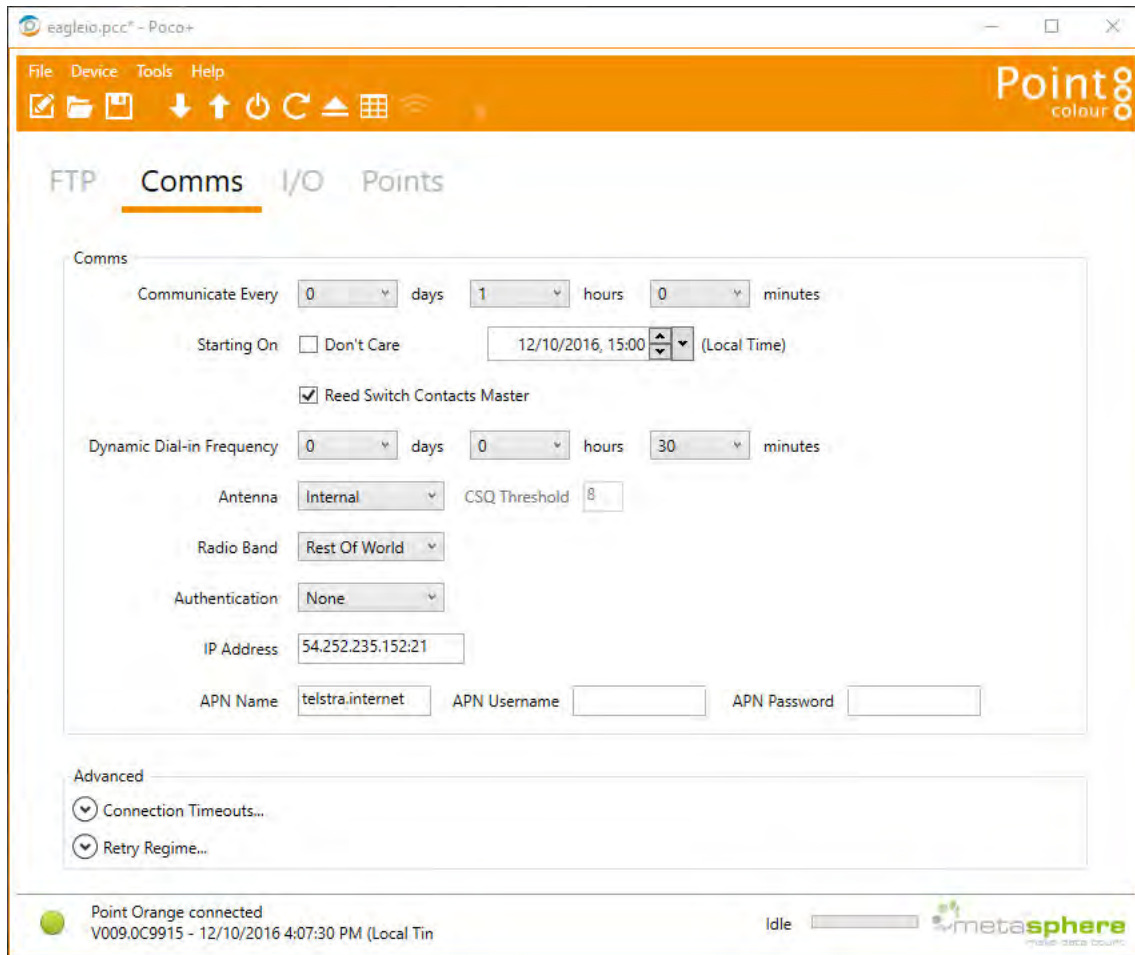
The following settings should be assigned in the **FTP** section:

**Username** Use the unique *FTP user* generated during Data Source creation.

**Password** Use the password you assigned during Data Source creation (or set to 'pass' for none).

**CSV Location** /

**Protocol** FTP



The following settings should be assigned in the **Comms** section:

**IP Address** 54.252.235.152:21

You should configure the communication frequency and APN settings as required.

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

### 2.14.13 Scout


The *Scout modem* is the easiest way to connect your data logger to [eagle.io](https://eagle.io) (<https://eagle.io>). The *Scout* takes any standard internet ready SIM Card without the need for fixed IP addresses and has a range of smart features including inbuilt GPS, signal strength monitoring, event based reporting and low power mode.





## Source configuration

You need to configure your Data Source to use the **Scout** transport. You can then use the *Device settings* button to display the device settings dialog. See [configuring a Data Source](#) (page 134) for full details.

Transport type	<input type="text" value="Scout"/>	
Device Id (IMEI)	<input type="text" value="356938035643809"/>	
	<input type="button" value="Device settings"/>	Last connected NEVER

## Device settings

The device settings dialog allows you to review and configure the settings for your Scout. The status message at the bottom of the dialog will indicate if the remote device is *up to date* or *out of date*. Configuration updates will automatically be applied on the next successful connection.

## General

The general section allows you to configure the device for use with your SIM.

The screenshot displays two sections of the eagle.io configuration interface. The top section contains four items: 'Status updates' with a dropdown menu set to '10 minutes', 'Phone number' with a text input field containing 'eg. +61400000001', 'SIM pin' with an empty text input field, and 'GPS enabled' with a checked checkbox. The bottom section is titled 'Automatic network configuration' (with a checked checkbox) and contains three text input fields: 'Access point name', 'APN username', and 'APN password'. Each input field has a help icon (a question mark in a circle) to its right.

**Status updates** Frequency of status updates - including GPS coordinates and signal strength. This option is only available when the Data Source schedule is *Enabled* and Schedule type set to *Server*. When the Data Source schedule is *Disabled* or set to *Remote* the status updates will only occur when the device performs a Callback.

**Phone number** Phone number of the SIM Card in the device. Follow the standard [international format](http://en.wikipedia.org/wiki/National_conventions_for_writing_telephone_numbers) ([http://en.wikipedia.org/wiki/National\\_conventions\\_for\\_writing\\_telephone\\_numbers](http://en.wikipedia.org/wiki/National_conventions_for_writing_telephone_numbers)) starting with '+' and including country and area code. When the Data Source schedule is *Disabled* or set to *Remote* the phone number is used to send an SMS to the device to initiate a callback.

**SIM pin** PIN of the SIM Card or leave blank for no pin.

**GPS enabled** Disable the GPS if not needed to reduce power consumption.

**Network configuration** Uncheck *Automatic network configuration* only when the automatic mode does not work with your SIM. You will need to configure the Access point name and username/password as specified by your provider.

## Communication

The communication section allows you to change the COM1 port settings to correctly pass data between eagle.io and your data logger.

**COM1 [Serial Interface]**

Baud rate	115200	?
Data bits	8	
Parity	None	
Stop bits	1	
Rts	Off	
Cts	Off	

Reset to defaults

**Note:** Some data loggers may require a specific Baud rate, but most other settings should use the defaults.

## Configuration

The configuration section allows you to view software versions and configure updates.

Software updates	Release	?
Device log level	Info	?
Bootloader version	SWI6200T_02.02.04.18BT	
Firmware version	FW_753_18.FX8Fx	
Application version	0.4.1.20140714155804	

**Software updates** Select an option for software updates:

*Release* will use the latest stable software (recommended).

*Development* will use an unstable version of the software still being tested.

*Do not update automatically* will disable the updates.

**Device log level** Select the detail level of the device logs. Debug is the lowest level and will record all events. Select a higher level to filter out less important events. Logs are only retrieved on request via the *Retrieve logs* button in the Operation section.

## Operation

The operation section can be used to remotely reset or reboot the *Scout modem* or retrieve its internal logs.

### Reset device configuration

Reset

This will reset the device to factory defaults.

### Reboot remote device

Reboot

This will send a reboot request to the device and cause it to immediately check for configuration updates.

### Retrieve device logs

Retrieve logs

This will send a logs request to the device and cause it to immediately return its logs to the server (available as Series in the Data Source).

#### 2.14.14 Senquip ORB-X1

The [Senquip](http://senquip.com) (<http://senquip.com>) **ORB-X1** connects to standard industrial sensors and allows remotely measured data to be delivered to users anywhere in the world. It can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.





## Source configuration

1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *JSON Time Series* from the list and click Next.

The screenshot shows the 'Data Source Wizard' window. At the top, there are four steps: 1. Source type (active), 2. Configure data logger, 3. Configure source, and 4. Configure parameters. Below the steps, there are two radio buttons: 'Data logger' and 'File'. The 'File' radio button is selected. To the right of the radio buttons is a list box containing 'Delimited Text' and 'JSON Time Series'. Below the list box is a text area. At the bottom of the wizard, there is a message: 'Please refer to our [documentation](#) for help configuring your specific device.' and two buttons: 'Next' and 'Cancel'.

2. Select *Publish to mqtt.eagle.io* as the transport type. A unique MQTT *Topic* will be automatically generated per Data Source. The *Topic* will always start with *io/eagle/source/*, followed by a unique 3-word code, for example in the screenshot below the *Topic* is *io/eagle/source/tail-chill-nic*. Change the Authentication setting from *None* to *CONNECT Message*, and you will notice that the *Username* becomes the 3-word code. Enter a password in the *Password* field, and click Next.

## Data Source Wizard

1 Source type

2 Configure source

3 Add files

4 Configure parameters

### Connection

Transport type

Publish to mqtt.eagle.io

Topic

io/eagle/source/tail-chill-nic

Authentication

CONNECT Message

Username

tail-chill-nic

Password

password\_here

IP whitelist

eg. 192.168.7.52/32

Add

Remove

### Collection

☐ Assign as Location coordinate source

Previous

Next

Cancel

3. In the next step, check the box to *Skip or Reset sample file and retrieve on next acquisition* and click Apply.



Sample data
×

Sample file

Browse
Select a sample file
?

☒ Skip or Reset sample file and retrieve on next acquisition

Apply
Cancel

4. Complete the Data Source wizard by clicking *Next*, *Next* and *Finish*.

5. Final configuration requires the device to send some sample data. Follow these [detailed device configuration steps](http://docs.senquip.com/app_notes/APN0014%20Rev%201.0%20-%20Integrating%20ORB%20Data%20with%20Eagle.io.pdf) ([http://docs.senquip.com/app\\_notes/APN0014%20Rev%201.0%20-%20Integrating%20ORB%20Data%20with%20Eagle.io.pdf](http://docs.senquip.com/app_notes/APN0014%20Rev%201.0%20-%20Integrating%20ORB%20Data%20with%20Eagle.io.pdf)), and then check the following settings:

**MQTT** Check the *Enabled* box

**Broker Address** `mqtt.eagle.io:1883`

**Client ID** Leave blank

**Data Topic** Use the topic generated in step 2, e.g. `io/eagle/source/tail-chill-nic`

**Username** Use the 3-word code generated in step 2, e.g. `tail-chill-nic`

**Password** Use the password you entered in step 2

## MQTT

MQTT

☒ Enabled

Broker Address

`mqtt.eagle.io:1883`

Client ID

Client ID

Data Topic

Data Topic

Username

`tail-chill-nic`

Password

`password_here`

Configure MQTTS

- Once sample data has been sent, final configuration of data series can be performed by right-clicking the Data Source in the eagle.io Workspace tree, and choosing *Properties*. Then click the *Series* tab, where data series' can be associated with Parameters.

General
Connection
Collection
Series
Historic
Metadata
Security

Location coordinates
⊗ None
?

Sensors

<input checked="" type="checkbox"/> Series	Parameter assignment			Units
<input checked="" type="checkbox"/> Ambient Temperature	New	Existing	Ambient Temperature ▼	°C
<input checked="" type="checkbox"/> Current	New	Existing	Current ▼	mA
<input checked="" type="checkbox"/> Battery Voltage	New	Existing	Battery Voltage ▼	V

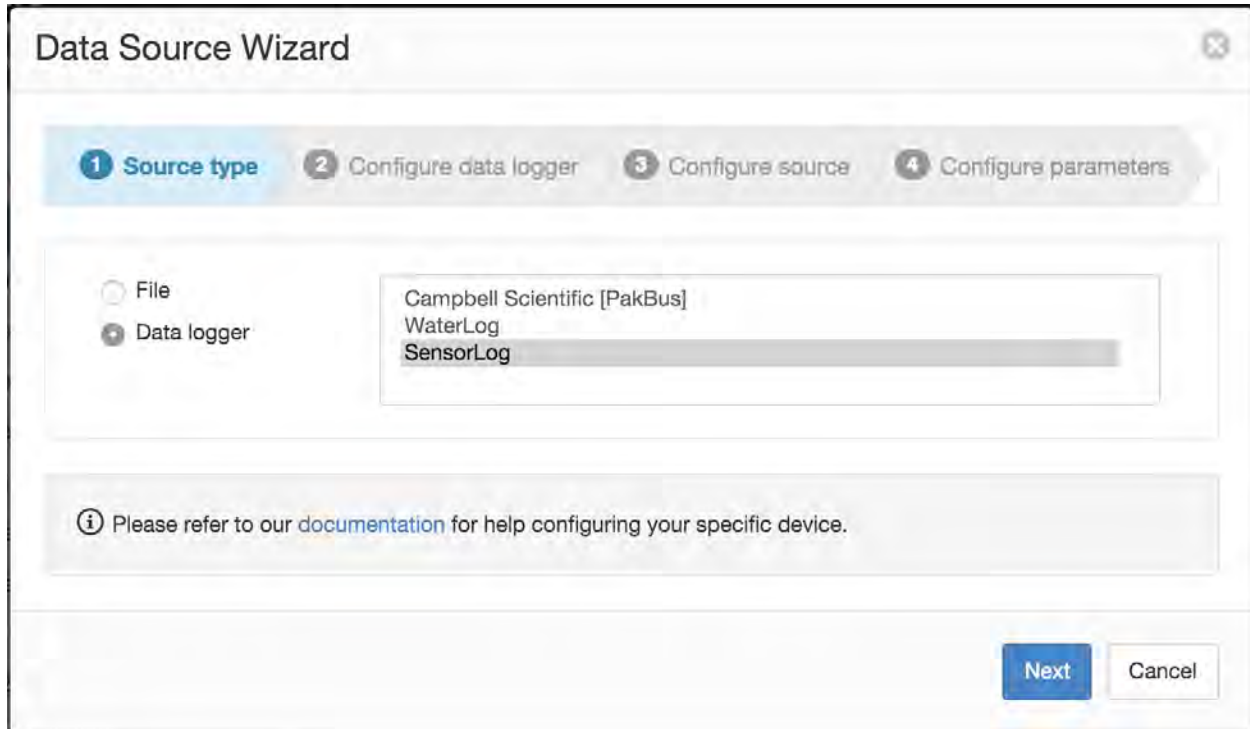
### 2.14.15 SensorLog

The [SensorLog](https://itunes.apple.com/au/app/sensorlog/id388014573/) (https://itunes.apple.com/au/app/sensorlog/id388014573/) iOS 8.0+ App for iPhone and iPad can be configured to send data to [eagle.io](https://eagle.io) (https://eagle.io) for demonstration purposes in a few easy steps.



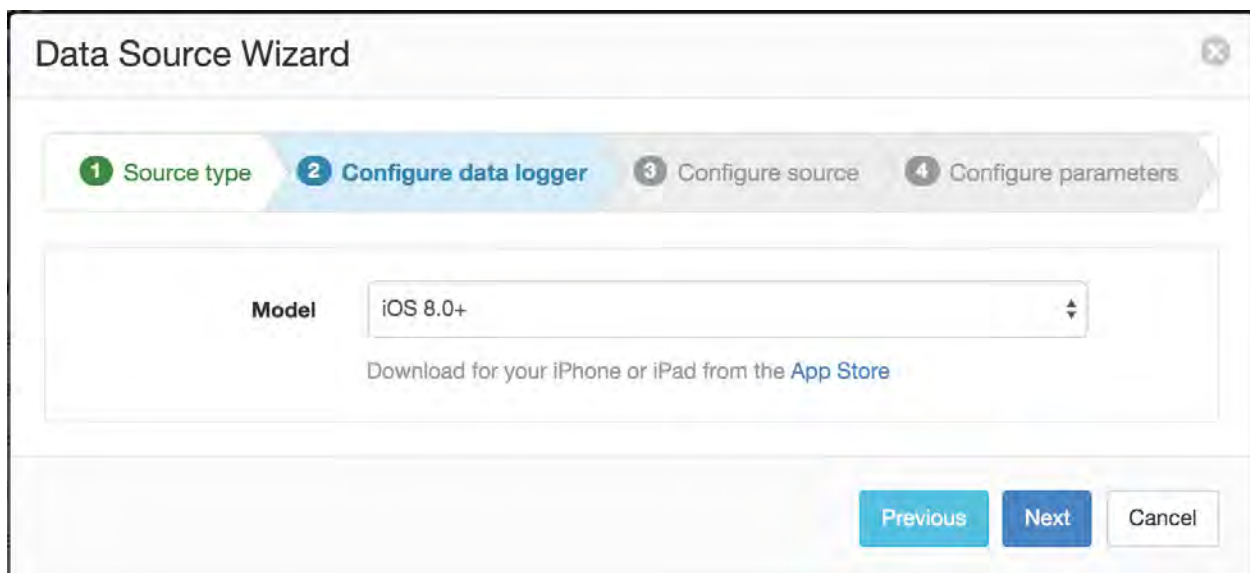
## Source configuration

1. Create a *Data Source* (page 114) and select *Data logger* as the Source Type. Choose *SensorLog* from the list and click Next.



The screenshot shows the 'Data Source Wizard' window with four steps: 1. Source type, 2. Configure data logger, 3. Configure source, and 4. Configure parameters. Step 1 is active. Under 'Source type', 'File' is unselected and 'Data logger' is selected. A list of data loggers is shown: 'Campbell Scientific [PakBus]', 'WaterLog', and 'SensorLog' (which is highlighted). A note at the bottom says: 'Please refer to our [documentation](#) for help configuring your specific device.' At the bottom right are 'Next' and 'Cancel' buttons.

2. Download the SensorLog App to your iPhone or iPad from the [App Store](https://itunes.apple.com/au/app/sensorlog/id388014573/) (<https://itunes.apple.com/au/app/sensorlog/id388014573/>) and click Next.



The screenshot shows the 'Data Source Wizard' window with four steps: 1. Source type, 2. Configure data logger, 3. Configure source, and 4. Configure parameters. Step 2 is active. A 'Model' dropdown menu is set to 'iOS 8.0+'. Below it is a link: 'Download for your iPhone or iPad from the [App Store](#)'. At the bottom right are 'Previous', 'Next', and 'Cancel' buttons.

3. The Device Id will be automatically generated. Select the Timezone of your device and click Next to continue.

The image shows a 'Data Source Wizard' dialog box with four steps: 1. Source type, 2. Configure data logger, 3. Configure source (highlighted), and 4. Configure parameters. The 'Connection' section shows 'Device Id' as 'nepal-dry-bead' with a 'Change' button. The 'Time' section shows 'Timezone' as '(GMT+10:00) Eastern Time - Melbourne, Sydney' with a dropdown arrow and a help icon. Below the timezone is an unchecked checkbox labeled 'Ignore Daylight Savings Time (data uses fixed UTC offset)'. At the bottom right are 'Previous', 'Next', and 'Cancel' buttons.

**Data Source Wizard**

1 Source type 2 Configure data logger 3 **Configure source** 4 Configure parameters

**Connection**

Device Id nepal-dry-bead Change

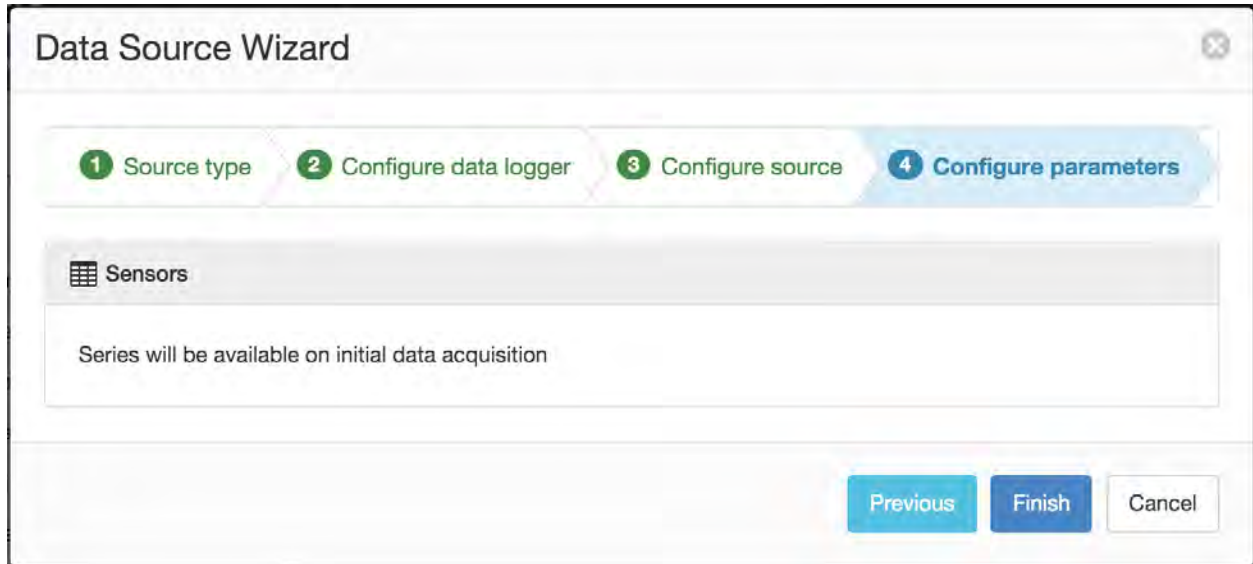
**Time**

Timezone (GMT+10:00) Eastern Time - Melbourne, Sydney ?

☐ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous Next Cancel

4. Click Finish to complete the setup. Parameters will be automatically created on initial data acquisition.



---

**Note:** You can enable or disable parameters for collection from the *Data Source* (page 114) properties dialog after creation.

---

## Device settings

Launch the SensorLog App and open the *Configuration* section.

amaysim

15:34

Done

Configuration

Info

HTTP upload:

☒

GET

POST

upload:

1s

5s

15s

30s

60s

300s

600s

URL:

http://device.eagle.io/sensorlog

device ID:

nepal-dry-bead

csv seperator:

.

;

|

default e-mail:

device ID

☒

DM

☒

CL

☒

IP

☒

ACC

☒

OR

☒

HEAD

☒

STATE

☒

GYRO

☒

PED

☒

ACT

☒

ALT

☒

battery

☒

X arbitrary

X arb. corr.

X mag. N

X true N

DM: CM attitude reference frame (Z vertical)

The following settings should be assigned:

**HTTP upload** Enabled. Select GET or POST

**upload** Set the upload frequency from 5s - 60s

**URL** <http://device.eagle.io/sensorlog>

**device ID** Use the unique *Device Id* generated during Data Source creation. eg. nepal-dry-bead

**csv separator** Comma ‘,’

**Sensors** Enabled the sensors you wish to log. The selected sensors will be automatically created as parameters on initial data acquisition.

---

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

---

### 2.14.16 Smithtek Mako V2

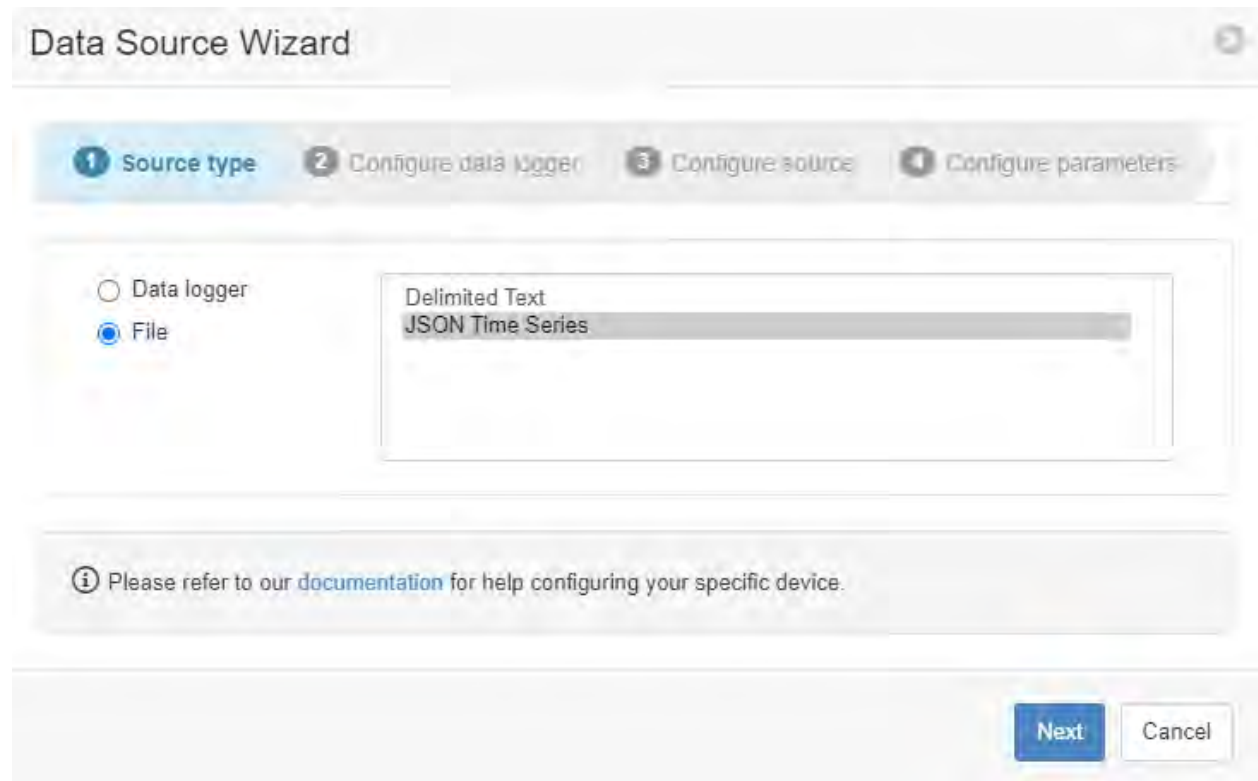
The [Smithtek](https://www.smithtek.com.au/products) (<https://www.smithtek.com.au/products>) **Mako V2** is a compact, low-power LoRa (RF) PLC designed for reliable, smart and wireless operation in applications such as water treatment plants, power generation, environmental, and pumping systems. With its embedded LoRa antenna, it offers long-range connectivity up to 15 km and can run 24/7 on minimal solar backup. It can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.





## Source configuration

1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *JSON Time Series* from the list and click Next.



The screenshot shows the 'Data Source Wizard' window. At the top, there are four steps: 1. Source type (highlighted), 2. Configure data logger, 3. Configure source, and 4. Configure parameters. Below the steps, there are two radio buttons: 'Data logger' (unselected) and 'File' (selected). To the right of the 'File' radio button is a list box containing two options: 'Delimited Text' and 'JSON Time Series', with 'JSON Time Series' selected. Below the list box, there is a message: 'Please refer to our [documentation](#) for help configuring your specific device.' At the bottom right, there are two buttons: 'Next' (highlighted) and 'Cancel'.

2. Select *Publish to mqtt.eagle.io* as the transport type. A unique MQTT *Topic* will be automatically generated per Data Source. The *Topic* will always start with *io/eagle/source/*, followed by a unique 3-word code, for example in the screenshot below the *Topic* is *io/eagle/source/tail-chill-nic*. Change the Authentication setting from *None* to *CONNECT Message*, and you will notice that the *Username* becomes the 3-word code. Enter a password in the *Password* field, and click Next.

## Data Source Wizard

1 Source type

2 Configure source

3 Add files

4 Configure parameters

### Connection

Transport type

Publish to mqtt.eagle.io

Topic

io/eagle/source/tail-chill-nic

Authentication

CONNECT Message

Username

tail-chill-nic

Password

password\_here

IP whitelist

eg. 192.168.7.52/32

Add

Remove

### Collection

☐ Assign as Location coordinate source

Previous

Next

Cancel

3. In the next step, check the box to *Skip or Reset sample file and retrieve on next acquisition* and click Apply.

Sample data

Sample file

Browse

Select a sample file

?

☒ Skip or Reset sample file and retrieve on next acquisition

Apply

Cancel

4. Complete the Data Source wizard by clicking *Next*, *Next* and *Finish*.
5. Final configuration requires the device to send some sample data. The device can be configured in two different ways; to send data directly to eagle.io (via a 4G dongle or WAN network), or using the PassPort Gateway over LoRa.
6. Direct connection via a 4G dongle or WAN network:



- 6.1. Download the V-net software ([https://26a6bc40-3386-4481-b01e-f59bb39d6cae.filesusr.com/archives/6ba956\\_63dc769f5a89455cbcc8b63a0d518781.zip?dn=V-Net35.zip](https://26a6bc40-3386-4481-b01e-f59bb39d6cae.filesusr.com/archives/6ba956_63dc769f5a89455cbcc8b63a0d518781.zip?dn=V-Net35.zip)) and install on your computer (Windows only).
- 6.2. Download the eagle.io template from Smithtek technical support.
- 6.3. Over Wifi, connect your Mako V2 to a local USB 4G dongle or a WAN network. Enter the Wifi SSID and Pass credentials.
- 6.4. Connect your physical sensors to the Mako V2 and within V-net software wire these sensors reading to the JTS form component.
- 6.5. Upload the program.
7. Connection using the PassPort Gateway over LoRa:



7.1. The PassPort Gateway can communicate with multiple Makos over long distances. This kind of system is very useful in locations with no mobile signal but has some form of fixed WAN network, like Starlink or NBN Sky Muster. It can also be used to minimize the need for multiple mobile SIMs, keeping your communications costs down.

7.2. Configure the Mako to send and receive data over its LoRa antenna within the V-net configuration software. Connect wires between single or arrays of sensors. Data formats can be binary packets, raw JSON data or some custom protocol.

7.3. In the LoRa properties, make sure your LoRa RF settings are the same on each device.

7.4. Start the PassPort Gateway and follow the connection guidelines to access the NodeRED editor and secure your instance.

7.5. Within the NodeRED editor, select the dedicated eagle.io nodes or the MQTT nodes

7.6. To configure the Smithtek eagle.io nodes follow [these instructions](https://flows.nodered.org/node/smithtek-node-red-eagleio) (<https://flows.nodered.org/node/smithtek-node-red-eagleio>)

8. Once sample data has been sent, final configuration of data series can be performed by right-clicking the Data Source in the eagle.io Workspace tree, and choosing *Properties*. Then click the *Series* tab, where data series' can be associated with Parameters.

## 2.14.17 Sigfox

The [Sigfox](https://www.sigfox.com/) (<https://www.sigfox.com/>) **network** can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in CSV format embedded in an email in a few easy steps.



## Source configuration

1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.

### Data Source Wizard

1 Source type

2 Configure data logger

3 Configure source

4 Configure parameters

☐ Data logger

☒ File

☐ Processor

Delimited Text

JSON Time Series

Please refer to our [documentation](#) for help configuring your specific device.

Next

Cancel

2. Select *Email to eagle.io* as the transport type. A unique *email address* will be automatically generated per Data Source. Click Next to continue.

Data Source Wizard

1 Source type

2 Configure source

3 Add files

4 Configure parameters

Connection

Transport type

Email to eagle.io

?

Email address

upload+walk-brian-self@eagle.io

Change

Sender address filter

user@company.com or @company.com or leave blank for no restrictic

?

Collection

☐ Assign as Location coordinate source

?

Time

Timezone

(GMT+10:00) Eastern Time - Melbourne, Sydney

?

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous

Next

Cancel

3. Select *Body* for the *Email data* and click the *Skip or Reset sample file and retrieve on next acquisition* checkbox.

### Add files

**Email data** Body ?

**Sample file** Browse Select a sample file ?

☒ Skip or Reset sample file and retrieve on next acquisition

**File name match**  Reset ?

☐ Match using regular expression

Apply Cancel

**Note:** If you do not have a sample available, click the *Skip* checkbox and click finish on the next screen. A sample will be saved automatically on the initial upload from the data logger and you will need to open the [Data Source](#) (page 134) properties dialog and [configure the series](#) (page 119) to create parameters.

- The Parser configuration screen allows the user to describe the structure and format of the data file so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details.

### Parser configuration Email Body

Column delimiter: , Format: Column series Labels row: 1 Units row: None Text qualifier: / Text encoding: UTF-8 (default) Match: Smart Reset to defaults

	Time	Lat	Lng	Temperature	Pressure	Photo
#	Record Time	Number	Number	Number	Number	Number
LABELS	Time	Lat	Lng	Temperature	Pressure	Photo
2	1506991886	-30.0	153.0	62985	23941	18177

Get help with parser configuration Apply Cancel

If the data embedded in the email contains a header row, set the *Labels row* control to the corresponding row number (1 by default) which will assign default labels for parameter creation.

Ensure the data in each column matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*.

You will need to use the drop-down menu above the columns to assign the *Record Time* data type to the column containing the record timestamp. Click the configure button to change the *Format* field to use the *time format* (page 583) **X** (Unix Timestamp).

Click Apply and Next.

**Note:** You can reconfigure the parser configuration and parameter assignment from the *Data Source* (page 114) properties dialog after creation.

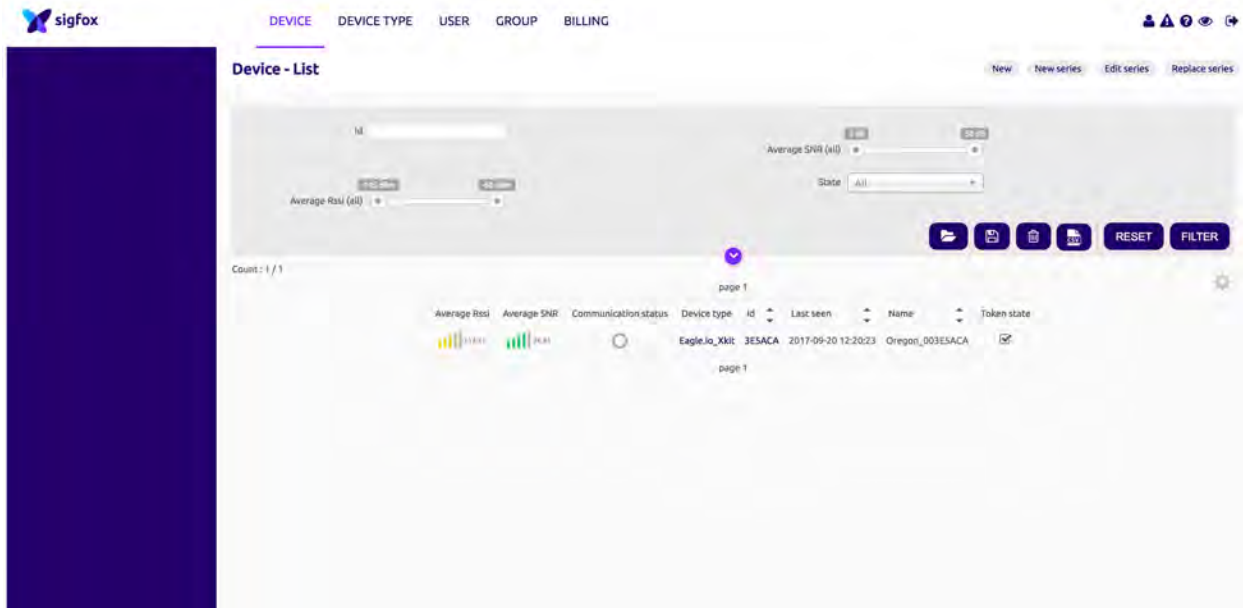
5. Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.

## Device settings

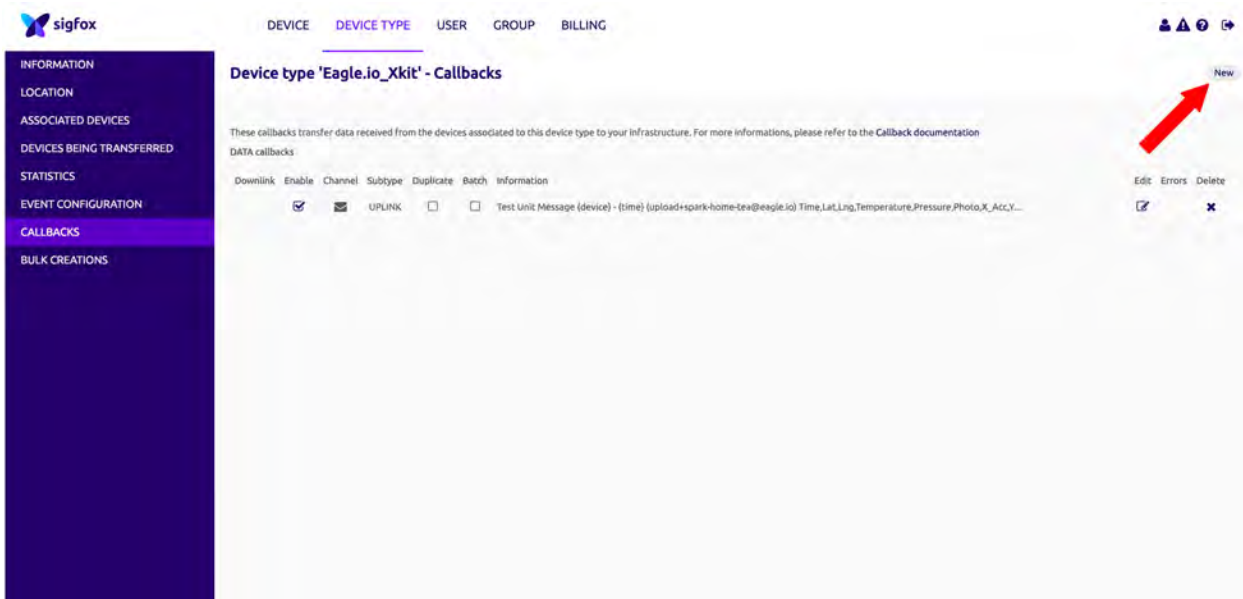
Configure the Sigfox Network to send data to eagle.io:

1. Login into your account and ensure your device is **activated** and **registered** (<http://makers.sigfox.com/getting-started/#register>) with the Sigfox network.





2. Select *Device Type* from the top navigation bar and select *Callbacks* from the menu on the left. Click the *New* button in the top-right of the Callbacks screen.



3. Create the Callback.

The following settings should be assigned:

**Type** DATA

**Channel** Email

**Custom payload config** You will need to set your own custom message type to decode your sigfox payload data into individual sensor values. Refer to [Sigfox Callbacks](http://makers.sigfox.com/getting-started/#callbacks) (<http://makers.sigfox.com/getting-started/#callbacks>) for help. Example:

```
temperature::uint:16 pressure:uint:16
```

**Recipient** Use the unique *Email address* generated during Data Source creation. eg. *upload+walk-brian-self@eagle.io*

**Subject** Any text can be used.

**Message** Use the Sigfox [built-in and custom fields](http://makers.sigfox.com/getting-started/#callbacks) (<http://makers.sigfox.com/getting-started/#callbacks>) for CSV output optionally including a header labels row. In the example below *time*, *lat* & *lng* are build-in fields. *temperature* and *pressure* are custom fields defined in the *Custom payload config*

```
Time,Lat,Lng, Temperature, Pressure
{time},{lat},{lng},{customData#temperature},{customData#pressure}
```

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

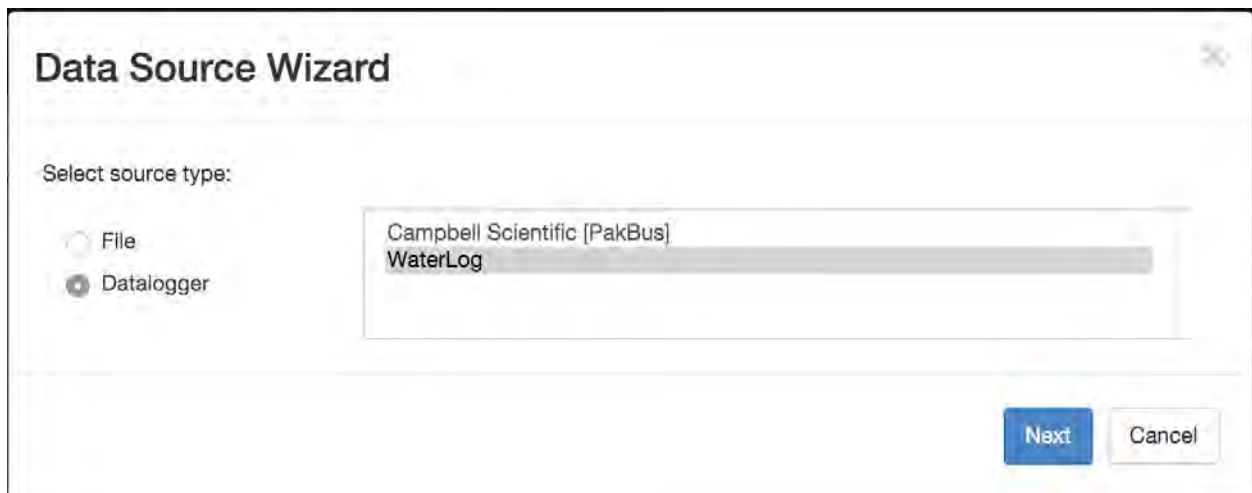
### 2.14.18 WaterLog

The [WaterLog](http://www.waterlog.com/) (<http://www.waterlog.com/>) **Storm 3** series data logger can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.

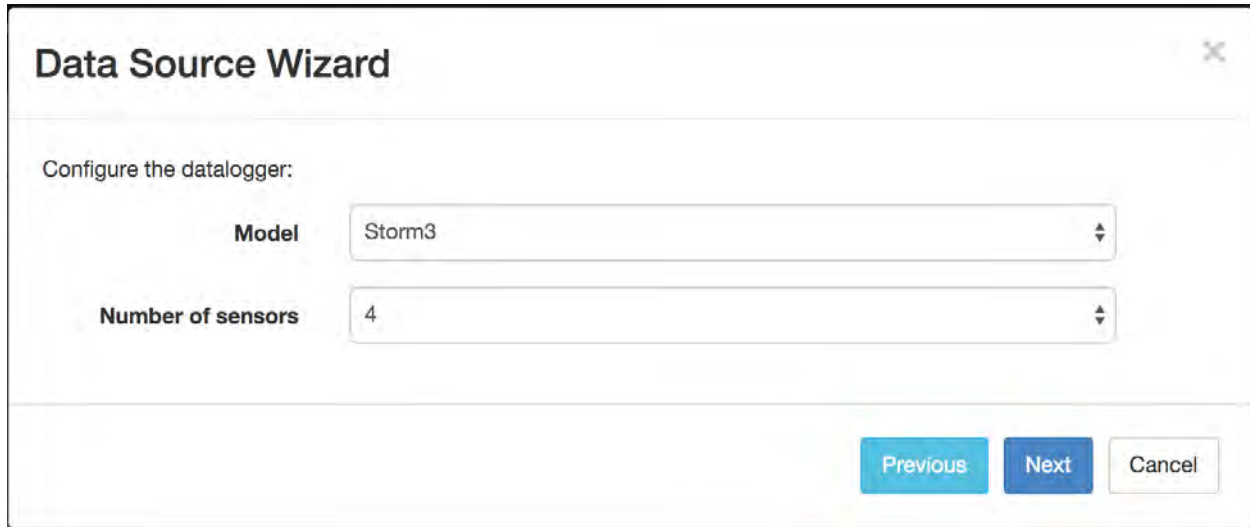


#### Source configuration

1. Create a *Data Source* (page 134) and select *Data logger* as the Source Type. Choose *WaterLog* from the list and click Next.

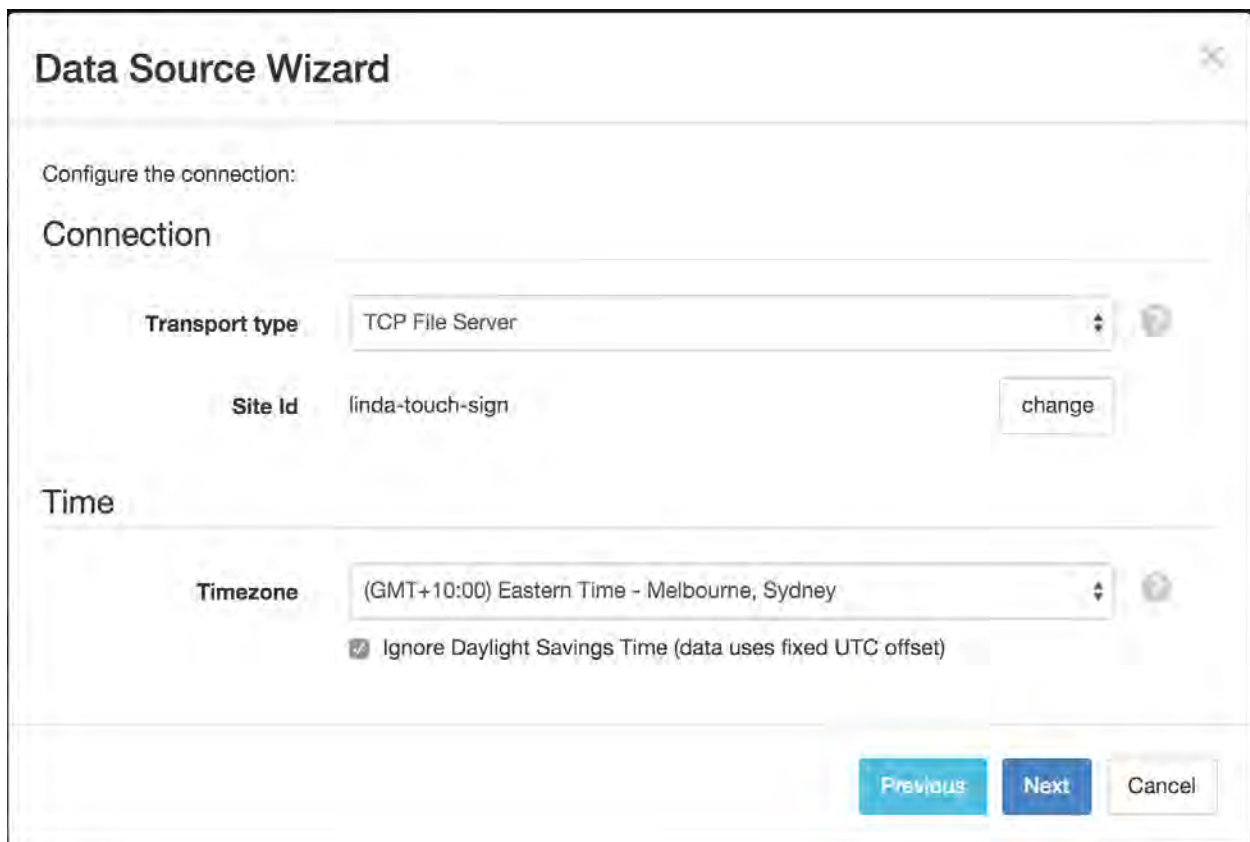


2. Select the data logger *Model* and specify the number of sensors connected (this can be changed later). Click Next.



The screenshot shows the 'Data Source Wizard' window with the title bar and a close button. The main heading is 'Data Source Wizard'. Below it, the instruction 'Configure the datalogger:' is displayed. There are two configuration fields: 'Model' with a dropdown menu showing 'Storm3' and 'Number of sensors' with a dropdown menu showing '4'. At the bottom right, there are three buttons: 'Previous' (blue), 'Next' (blue), and 'Cancel' (white).

3. Select *TCP Server* as the transport type and a *Site Id* will be automatically generated. Select the timezone of the logger's clock and click Next.



The screenshot shows the 'Data Source Wizard' window with the title bar and a close button. The main heading is 'Data Source Wizard'. Below it, the instruction 'Configure the connection:' is displayed. The 'Connection' section has a 'Transport type' dropdown menu showing 'TCP File Server' and a 'Site Id' text field showing 'linda-touch-sign' with a 'change' button next to it. The 'Time' section has a 'Timezone' dropdown menu showing '(GMT+10:00) Eastern Time - Melbourne, Sydney' and a checkbox labeled 'Ignore Daylight Savings Time (data uses fixed UTC offset)' which is checked. At the bottom right, there are three buttons: 'Previous' (blue), 'Next' (blue), and 'Cancel' (white).

4. Specify the parameter names for each connected sensor and optionally assign units. Click Finish to

complete the setup.

## Data Source Wizard

Configure series to be acquired

Location coordinates
None

### Sensors

Series	Parameter assignment		Units
<input checked="" type="checkbox"/> 1	New	Existing	Battery VDC
<input checked="" type="checkbox"/> 2	New	Existing	Level m
<input checked="" type="checkbox"/> 3	New	Existing	Velocity m/s
<input checked="" type="checkbox"/> 4	New	Existing	Temperature °C

Previous
Finish
Cancel

**Note:** You can reconfigure the number of sensors, ordering of sensors and Site Id from the [Data Source](#) (page 134) properties dialog after creation.

## Device settings

Use the WaterLog Storm3 configuration software to setup your Storm3 data logger for communication with eagle.io.

## Overview

The *System Overview* section accessible from the *Home* menu requires the following settings to be applied:

**Site ID** Use the unique *Site Id* identification code generated during Data Source creation.

**System Date/Time** Ensure the correct local time is set in the device. You should set the timezone of the Data Source to match your local timezone.



## Storm Central Setup

The *Storm Central Setup* section accessible from the *Outputs* menu requires the following settings to be applied:

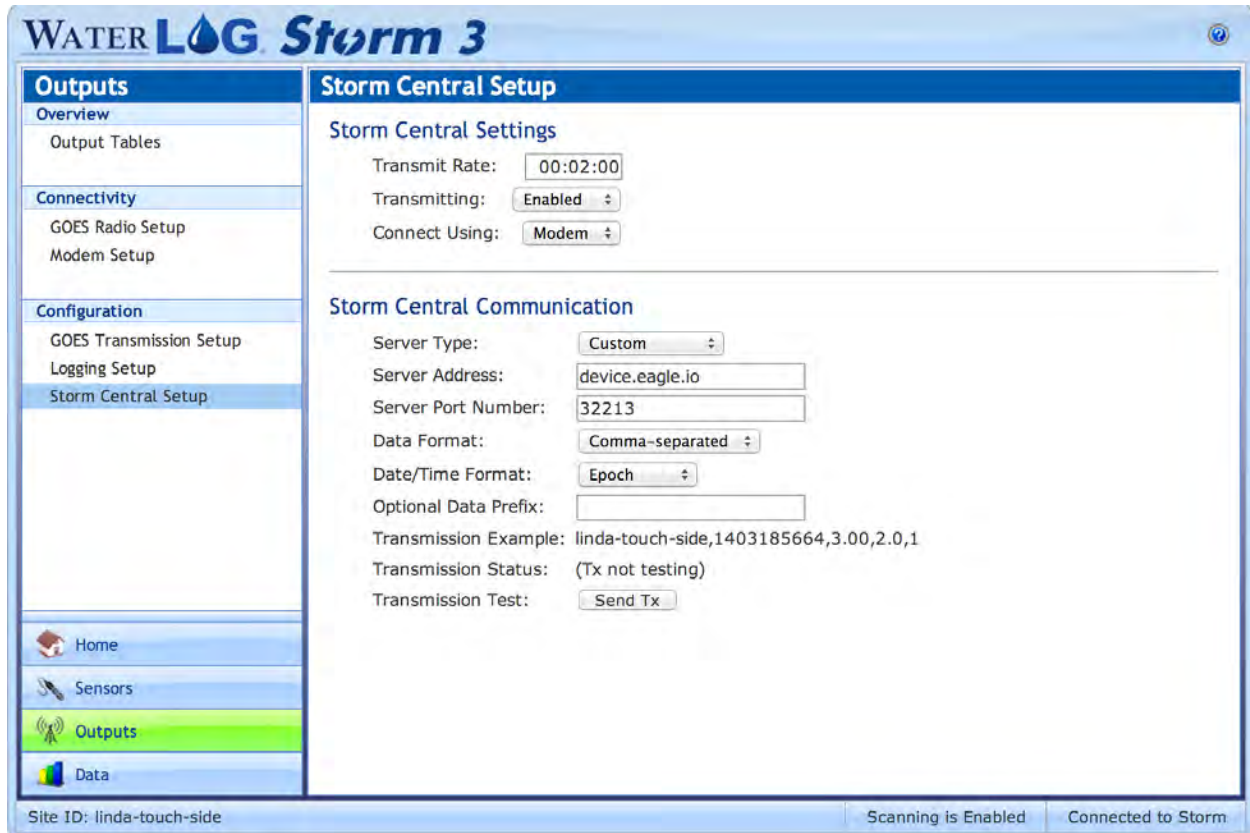
**Server Type** Custom

**Server Address** device.eagle.io

**Server Port Number** 32213

**Data Format** Comma-separated

## Date/Time Format Epoch



The screenshot shows the 'WATERLOG Storm 3' web interface. The left sidebar contains a navigation menu with 'Outputs' selected. The main content area is titled 'Storm Central Setup' and contains two sections: 'Storm Central Settings' and 'Storm Central Communication'.

**Storm Central Settings**

- Transmit Rate: 00:02:00
- Transmitting: Enabled
- Connect Using: Modem

**Storm Central Communication**

- Server Type: Custom
- Server Address: device.eagle.io
- Server Port Number: 32213
- Data Format: Comma-separated
- Date/Time Format: Epoch
- Optional Data Prefix:
- Transmission Example: linda-touch-side,1403185664,3.00,2.0,1
- Transmission Status: (Tx not testing)
- Transmission Test: Send Tx

At the bottom of the interface, there is a status bar showing 'Site ID: linda-touch-side', 'Scanning is Enabled', and 'Connected to Storm'.

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

### 2.14.19 YDOC

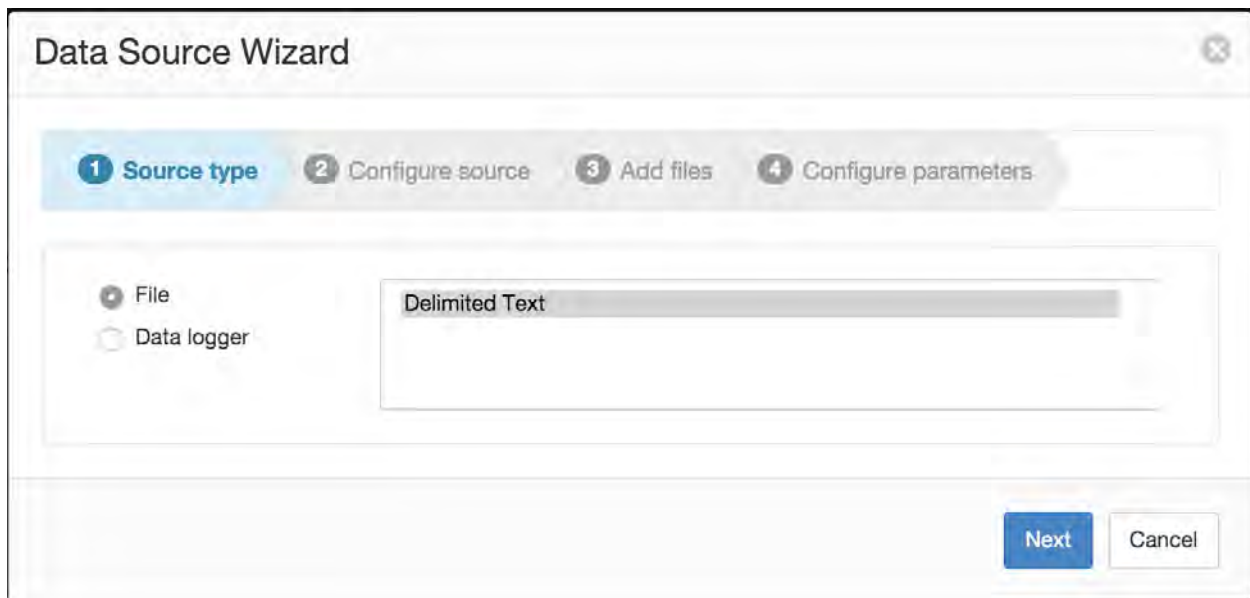
The YDOC (<http://www.your-data-our-care.com/>) ML-315 range of data loggers can be configured to send data to [eagle.io](https://eagle.io) (<https://eagle.io>) in a few easy steps.





## Source configuration

1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.



The screenshot shows the 'Data Source Wizard' dialog box. At the top, there are four steps: 1. Source type (highlighted in blue), 2. Configure source, 3. Add files, and 4. Configure parameters. Below the steps, there are two radio buttons: 'File' (selected) and 'Data logger'. To the right of the 'File' radio button is a list box containing 'Delimited Text'. At the bottom right of the dialog, there are two buttons: 'Next' (blue) and 'Cancel' (white).

2. Select *Upload to ftp.eagle.io* as the transport type. A unique *Ftp user* account will be automatically generated per Data Source. You can optionally specify a password to be used by the logger for additional security. Select the *Timezone* you have configured in the YDOC logger and leave the *Ignore Daylight Savings Time* option checked (unless you have enabled the DST adjustment option in the logger). Click Next to continue.



The screenshot shows the 'Data Source Wizard' window with a progress bar at the top indicating four steps: 1. Source type, 2. Configure source (active), 3. Add files, and 4. Configure parameters. The 'Connection' section contains three fields: 'Transport type' set to 'Upload to ftp.eagle.io', 'FTP user' set to 'cap-golf-wasp' with a 'Change' button, and 'FTP password' set to 'optional' with a password icon. The 'Time' section has a 'Timezone' dropdown set to '(GMT+10:00) Eastern Time - Melbourne, Sydney' and a checked checkbox for 'Ignore Daylight Savings Time (data uses fixed UTC offset)'. At the bottom right are 'Previous', 'Next', and 'Cancel' buttons.

**Data Source Wizard**

1 Source type 2 **Configure source** 3 Add files 4 Configure parameters

**Connection**

Transport type Upload to ftp.eagle.io

FTP user cap-golf-wasp Change

FTP password optional

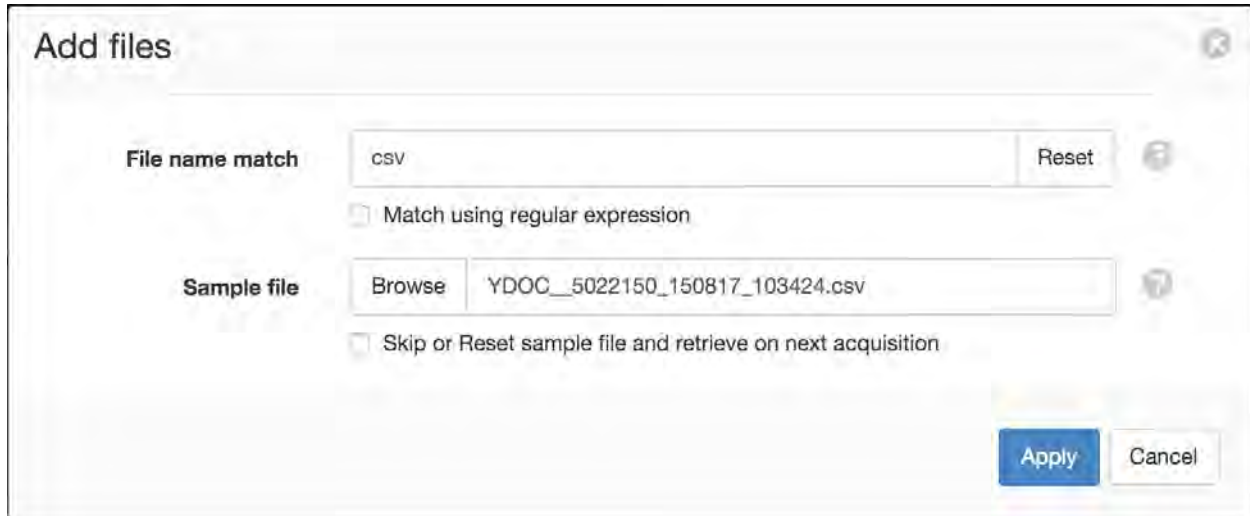
**Time**

Timezone (GMT+10:00) Eastern Time - Melbourne, Sydney

☒ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous Next Cancel

3. Specify *csv* as the *File name match*. This will match all incoming files for this Source containing 'csv' in the filename. To continue the setup you will need a sample copy of the logger's output data file(s).



**Add files**

**File name match**

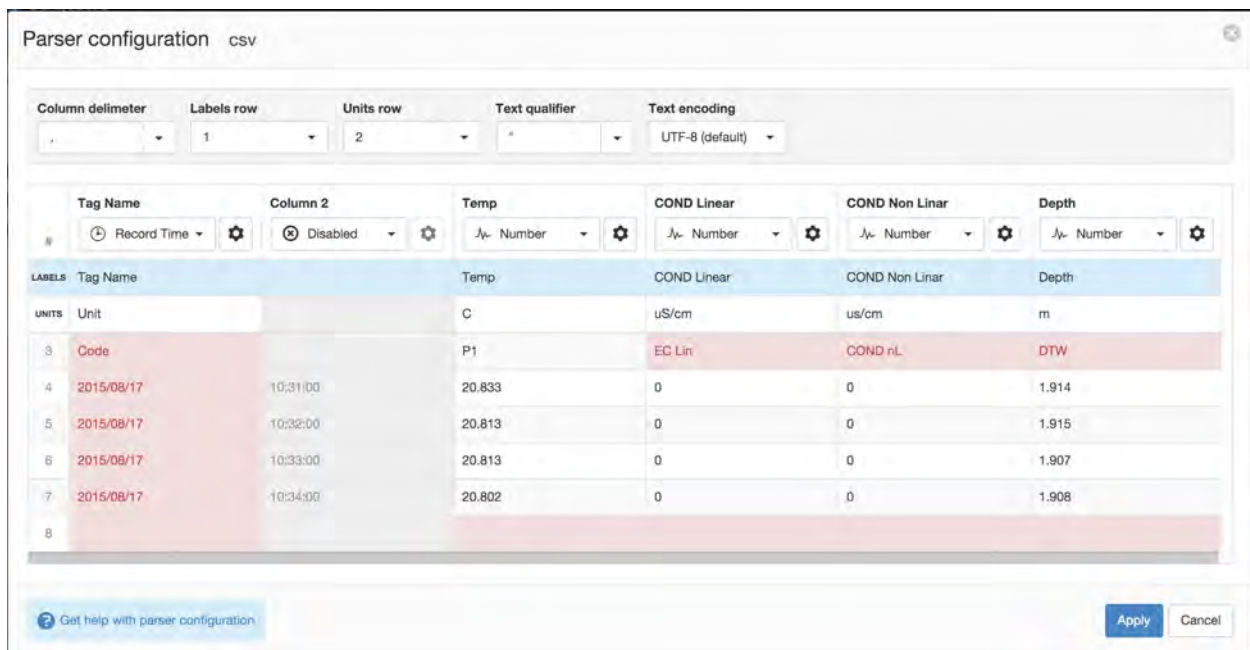
☐ Match using regular expression

**Sample file**

☐ Skip or Reset sample file and retrieve on next acquisition

**Note:** If you do not have a sample available, click the *Skip* checkbox and click finish on the next screen. A sample will be saved automatically on the initial upload from the data logger and you will need to open the [Data Source](#) (page 134) properties dialog and [configure the series](#) (page 119) to create parameters.

- The Parser configuration screen allows the user to describe the structure and format of the data file so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details.



**Parser configuration** csv

Column delimiter: , Labels row: 1 Units row: 2 Text qualifier: " Text encoding: UTF-8 (default)

Tag Name	Column 2	Temp	COND Linear	COND Non Linear	Depth
#	Record Time	Disabled	Number	Number	Number
LABELS	Tag Name	Temp	COND Linear	COND Non Linear	Depth
UNITS	Unit	C	uS/cm	us/cm	m
3	Code	P1	EC Lin	COND nL	DTW
4	2015/08/17	10:31:00	20.833	0	1.914
5	2015/08/17	10:32:00	20.813	0	1.915
6	2015/08/17	10:33:00	20.813	0	1.907
7	2015/08/17	10:34:00	20.802	0	1.908
8					

If the file contains header rows, set the *Labels row* control to the corresponding row number (1 by default)

which will assign default labels for parameter creation. You can also specify the *Units row* if it is available (2 by default).

Ensure the data in each column matches the expected column data type and format. Validation errors are shown in red. Any columns not required for collection can optionally be *Disabled*.

You will need to use the drop-down menu above the columns to assign the *Record Time* data type to the first column containing part of the record timestamp. Click the configure button to change the *Format* field to use the *time format* (page 583) **YYYY/MM/DD**.

The record timestamp spans multiple columns. Add any additional columns containing part of the timestamp and specify the format. By default you will need to add 'Column 2' to the list of *Column joins* and change the *Format* field to use the *time format* (page 583) **HH:mm:ss**. Click Apply and Next.

Parser configuration csv

Column delimiter: , Labels row: 1 Units row: 2 Text qualifier: " Text encoding: UTF-8 (default)

Tag Name + Column 2

Record Time

LABELS Tag Name

UNITS Unit

	Code	Unit
3	TIME: 2015-08-17 10:31:00	
4	2015/08/17 10:31:00	
5	2015/08/17 10:32:00	
6	2015/08/17 10:33:00	
7	2015/08/17 10:34:00	
8		

Column joins and configuration (Tag Name)

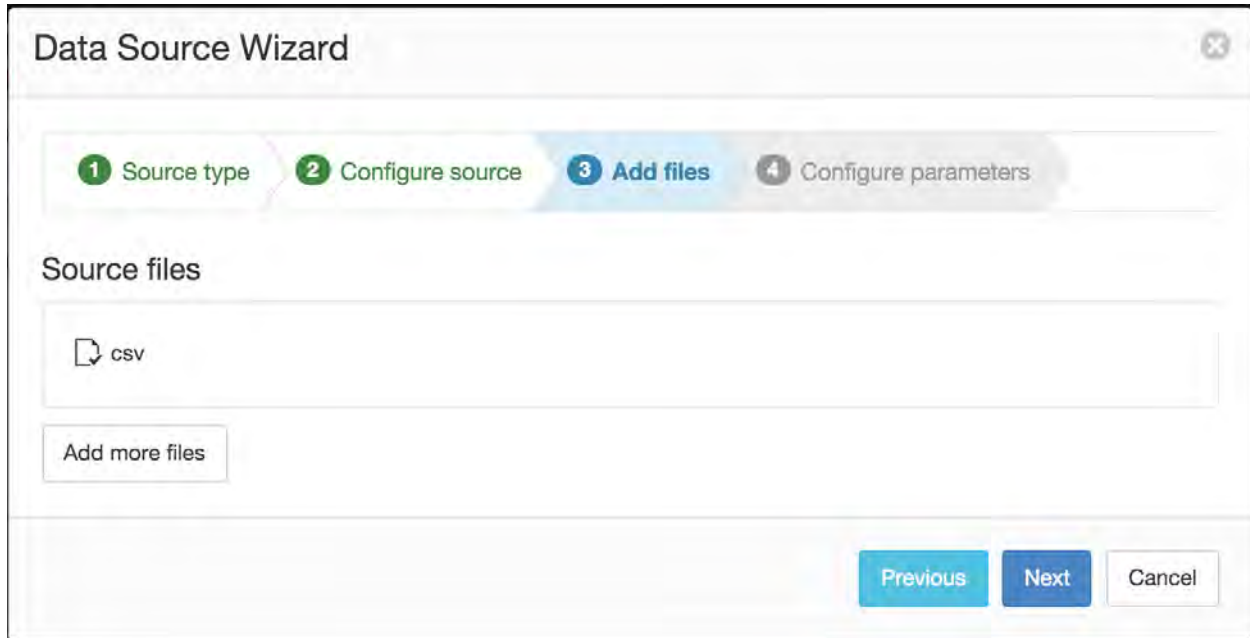
Column joins	Field	Format
Tag Name	Time	YYYY/MM/DD
Column 2	Time	HH:mm:ss

Tag Name Quality Add join

Get help with parser configuration

Apply Cancel

5. Click next to *Configure parameters*.



6. Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.

**Data Source Wizard**

1 Source type 2 Configure source 3 Add files 4 **Configure parameters**

Location coordinates: ⊗ None

csv

Series	Parameter assignment		Units
<input checked="" type="checkbox"/> Temp	<span>New</span>	<span>Existing</span>	Temperature °C
<input checked="" type="checkbox"/> COND Linear	<span>New</span>	<span>Existing</span>	Conductivity uS/cm
<input type="checkbox"/> COND Non Linar	<span>New</span>	<span>Existing</span>	COND Non Linar uS/cm
<input checked="" type="checkbox"/> Depth	<span>New</span>	<span>Existing</span>	Depth m

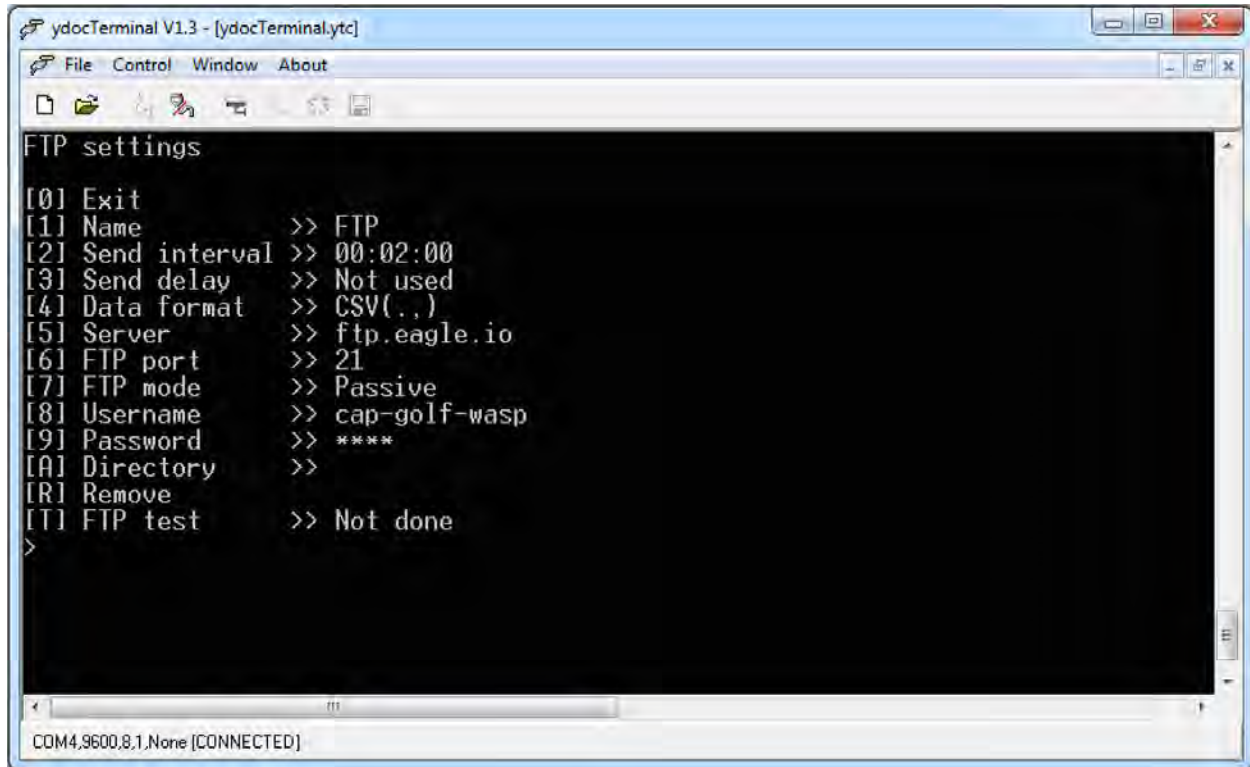
Previous Finish Cancel

**Note:** You can reconfigure the file name match/sample file, parser configuration and parameter assignment from the [Data Source](#) (page 114) properties dialog after creation.

## Device settings

Use the YDOC terminal software to configure the logger for communication with eagle.io. Refer to your YDOC [user manual](http://www.your-data-our-care.com/) (<http://www.your-data-our-care.com/>) for connection instructions.

Select option *[D] FTP output* from the *Configuration setup* menu.



The following FTP settings should be assigned:

[2] **Send interval** Specify the frequency of the data uploads to eagle.io.

[4] **Data format** CSV (Option 1)

[5] **Server** ftp.eagle.io

[6] **FTP port** 21

[7] **FTP mode** Passive

[8] **Username** Use the unique *FTP user* generated during Data Source creation. eg. cap-golf-wasp

[9] **Password** Use the password you assigned during Data Source creation (or set to 'pass' for none).

You should also ensure the YDOC data logger has been configured with the correct time and timezone offset.

---

**Note:** Use the [Events](#) (page 43) view with the detail level set to *Debug* for reviewing communication history and diagnosing configuration issues.

---


### 2.14.20 YuDash

The [YuDash LYNX](http://www.yudash.com) (<http://www.yudash.com>) is a flexible Industrial IoT gateway that can read various industrial sensors/instrumentation over different protocols: Modbus RS485, analog inputs, Modbus TCP/IP, I2C sensors, and many more. Connectivity is via WiFi/Ethernet LAN or 4G/LTE. It supports HTTP POST and MQTT Publish to send data to eagle.io, and has a flexible payload format.



## Source configuration

1. Create a *Data Source* (page 114) and select *File* as the Source Type. Choose *Delimited Text* from the list and click Next.



Data Source Wizard

1 Source type 2 Configure source 3 Add files 4 Configure parameters

☒ File  
☐ Data logger

Delimited Text

Next Cancel

2. Select *Upload via HTTP* as the transport type. A unique URL will be automatically generated per Data Source. Select the *Timezone* and *Ignore Daylight Savings Time* options which match the timezone of the corresponding YuDash device. Note that in the following example, the URL ends with the three words **moon-soda-calf**; these three words will be different for each Data Source that is created.

Data Source Wizard

1 Source type

2 Configure source

3 Add files

4 Configure parameters

Connection

Transport type

Upload via HTTP

?

URL

https://upload.eagle.io/source/moon-soda-calf

Authentication

None

?

IP whitelist

eg. 192.168.7.52/32

Add

?

Collection

☐ Assign as Location coordinate source

?

Time

Timezone

(GMT+10:00) Eastern Time - Melbourne, Sydney

?

☐ Ignore Daylight Savings Time (data uses fixed UTC offset)

Previous

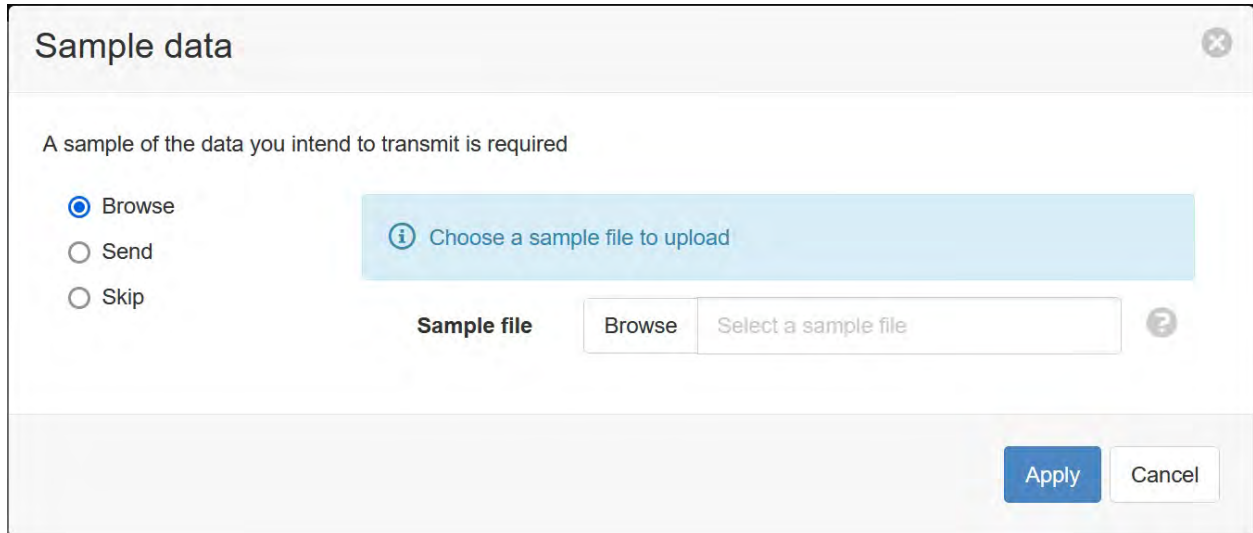
Next

Cancel

- Click the *Browse* button and select a text file that contains a sample data format consistent with the data which will be sent by the device. An example data format would be:

```
2023-02-19 21:20:00,234.5,50.5,3456
```





**Sample data** ✕

A sample of the data you intend to transmit is required

☒ Browse
 ☐ Send
 ☐ Skip

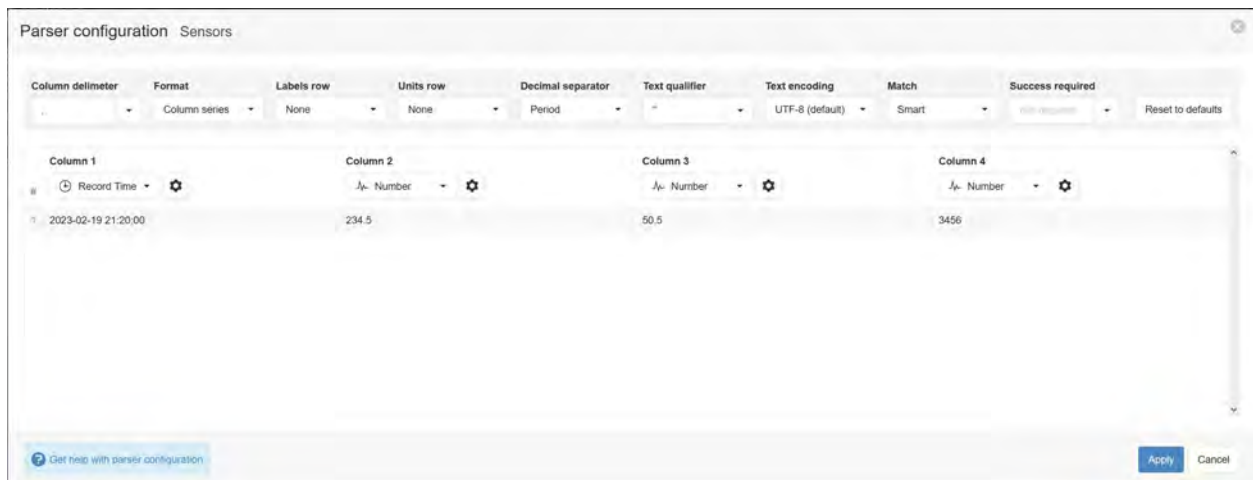
i Choose a sample file to upload

Sample file
 

?

**Note:** If you do not have a sample text file available, instead select the *Skip* option and click finish on the next screen. A sample will be saved automatically on the initial upload from the device and you will need to open the [Data Source](#) (page 134) properties dialog and *configure the series* (page 119) to create parameters.

- The Parser configuration screen allows the user to describe the structure and format of the data file so it can be correctly interpreted by the system in subsequent uploads. Refer to [Text Parser Configuration](#) (page 123) for full details. Click Apply to *Configure parameters*.



**Parser configuration** Sensors ✕

Column delimiter	Format	Labels row	Units row	Decimal separator	Text qualifier	Text encoding	Match	Success required
.	Column series	None	None	Period	"	UTF-8 (default)	Smart	Not required

Column 1	Column 2	Column 3	Column 4
Record Time	Number	Number	Number
2023-02-19 21:20:00	234.5	50.5	3456

i Get help with parser configuration

- Specify the parameter names (and optionally units) for each series/column defined in the Parser configuration. Uncheck any series you do not want to create as parameters. Click Finish to complete the setup.

Data Source Wizard

1 Source type

2 Configure source

3 Add files

4 Configure parameters

Location coordinates

⊗ None

Sensors

<input checked="" type="checkbox"/> Series	Parameter assignment		Units
<input checked="" type="checkbox"/> Column 2	New	Existing	Column 2
<input checked="" type="checkbox"/> Column 3	New	Existing	Column 3
<input checked="" type="checkbox"/> Column 4	New	Existing	Column 4

Previous

Finish

Cancel

## Device settings

1. Configure the YuDash HTTP settings to match the unique URL that was previously generated during data source configuration. Because the unique three words in the example URL were **moon-soda-calf**, the HTTP API name is entered as **/source/moon-soda-calf**. The other details will remain the same for every eagle.io Data Source that is added.

2. The Payload Format to send parameters to eagle.io is **TEXT\_FORMAT\_20**. The timestamp format is **TS\_53**. This will be inserted in the payload CSV string.

3. Within payload settings, the YuDash Text Generator Engine is used to generate CSV. The Text Encoder text box is populated to generate required payload.
- ## is a general separator in the Encoder Text.
  - yu1\_<abc> [name] is a generic place-holder name to access process values.

- c) Other characters are passed in payload directly.
- d) **yu1\_vts** is the placeholder for timestamp in selected format.
- e) **yu1\_vnv** [**variable\_name**] is the placeholder for a named variable; **yu1\_vnv volt1\_in** will place the value of voltage **volt1\_in** read by LYNX.

LYNX Settings

Save/Load Settings

Modbus Settings

Analog Input Settings

DataScale Settings

YuDash Text Generator

Enable YuDash Text ☒

Text Key Name

YuDash Text Encoder

yu1\_vts ##, ## yu1\_vnv volt1\_in ##, ## yu1\_vnv freq\_in ##, ## yu1\_vnv kwhr\_in

#### 4. Generic LYNX Settings and Features.

- 1) The variable names (fields) are filled in Modbus settings of LYNX. This is a generic Modbus/RS485 setting of LYNX. Similar to Modbus/RS485; MODBUS/TCPIP, analog inputs and many other protocols can be used.
- 2) In this example, the variable names **volt1\_in**, **freq\_in** and **kwhr\_in** are sent to eagle.io by YuDash LYNX IoT gateway.

**LYNX Configuration**

- LYNX Device Info
- Network Settings
- LYNX Settings
- Save/Load Settings
- Modbus Settings
- Analog Input Settings
- DataScale Settings
- Advance Feature Settings
- Cloud Settings
- Payload Settings
- Ethernet Settings

**Modbus RS485 Settings**

BaudRate (bps) 9600  
Data Parity StopBit 8 N 1  
Read Delay (ms)  
Read Try Count  
Update Modbus Settings

**Modbus RS485 Slave Settings**

Read Modbus Slaves  
Add Slave Delete Last Slave  
Modbus Slaves : 1 Slave# 1  
Read MB Slave Update MB Slave  
Slave ID : 1 No of Registers# 3

Variable Name	#Register No	Type	Factor
volt1_in	30020	2	1
freq_in	30026	2	1
kwhr_in	30149	2	1

5. Select **HTTP** in the *Custom Cloud* section to send data to eagle.io.



LYNX Device Info

Network Settings

LYNX Settings

Save/Load Settings

Modbus Settings

Analog Input Settings

DataScale Settings

Advance Feature Settings

Cloud Settings

Payload Settings

Ethernet Settings

General Settings

Loop Delay (sec) 60

Device Name: YUDASH EAGLE.IO DEMO

Feature Settings

Modbus (RS485) ☒ Analog Inputs ☐

Digital IOs ☐ Digital Sensors ☐

Data Scaling ☐

Ethernet/LAN ☐ Modbus TCP/IP ☐

SNMP ☐

Cloud/Network Server

Use YuDash Cloud ☐

Custom Cloud ☐ YuDash ☐ Ubidots ☐ Offline

☐ MQTT ☒ HTTP ☐ FTP

Save/Load local LYNX Settings

6. LYNX has an on-device screen which shows connectivity with eagle.io



---

**Note:** Use the [Events](#) (page 43) view in eagle.io, with the detail level set to *Debug*, for reviewing communication history and diagnosing data source configuration issues.

---

---

**Note:** We are continually adding support for new devices. If your device is not listed [contact us](#) (<https://eagle.io/contact>) to get support.

---

## 2.15 Troubleshooting

**The interface is not displaying correctly or does not appear to be up to date** Try clearing your browser cache, refresh the page and ensure you are using the latest version of a supported web browser: Chrome, Firefox, Internet Explorer, Safari. Some browser addons or plugins are known to degrade performance and should be disabled when using eagle.io.

We recommend [Google Chrome](https://www.google.com/intl/en/chrome/browser/) (https://www.google.com/intl/en/chrome/browser/) for the best eagle.io experience.

**Uploading attachment failed** Ensure you have a stable internet connection and retry. Some camera phones generate large files which may be time consuming to upload. Note: there is a file size limit of 100MB per file.

**Location Nodes do not expand to show their contained Nodes in the Workspaces Tree** You must have *configure* permission on the Workspace to see Nodes contained within Locations from the Workspaces Tree. The content associated with the hidden Nodes will still be displayed in the Views.

**Context Menu not displayed or has limited options** Most context menu items require certain Workspace permissions. If there are no context menu items available to the user then the context menu will not be displayed.

**Other issue?** We are here to help, so please post a question or bug report on our [forum](http://forum.eagle.io) (http://forum.eagle.io) or contact us at [support@eagle.io](mailto:support@eagle.io)

### 2.15.1 Debug logging

If you are experiencing intermittent bugs or other software related issues, our software development team may instruct you to use our in-built *debug logging* feature. This will temporarily capture user interactions and other eagle.io specific information which will assist our team to find and fix the problem.

To use this feature:

1. Open the *Support request* dialog from the navigation bar and click the *Start debug logging* button.
2. Close the dialog (or refresh browser) and perform the interactions/navigation steps needed to reproduce the issue.
3. Re-open the *Support request* dialog and click the *Stop debug logging* button.
4. If any critical data has been captured, a log file will be automatically saved to your device.
5. Email the log file (along with a description of the issue) to [support@eagle.io](mailto:support@eagle.io)





The HTTP API has been designed to make it easy for developers to interact with [eagle.io](https://eagle.io) (https://eagle.io) from a 3rd party application or service.

Refer to the *Quick Start* (page 433) guide to get up and running. Be sure to review our recommended *Best Practices* (page 440) for working with the API.

*API Resources* (page 442) provides detailed descriptions and usage instructions for all API endpoints.

### 3.1 Quick Start

The [eagle.io](https://eagle.io) (<https://eagle.io>) API conforms to the design principles of Representational State Transfer (REST) ([http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer)).

Our API is designed to have predictable, resource-oriented URLs and to use HTTP response codes to indicate API errors. We use built-in HTTP features which can be understood by off-the-shelf HTTP clients.

Example code is available on [GitHub](https://github.com/eagle-io/api-examples) (<https://github.com/eagle-io/api-examples>).

### 3.1.1 Authentication

You authenticate to the eagle.io API by providing one of your *API keys* (page 272) in the **X-Api-Key** request header:

```
GET /api/v1/nodes/  
Host: api.eagle.io  
X-API-Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

You can manage your API keys from the account settings [API keys](#) (page 272) security tab. You can have multiple API keys active at one time. Your API keys carry many privileges, so be sure to keep them secret.

All API requests must be made over **HTTPS**. Calls made over plain HTTP will fail. You must authenticate for all requests.

### 3.1.2 HTTP Methods

The following HTTP methods are used to determine which action to take on a particular resource:

Method	Action
GET	Retrieve the current representation of a resource
POST	Create a new resource or action
PUT	Update a specific resource
DELETE	Delete a specific resource

---

**Note:** If your client does not support the required method the **X-HTTP-Method-Override** header can be added to a POST request to inform the server of the desired method.

---

```
POST /api/v1/nodes/5a2a5a129bc9b4035f906d70
Host: api.eagle.io
X-API-Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
X-HTTP-Method-Override: DELETE
```

### 3.1.3 HTTP Headers

The following HTTP headers are used by the eagle.io API:

Header	Value	Re-quired	Description
X-API-Key	YOUR_API_KEY	Yes	Unique API authentication key generated from <a href="#">account settings</a> (page 272)
Content-Type	application/json	Yes	All requests containing JSON data in the body must specify this header
Content-Encoding	gzip	No	Request body is gzip compressed
Accept-Encoding	gzip	No	Response will use gzip compression when content-length exceeds 1024 bytes
X-HTTP-Method-Override	PUT, DELETE	No	Clients not supporting PUT/DELETE methods can send via POST with this header

### 3.1.4 HTTP Status Codes

The eagle.io API will return appropriate [HTTP status codes](http://en.wikipedia.org/wiki/List_of_HTTP_status_codes) ([http://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](http://en.wikipedia.org/wiki/List_of_HTTP_status_codes)) for every request.

Code	Status	Description
200	OK	Completed successfully
201	Created	Resource created successfully
202	Operation accepted	Operation accepted but not yet complete
304	Not Modified	There was no new data to return
400	Bad Request	The request was invalid or cannot be otherwise served
401	Unauthorized	Authentication failed
403	Forbidden	No permission for HTTP Method or requested resource
404	Requested resource not found	Requested resource not found
405	Method Not Allowed	Resource does not support the requested HTTP Method
413	Request Too Large	Request body or response exceeds limit
429	Too Many Requests	Request has been rejected for exceeding usage limits
500	Internal Server Error	Error occurred processing the request
503	Service Unavailable	Service is temporarily unavailable

## Error and Status messages

Error and Status messages are always returned in a standard JSON format with the status code and message in the body of the response. An example error response:

```
{ "error": { "code": 405, "message": "Method not allowed for requested resource" } }
```

Status messages are returned for certain requests that have been accepted but do not return a resource in the response body:

```
{ "status": { "code": 202, "message": "Operation accepted but not yet complete" } }
```

### 3.1.5 HTTP Compression

HTTP compression is supported on both incoming and outgoing requests. Compression is highly recommended where large requests are being submitted.

We use the standard HTTP method of specifying compression encoding, with support for the gzip compression schema.

#### Incoming compression

In order to send compressed data to the eagle.io API it is necessary to send it with gzip encoding. Compress your data using gzip and add the **Content-Encoding** header to your request:

```
Content-Encoding: gzip
```

## Outgoing compression

To receive data from the eagle.io API in gzip compressed format simply add the **Accept-Encoding** header to your request:

```
Accept-Encoding: gzip
```

Any response exceeding 1024 bytes will be compressed and the **Content-Encoding** header added.

### 3.1.6 Resource Attributes

*API Resource* (page 442) requests commonly result in a response containing the current representation of one or more resources. The representations are a combination of attributes used to describe the individual resources.

Consumers of Resources should tolerate the addition of new attributes and variance in ordering of attributes with ease. Not all attributes appear in all contexts. It is generally safe to consider a nulled attribute, an empty set, and the absence of an attribute as the same thing.

Attributes are assigned specific *Data types* which are enforced for all requests and include:

Data type	Example	Description
<b>Num-ber</b>	12.4	Any number. Also expressed as <b>Int32</b> or <b>Int64</b> for integers and <b>Double</b> for double precision floating point numbers
<b>Boolean</b>	true	<i>true</i> or <i>false</i>
<b>String</b>	“Valve Open”	Any text surrounded in quotes
<b>Time</b>	“2014-09-24T19:48:32.000Z”	<i>ISO8601</i> (page 581) timestamp surrounded in quotes
<b>Ob-jectId</b>	“53ce0b6a0eba8dc469000000”	Identifier surrounded in quotes
<b>Array</b>	[“item1”, “item2”]	List of items. May contain any data type but generally not mixed
<b>Ob-ject</b>	{“key”: “value”}	Nested attributes
<b>Vari-able</b>		May contain various data types but usually specific for each resource _class

### 3.1.7 Request Arguments

Most *API Resource* (page 442) requests allow for arguments to be specified as part of the request URL. While some resources have *required* arguments, most are *optional* and are used to alter, refine or filter the response.

Arguments are specified using the standard *Query string* ([http://en.wikipedia.org/wiki/Query\\_string](http://en.wikipedia.org/wiki/Query_string)) format and should be appended to the resource endpoint, prefixed with a ‘?’.

```
/api/v1/nodes/?arg1=val1&arg2=val2&arg3=val3...
```

Certain resource request arguments may contain sub-arguments (denoted by indentation in arguments table). Sub-arguments should be included in parentheses with key/value separated by colons and multiple arguments semicolon delimited.

```
/api/v1/nodes/?arg1=val1(subarg1:subval1;subarg2:subval2...),val2,val3
```

All special characters should be [Url Encoded](http://en.wikipedia.org/wiki/Percent-encoding) (<http://en.wikipedia.org/wiki/Percent-encoding>).

The query string should not exceed 4096 bytes. A **413 Request Too Large** response will be returned for requests exceeding the size limit.

## Filter Argument

Certain *API Resource* (page 442) requests allow for an optional `filter` argument to be specified. When a filter is provided, only resources with attributes matching the filter criteria will be returned. If no criteria matches than an empty set will be returned.

The filter argument includes the attribute name with an operator and operand colon delimited and included in parentheses directly after the attribute name:

```
/?filter=attribute(operator:operand)
```

An example filter request for recently created nodes:

```
/api/v1/nodes/?filter=createdTime($gt:2015-01-01T00:00:00Z)
```

The following operators can be used when specifying filter criteria. Use is limited to specific attribute data types:

Operator	Supported data types	Description
<b>\$gt</b>	Number, Time	Greater than
<b>\$gte</b>	Number, Time	Greater than or equal to
<b>\$lt</b>	Number, Time	Less than
<b>\$lte</b>	Number, Time	Less than or equal to
<b>\$eq</b>	Number, Time, Boolean, String, ObjectId	Equal
<b>\$ne</b>	Number, Time, Boolean, String, ObjectId	Not equal
<b>\$match</b>	String	Text match

Multiple filter criteria can be specified using logical AND, OR operators. To specify an AND between criteria, use a comma character ‘,’. To specify an OR between criteria, use a pipe character ‘|’ [Url Encoded](http://en.wikipedia.org/wiki/Percent-encoding) (<http://en.wikipedia.org/wiki/Percent-encoding>) as ‘%7C’. The AND criteria is always evaluated before the OR criteria.

An example of a multi-criteria filter request for retrieving nodes with (name matching Batt\* AND isActive true) OR (createdTime greater than 2015-01-01T00:00:00Z):

```
/api/v1/nodes/?filter=name($match:Batt),isActive($eq:true)%7CcreatedTime(
↪$gt:2015-01-01T00:00:00Z)
```

---

**Note:** If the attribute data type is **Variable** you must explicitly specify the data type of the operand.

---

Example specifying currentValue operand as a Time:

```
/?filter=currentValue($eq:Time(2015-01-01T00:00:00Z))
```

### 3.1.8 Data Formats

The eagle.io API uses **JSON data format** (<http://json.org>) by default for all requests and responses. Certain resources allow for alternate data formats by specifying the **format** argument as part of the request URL:

```
/?format=csv
```

**Content-Type** is always included in the header of the response:

```
Content-Type: text/csv; charset=utf-8
```

### Request

**Content-Type** must be defined in the header of all requests that contain data:

```
Content-Type: application/json
```

The body of the request must be valid JSON or a **400 Bad Request** response will be returned. The complete HTTP request should not exceed 5MB before compression. A **413 Request Too Large** response will be returned for requests exceeding the size limit.

Example request for inserting historic data:

```
POST /api/v1/nodes/542a5a129bc9b4035f906d70/historic/
Host: api.eagle.io
X-API-Key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/json
```

```
{
  "docType": "jts",
  "version": "1.0",
  "data": [
    { "ts": "2014-10-07T01:25:00.000Z", "f": { "0": { "v": 10.0, "q": 100_
↪} } },
    { "ts": "2014-10-07T01:26:00.000Z", "f": { "0": { "v": 10.2, "q": 100_
↪} } }
  ]
}
```

## Response

Example response from a Nodes GET request:

```
HTTP/1.1 200 OK
Content-Length: 831
Content-Type: application/json; charset=utf-8
```

```
{
  "_class": "io.eagle.models.node.point.NumberPoint",
  "_id": "541a5a129bc9b4035f906d70",
  "createdTime": "2014-09-18T04:05:38.699Z",
  "currentStateId": "541a58dc9bc9b4035f906938",
  "currentTime": "2014-10-06T21:06:29.000Z",
  "currentValue": 157.7,
  "displayType": "VALUE",
  "format": "0.###",
  "isActive": true,
  "name": "Temperature",
  "oldestStateId": "541a58dc9bc9b4035f906938",
  "oldestTime": "2014-02-13T22:58:53.000Z",
  "oldestValue": 144.7,
  "ownerId": "52969365593a1a3a3200000f",
  "parentId": "541a59fb9bc9b4035f906d58",
  "previousTime": "2014-10-06T21:01:28.000Z",
  "previousValue": 164.1,
  "states": [
    {
      "name": "NORMAL",
      "index": 0,
      "categoryId": null,
      "_id": "541a58dc9bc9b4035f906938",
      "notify": "NEVER",
      "description": "",
      "isAlarm": false,
      "_class": "io.eagle.models.node.point.PointState"
    }
  ],
  "statesType": "RANGE",
  "units": "",
  "workspaceId": "536884ecb5a76fd5d3000014"
}
```

### 3.1.9 Resource Ids

Most resources must be accessed with an *id*. The automatically assigned id is stored as the *\_id* attribute of the resource. Nodes can optionally be assigned a unique *customId* which can also be used to access the resource prepended with '@'. Learn how to assign a customId.

Example accessing a node by its automatically assigned *\_id*

```
/api/v1/nodes/536884ecb5a76fd5d3000014
```

Example accessing a node by its *customId*

```
/api/v1/nodes/@sensor_254
```

### 3.1.10 Versioning

The API version number is specified as part of the URL. **v1** is the currently supported version and has a base URL:

```
https://api.eagle.io/api/v1/
```

As we make updates and additions to the eagle.io API, we will only make changes, add new features and/or implement fixes to the current released API that do not affect backward compatibility.

### 3.1.11 Usage Limits

API usage is rate limited by IP Address to prevent misuse. If you exceed more than 350 requests per 5 minute window the request will be refused and you will receive a **429 Too Many Requests** response.

## 3.2 Best Practices

As you build with the [eagle.io](https://eagle.io) (<https://eagle.io>) API, take note of the following best practices to ensure you develop safe, reliable and efficient code.

### 3.2.1 Enforce Security

All requests must be made over **HTTPS**.

*API keys* (page 272) should be kept secret and the *Read only* permission assigned to keys that only need to perform read operations (GET requests). For additional security, you can restrict the API key to certain Workspaces by setting the *Access level* to *Limited Workspaces*.

### 3.2.2 Use Compression for large requests

Take advantage of *HTTP compression* (page 435) for sending and receiving large requests.

### 3.2.3 Limit Size and Rate of requests

When working with *time series* data, ensure you limit the size of the requests for both extraction and insertion. If the request or response is too large or exceeds our processing time threshold, the **413 Request Too Large** error response will be returned.



The body of the request should not exceed 5MB before compression. The query string should not exceed 4096 bytes. A **413 Request Too Large** response will be returned for requests exceeding the size limit.

You should limit the rate of requests to the API to ensure you do not exceed the *usage limits* (page 440).

### 3.2.4 Handle Errors and Status responses

Responses will always use *HTTP Status Codes* (page 434) to indicate the result of the request. Be prepared to handle unexpected errors which may arise from:

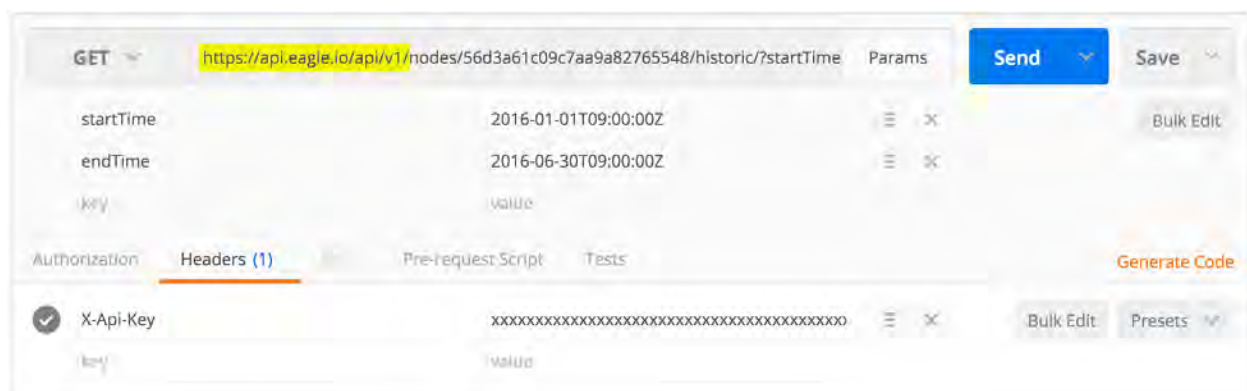
- Large or long running requests
- Moved, renamed or deleted resources
- API key permission and account changes
- Usage limits exceeded
- Service temporarily unavailable

### 3.2.5 Discover Node Ids from the user interface

Node Ids are shown in the associated Node properties dialog. You can also assign your own unique Custom Id to every node for use with the API.

### 3.2.6 Use Development Tools for testing

Use development tools such as [cURL](http://curl.haxx.se) (<http://curl.haxx.se>) or [Postman](http://www.getpostman.com) (<http://www.getpostman.com>) to test API requests and responses. We recommend [Postman](http://www.getpostman.com) (<http://www.getpostman.com>) for its simple to use, powerful user interface.



### 3.2.7 Synchronise data efficiently

When synchronising data to an external system, ensure you only query for data that has changed since your previous request. By specifying a *timeQuery* of *MODIFIED*, only records added or changed within your query range will be returned.

## 3.3 API Resources

API Resources are the URL endpoints for interacting with the system. Interactions may include creating, reading, updating or deleting an asset, extracting or inserting historic data, or performing operations such as triggering data acquisition, control operations or device configuration.

Refer to the *Quick Start* (page 433) guide for instructions on using the API.

### 3.3.1 Nodes

Nodes are the container objects that store all data and configuration for your projects. There are various *types of nodes* (page 49) including: Workspaces, Groups, Locations, Dashboards, Attachments, Sources and Parameters.

#### Quick Links

- *Node References* (page 443)
- *Node Attributes* (page 444)
- *Retrieve all nodes* (page 448)
- *Retrieve a node* (page 451)
- *Create a node* (page 452)
- *Update a node* (page 453)
- *Delete a node* (page 463)
- *Clone a node or create instance from template* (page 463)
- *Acknowledge node alarms* (page 465)
- *Clear node alarms* (page 465)
- *Trigger an acquisition* (page 466)
- *Retrieve user and group notification preferences* (page 467)
- *Update user notification preferences* (page 469)
- *Subscribe users and groups to notifications* (page 471)
- *Unsubscribe users and groups from notifications* (page 472)

- *Retrieve Workspace user and group permissions* (page 473)
- *Add or Update Workspace user and group permissions* (page 475)
- *Remove users and groups from a Workspace* (page 477)
- *Retrieve node historic data* (page 478)
- *Retrieve node historic chart* (page 481)
- *Retrieve node report* (page 483)
- *Update Parameter or Location historic data* (page 486)
- *Update Data Source historic data* (page 487)
- *Update node historic data with a single value* (page 489)
- *Delete node historic data* (page 490)
- *Retrieve node events* (page 490)
- *Create node events* (page 492)
- *Delete node events* (page 493)

## Node References

A Node can be referenced programmatically using either a **Node Id** (globally unique) or a user-defined **Custom Id** (unique to your account). These identifiers can be found (and **Custom Id** configured) at the bottom of the Properties for each Node.

Properties demo/01 Random/Location 1/Data Source/Pump

General States Chart Historic Processing Security

Display type Value ?  
Default data format for map, charts and tabular data.

Multiplier 1.0 ?

Offset 0 ?

Units ?

Format 0.### ?

Node Id: 580d5b60ee0613528391e2e5

@ sensor\_254

Save Cancel

## Node Id

A **Node Id** is a 24-character alphanumeric string that uniquely identifies a Node in eagle.io. It is generated by the system and cannot be changed.

## Custom Id

A **Custom Id** is a 255-character user-defined string that uniquely identifies a Node in your account. A **Custom Id** is prefixed with the @ symbol and must not contain the following special characters: dot ., underscore \_ and hyphen -. In the example above, the **Custom Id** has been set to `sensor_254`.

## Node Attributes

Consumers of Nodes should tolerate the addition of new attributes and variance in ordering of attributes with ease. Not all attributes appear in all contexts. It is generally safe to consider a nulled attribute, an empty set, and the absence of an attribute as the same thing.

Standard attributes	Type	Update	Description
<b>_class</b>	String		Identifies resource type: <i>io.eagle.models.node</i> .*
<b>_id</b>	Object-Id		Unique identifier for this node
<b>alarms</b>	Object	Yes	<i>Alarm configuration</i> (page 454) and status
<b>created-Time</b>	Time		<i>ISO8601</i> (page 581) timestamp the node was created
<b>customId</b>	String	Yes	Optional user-assigned identifier for this node. Unique to owner (account)
<b>isActive</b>	Boolean		Flag to indicate if node is active
<b>isPublic</b>	Boolean	Yes	Flag to indicate if <i>Public access</i> (page 221) is enabled
<b>metadata</b>	Array	Yes	Array of metadata fields and values for a location
<b>name</b>	String	Yes	Name of node. <i>Restrictions apply</i> (page 15).
<b>ownerId</b>	Object-Id		Unique owner _id
<b>parentId</b>	Object-Id		Parent node _id (not included in Workspace)
<b>publicId</b>	String		Unique identifier used in public URL when <i>isPublic</i> is enabled. eg. <a href="https://public.eagle.io/public/dash/w31s3dndr408wcq">https://public.eagle.io/public/dash/w31s3dndr408wcq</a>
<b>workspaceId</b>	Object-Id		Associated Workspace _id (not set on Workspace node)

Attachment attributes	Type	Update	Description
<b>description</b>	String	Yes	Description of attachment as set by user.
<b>fileSize</b>	Int64		Size of attachment in bytes
<b>fileUrl</b> <sup>3</sup>	String		Attachment URL. Requires <i>ATTACHMENT_READ</i> permission. Optionally specify an <i>expiry</i> period for the URL.
<b>filePreviewUrl</b> <sup>3</sup>	String		Preview URL. Only included if the attachment is a valid media file. Requires <i>ATTACHMENT_READ</i> permission. Optionally specify an <i>expiry</i> period for the URL. Note: image dimensions may change without notice.
<b>mime-Type</b>	String		Mime-type of the attachment

<sup>3</sup> Certain attributes can not be used in filter criteria.

Parameter & Location attributes	Type	Update	Description
<b>coordinateSourceId</b>	ObjectId		Data Source _id used to set location coordinates. Applies to Locations only.
<b>chart</b>	Object	Yes	<a href="#">Parameter chart configuration</a> (page 458)
<b>controlValue</b>	Variable	Yes	Pending control value. Type inherited from _class.
<b>controlValueMaximum</b>	Double	Yes	Maximum control value when controlValueRestriction set to RANGE
<b>controlValueMinimum</b>	Double	Yes	Minimum control value when controlValueRestriction set to RANGE
<b>controlValueRestriction</b>	String	Yes	Control value is restricted to: <i>[NONE, RANGE, STATES]</i>
<b>currentQuality</b>	Int32		Quality code associated with current value
<b>currentStateId</b>	ObjectId		State _id associated with current value
<b>currentTime</b>	Time		<a href="#">ISO8601</a> (page 581) timestamp of the current value
<b>currentValue</b>	Variable		Latest value with multiplier and offset applied. Type inherited from _class
<b>displayType</b>	String	Yes	Default display type of parameter: <i>[VALUE, STATE]</i>
<b>format</b>	String	Yes	Formatting to apply to displayed value
<b>imageSourceId</b>	ObjectId		Attachment Source _id used to set location image. Applies to Locations only.
<b>multiplier</b>	Double	Yes	Value to multiply rawValue and historic data on extraction. Applies to number parameters only
<b>offset</b>	Double	Yes	Value to add to rawValue and historic data on extraction. Applies to number parameters only
<b>oldestQuality</b>	Int32		Quality code associated with oldest value
<b>oldestStateId</b>	ObjectId		State _id associated with oldest value
<b>oldestTime</b>	Time		<a href="#">ISO8601</a> (page 581) timestamp of the oldest value
<b>oldestValue</b>	Variable		Oldest value. Type inherited from _class
<b>previousQuality</b>	Int32		Quality code associated with oldest value
<b>previousStateId</b>	ObjectId		State _id associated with previous value
<b>previousTime</b>	Time		<a href="#">ISO8601</a> (page 581) timestamp of the previous value
<b>previousValue</b>	Variable		Previous value. Type inherited from _class
<b>rating</b>	Object	Yes	<a href="#">Rating parameter configuration</a> (page 461)
<b>rawValue</b>	Double		Latest raw value. Applies to number parameters only
<b>states</b>	Array	Yes	Array of <a href="#">state configuration</a> (page 456) objects
<b>statesType</b>	String	Yes	States evaluation mode: <i>[RANGE, DISCRETE]</i>
<b>units</b>	String	Yes	Units to display with value
<b>updatedTime</b>	Time		<a href="#">ISO8601</a> (page 581) timestamp the historic data was last updated.

Source attributes	Type	Update	Description
<b>currentAttachmentId</b>	ObjectId		Current attachment _id associated with Attachment Source
<b>currentStatus</b>	String		Current status: <i>[DISABLED, FAILED, IDLE, SCHEDULED, QUEUED, CONNECTING, DOWNLOADING, PROCESSING, RETRYING, CONFIGURING, CONTROLLING, REMOTE]</i>
<b>clockSync</b>	Boolean		Flag to indicate if data logger clock is being synchronized with server
<b>firmwareVersion</b>	String		Firmware version in-use by data logger
<b>isEnabled</b>	Boolean	Yes	Flag to enable or disable the Source
<b>lastAcquireRecordCount</b>	Int32		Number of records acquired on last acquisition
<b>lastCommsAttempt</b>	Time		<a href="#">ISO8601</a> (page 581) timestamp of the last communications attempt
<b>lastCommsSuccess</b>	Time		<a href="#">ISO8601</a> (page 581) timestamp of the last successful communication with Source
<b>model</b>	String		Model of data logger in-use
<b>pakBusAddress</b>	Int32	Yes	PakBus address of Campbell data logger
<b>pakBusGatewayAddress</b>	Int32	Yes	PakBus gateway address of Campbell data logger
<b>progName</b>	String		Name of program currently loaded in Campbell data logger
<b>quietMaxSeconds</b>	Int32	Yes	Maximum period in seconds between communication updates before triggering an Outdated alarm (non-scheduled collection only). <i>120 - 31536000</i> seconds
<b>scheduleType</b>	String		Type of Schedule for automatic collection: <i>[SERVER, REMOTE, NONE]</i>
<b>timezone</b>	String	Yes	<a href="#">Timezone</a> (page 581) of Source data.
<b>timezoneAdjustForDst</b>	Boolean	Yes	Flag to indicate if acquired data is adjusted for DST in selected timezone
<b>totalCommsAttempt</b>	Int32		Total count of communications attempts
<b>totalCommsSuccess</b>	Int32		Total count of successful communications

## Retrieve all nodes

Retrieve a list of all nodes available to the authenticated API key. Optionally *filter* by attribute values.

## Arguments

Argument	Example	Description
<b>attr</b>	<code>_id,slug</code>	<i>Optional.</i> Comma delimited list of attributes to include in response
<b>type</b>	<code>TREE</code>	<i>Optional - Default is LIST<sup>4</sup></i>   <b>LIST</b> returns a flat list of nodes   <b>TREE</b> returns hierarchical list of nodes (only available for requests that return less than 5000 nodes)
<b>filter</b>	<code>isActive(\$eq:true)</code>	<i>Optional.</i> <a href="#">Filter</a> (page 437) the records based on attribute value(s)
<b>limit</b>	<code>100</code>	<i>Optional - Default is 50000</i> Maximum number of records to be returned. 0-50000
<b>skip</b>	<code>50</code>	<i>Optional.</i> Skip the first <i>n</i> records returned. Can be used with <code>limit</code> to paginate results
<b>sort</b>	<code>created-Time(DESC)</code>	<i>Optional.</i> Comma delimited list of attributes to sort by. Optionally include sort direction in parentheses or default to ASC: <i>[ASC, DESC]</i>
<b>expiry</b>	<code>60</code>	<i>Optional.</i> Expiry period in minutes for any included resource links. ie. <i>fileUrl</i> for Attachments. Default is 0 (no expiry). Expired resource links will return 403 Forbidden.

## Request

```
GET /api/v1/nodes
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
[
  {
    "_class": "io.eagle.models.node.Workspace",
    "_id": "536884ecb5a76fd5d3000014",
    "createdTime": "2014-05-06T06:45:00.061Z",
    "isActive": true,
    "name": "My Workspace",
    "ownerId": "52969365593a1a3a3200000f"
```

(continues on next page)

<sup>4</sup> When `TREE` type is specified the filter, limit, skip and sort arguments are not allowed. A 413 'request too large' response will be returned if there are more than 5000 nodes when using type `TREE`.



(continued from previous page)

```

    },
    {
      "_class": "io.eagle.models.node.Group",
      "_id": "53eada0ada91110000000046",
      "createdTime": "2014-08-13T03:22:50.489Z",
      "isActive": true,
      "name": "Australia",
      "ownerId": "52969365593a1a3a3200000f",
      "parentId": "536884ecb5a76fd5d3000014",
      "workspaceId": "536884ecb5a76fd5d3000014"
    },
    {
      "_class": "io.eagle.models.node.location.Location",
      "_id": "542cbd18815043850e995f84",
      "alarms": {
        "stateAlarm": {
          "status": {
            "acknowledgeComment": null,
            "acknowledgeTime": null,
            "acknowledgeUsername": null,
            "categoryId": null,
            "message": null,
            "alarmState": "CLEARED"
          },
          "_class": "io.eagle.models.node.alarm.StateAlarm"
        }
      },
      "createdTime": "2014-10-02T02:48:56.186Z",
      "displayType": "VALUE",
      "isActive": true,
      "metadata": [],
      "name": "New Location",
      "ownerId": "52969365593a1a3a3200000f",
      "parentId": "53eada0ada91110000000046",
      "states": [
        {
          "name": "NORMAL",
          "index": 0,
          "categoryId": null,
          "_id": "542cbc4f815043850e995ebb",
          "threshold": {
            "_class": "io.eagle.util.geo.GeoPoint"
          },
          "notify": "NEVER",
          "description": null,
          "isAlarm": null,
          "_class": "io.eagle.models.node.point.PointState"
        }
      ],
      "statesType": "RANGE",
      "workspaceId": "536884ecb5a76fd5d3000014"
    }
  ]
}

```

(continues on next page)

(continued from previous page)

]

Example response when type argument is TREE. Descendants are wrapped in a children array:

```
[
  {
    "_class": "io.eagle.models.node.Workspace",
    "_id": "536884ecb5a76fd5d3000014",
    "createdTime": "2014-05-06T06:45:00.061Z",
    "isActive": true,
    "name": "My Workspace",
    "ownerId": "52969365593a1a3a3200000f",
    "children": [
      {
        "_class": "io.eagle.models.node.Group",
        "_id": "53eada0ada91110000000046",
        "createdTime": "2014-08-13T03:22:50.489Z",
        "isActive": true,
        "name": "Australia",
        "ownerId": "52969365593a1a3a3200000f",
        "parentId": "536884ecb5a76fd5d3000014",
        "workspaceId": "536884ecb5a76fd5d3000014",
        "children": [
          {
            "_class": "io.eagle.models.node.location.Location",
            "_id": "542cbd18815043850e995f84",
            "alarms": {
              "stateAlarm": {
                "status": {
                  "acknowledgeComment": null,
                  "acknowledgeTime": null,
                  "acknowledgeUsername": null,
                  "categoryId": null,
                  "message": null,
                  "alarmState": "CLEARED"
                },
                "_class": "io.eagle.models.node.alarm.
↪StateAlarm"
              }
            },
            "createdTime": "2014-10-02T02:48:56.186Z",
            "displayType": "VALUE",
            "isActive": true,
            "metadata": [],
            "name": "New Location",
            "ownerId": "52969365593a1a3a3200000f",
            "parentId": "53eada0ada91110000000046",
            "states": [
              {
                "name": "NORMAL",
                "index": 0,
                "categoryId": null,
```

(continues on next page)

```

    "_id": "542cbc4f815043850e995ebb",
    "threshold": {
      "_class": "io.eagle.util.geo.GeoPoint"
    },
    "notify": "NEVER",
    "description": null,
    "isAlarm": null,
    "_class": "io.eagle.models.node.point.
↪PointState"
  },
  ],
  "statesType": "RANGE",
  "workspaceId": "536884ecb5a76fd5d3000014",
  "children": []
}
]
}
]
}
]

```

Argument	Example	Description
<b>attr</b>	<code>_id, class</code>	<i>Optional.</i> Comma delimited list of attributes to include in response
<b>expiry</b>	60	<i>Optional.</i> Expiry period in minutes for any included resource links. ie. <i>fileUrl</i> for Attachments. Default is 0 (no expiry). Expired resource links will return 403 Forbidden.

```
GET /api/v1/nodes/:id
```

```
/api/v1/nodes/536884ecb5a76fd5d3000014
```

```
/api/v1/nodes/@workspace-01
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "_class": "io.eagle.models.node.Workspace",
  "_id": "536884ecb5a76fd5d3000014",
  "createTime": "2014-05-06T06:45:00.061Z",
  "isActive": true,
  "name": "My Workspace",
  "ownerId": "52969365593a1a3a3200000f",
  "customId": "workspace-01"
}
```

## Create a node

Create a node (currently limited to creation of *Workspace* and *Group*). You must specify both the *\_class* and a unique *name*. Creating a *Group* also requires a valid *parentId*.

---

**Note:** Required API key permission: *Modify*

---

## Request

```
POST /api/v1/nodes
```

### Example creating a Workspace

```
{
  "_class": "io.eagle.models.node.Workspace",
  "name": "New Workspace"
}
```

### Example creating a Folder

```
{
  "_class": "io.eagle.models.node.Group",
  "name": "New Folder",
  "parentId": "5a554eed0b64aabe5738da86"
}
```

## Response

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
```

```
{
  "_class": "io.eagle.models.node.Workspace",
  "_id": "5ae7c3d03549e867c718ff97",
  "createTime": "2018-05-01T01:33:04.830Z",
  "isActive": true,
  "metadata": [],
  "name": "New Workspace",
  "ownerId": "56d3a61a09c7aa9a82765540"
}
```

## Update a node

Update a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. Updates are limited to the attributes listed with the *Update* flag. Multiple attributes can be updated in a single request.

The updated node will be returned in the response if the request is successful. You can optionally limit the returned attributes by specifying the **attr** argument.

---

**Note:** Required API key permission: *Modify*

---

## Arguments

Argument	Example	Description
<b>attr</b>	<code>_id,_class</code>	<i>Optional.</i> Comma delimited list of attributes to include in successful response

## Request

```
PUT /api/v1/nodes/:id
```

```
{
  "name": "My Renamed Workspace"
}
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "_class": "io.eagle.models.node.Workspace",
  "_id": "536884ecb5a76fd5d3000014",
  "createdTime": "2014-05-06T06:45:00.061Z",
  "isActive": true,
  "name": "My Renamed Workspace",
  "ownerId": "52969365593a1a3a3200000f"
}
```

## Complex attribute updates

Certain attributes such as alarms and states have specific update requirements:

- *Alarm configuration* (page 454)
- *States configuration* (page 456)
- *Parameter chart configuration* (page 458)
- *Rating configuration* (page 461)

## Alarm configuration

Alarm configuration can be included in the update request for Location, Source and Parameter nodes. You must specify each alarm type as a nested attribute containing the *config* to be updated. Only changed attributes need to be specified.

Alarm config attributes	Type	Description
<b>isEnabled</b>	Boolean	Enable or disable the alarm. Overload alarm can not be disabled.
<b>categoryId</b>	ObjectId	Id of <i>owner category</i> (page 508) to assign to this alarm. Required when <i>notify</i> option is not <i>NEVER</i>
<b>notify</b>	String	When notifications should be triggered: <i>[NEVER, ALWAYS, AWAY_FROM_NORMAL, TOWARDS_NORMAL]</i> .
<b>message-Away</b>	String	<i>Optional - Only valid for outdatedAlarm.</i> Custom message to send when the alarm becomes active. Maximum of 255 characters. Leave empty for default message.
<b>message-Towards</b>	String	<i>Optional - Only valid for outdatedAlarm.</i> Custom message to send when the alarm becomes inactive. Maximum of 255 characters. Leave empty for default message.
<b>quality-Types</b>	Array	Only valid for <i>qualityAlarm</i> . List of quality types that will trigger this alarm: <i>[GOOD, BAD, UNCERTAIN]</i> .
<b>exclude-BeforeSeconds</b>	Int32	<i>Optional - Default is null.</i> Only valid for <i>dataAlarm</i> . Exclude data older than this duration and raise alarm if encountered. <i>120 - 315360000</i> seconds.
<b>exclude-AfterSeconds</b>	Int32	<i>Optional - Default is null.</i> Only valid for <i>dataAlarm</i> . Exclude data newer than this duration and raise alarm if encountered. <i>120 - 315360000</i> seconds.

Example:

```
{
  "alarms": {
    "communicationsAlarm": {
      "config": {
        "isEnabled": true,
        "notify": "ALWAYS",
        "categoryId": "52969367593a1a3a32000012"
      }
    },
    "qualityAlarm": {
      "config": {
        "isEnabled": true,
        "qualityTypes": ["BAD", "UNCERTAIN"]
      }
    },
    "configurationAlarm": { "config": { "isEnabled": false } }
  }
}
```

The alarm types available are specific to the type of node being updated:

Alarm types	Supported nodes
<b>communicationsAlarm</b>	Sources
<b>configurationAlarm</b>	Sources
<b>controlAlarm</b>	Control parameters
<b>dataAlarm</b>	Sources
<b>outdatedAlarm</b> <sup>1</sup>	Sources
<b>overloadAlarm</b>	Sources
<b>processAlarm</b>	Processor Sources, Process Parameters
<b>qualityAlarm</b>	Locations, Parameters
<b>stateAlarm</b> <sup>2</sup>	Locations, Parameters

## States configuration

States configuration can be included in the update request for Number and Text Parameter nodes. Number parameters can use either *DISCRETE* or *RANGE* states. Text parameters are limited to *DISCRETE* states only.

States can be reset to parameter defaults by updating the *states* attribute with *null*.

**Update** existing states by providing the *\_id* attribute for each state in the request. Any states without the *\_id* attribute will be ignored. Only changed attributes need to be specified.

If no states in the request contain an *\_id* attribute the update will **Replace** the existing states. **RANGE** states for number parameters must always be specified in *ascending* order based on *threshold* and there must be one state with the *name* attribute set to *NORMAL*.

Refer to [Parameter states](#) (page 147) for further information.

---

**Note:** State type for number parameters can be changed with the *statesType* attribute. When changing *statesType* you should specify new *states* configuration or default states will be applied.

---

---

<sup>1</sup> outdatedAlarm *Maximum timeout* is set via the node *quietMaxSeconds* attributes.

<sup>2</sup> stateAlarm has no direct configuration. Modify *states* to change state alarm behaviour.



State attributes	Type	Description
<b>name</b>	String	<i>Required.</i> Unique name assigned to the state
<b>threshold</b>	Variable	<i>Required.</i> Unique threshold value for the state. Number parameters require a <i>Double</i> . Text parameters require a <i>String</i> .
<b>occurrences</b>	Int32	<i>Optional - Default is 1.</i> Minimum number of repeat values equal or beyond threshold required to trigger state: <i>1-15</i> .
<b>isAlarm</b>	Boolean	<i>Optional - Default is FALSE.</i> Flag to indicate if this state should raise an alarm. Not valid for <i>NORMAL RANGE</i> state.
<b>notify</b>	String	<i>Optional - Default is NEVER.</i> When notifications should be triggered: <i>[NEVER, ALWAYS, AWAY_FROM_NORMAL, TOWARDS_NORMAL]</i> . The <i>NORMAL RANGE</i> state is restricted to: <i>[NEVER, ALWAYS]</i>
<b>categoryId</b>	ObjectId	Id of <i>owner category</i> (page 508) to assign to this state. Required when <i>notify</i> option is not <i>NEVER</i>
<b>messageAway</b>	String	<i>Optional.</i> Custom message to send when the state becomes active (away from normal). Maximum of 255 characters. Leave empty for default message.
<b>messageTowards</b>	String	<i>Optional.</i> Custom message to send when the state becomes inactive (towards normal). Maximum of 255 characters. Leave empty for default message.
<b>qualityCode</b>	Int32	<i>Optional - Default is null.</i> Quality code to apply to acquired data that matches this state: <i>0-65535</i> .

Example replacing DISCRETE states

```
{
  "statesType": "DISCRETE",
  "states": [
    {
      "name": "OFF",
      "threshold": 0
    },
    {
      "name": "ON",
      "threshold": 1,
      "occurrences": 1,
      "isAlarm": true,
      "notify": "ALWAYS",
      "categoryId": "52969367593a1a3a32000012",
      "messageAway": "PUMP IS RUNNING",
      "messageTowards": "PUMP IS OFF",
      "qualityCode": 192
    }
  ]
}
```

### Example updating existing DISCRETE states

```
{
  "statesType": "DISCRETE",
  "states": [
    {
      "_id": "52969367593a1a3a32000091",
      "name": "SIREN OFF"
    },
    {
      "_id": "52969367593a1a3a32000092",
      "name": "SIREN ON",
      "notify": "NEVER"
    }
  ]
}
```

### Example replacing RANGE states

```
{
  "statesType": "RANGE",
  "states": [
    {
      "name": "LOW",
      "threshold": 20
    },
    {
      "name": "NORMAL"
    },
    {
      "name": "HIGH",
      "threshold": 50
    },
    {
      "name": "CRITICAL",
      "threshold": 100,
      "occurrences": 2,
      "isAlarm": true,
      "notify": "ALWAYS",
      "categoryId": "52969367593a1a3a32000012",
      "qualityCode": 192
    }
  ]
}
```

## Parameter chart configuration

Parameter chart configuration is used when displaying parameters as series on automatically generated charts. It can be included in the update request for Parameter nodes. Only changed attributes need to be specified.

Parameter chart config attributes	Type	Description
<b>_class</b>	String	Determines chart series type (Line or Column): <i>[io.eagle.models.node.point.chart.LineChart, io.eagle.models.node.point.chart.ColumnChart]</i>
<b>stateThresholds</b>	String	Threshold lines to display (when in y-axis range): <i>[NONE, ALL, ALARM, NONALARM, USER_NOTIFICATION]</i>
<b>markerType</b>	String	Marker type: <i>[AUTOMATIC, CIRCLE, SQUARE, DIAMOND, TRIANGLE, TRIANGLE-DOWN]</i>
<b>markerSize</b>	Int32	Marker size (0 is Hidden): 0-6.
<b>qualityStyle</b>	String	Specify how quality colors are display on chart: <i>[NONE, MARKER, MARKER_HOVER, MARKER_LINE, MARKER_FILL]</i>
<b>shadow</b>	Boolean	<i>Optional - default is false.</i> Drop shadow effect
<b>primaryColor</b>	String	<i>Optional - default is AUTOMATIC.</i> Hex color code (eg. <i>#ff3399</i> ) or <i>AUTOMATIC</i> used as primary color.
<b>secondaryColor</b>	String	<i>Optional - default is AUTOMATIC.</i> Hex color code (eg. <i>#ffffff</i> ) or <i>AUTOMATIC</i> used as secondary color when <i>fillStyle</i> is a gradient.
<b>fillStyle</b>	String	Fill style: <i>[NONE, SOLID, LINEAR_TOP, LINEAR_BOTTOM, LINEAR_LEFT, LINEAR_RIGHT, PIPE_VERTICAL, PIPE_HORIZONTAL, RADIAL_INSIDE, RADIAL_OUTSIDE]</i>
<b>fillOpacity</b>	Int32	Fill opacity (0 is Transparent): 0-100.
<b>lineWidth</b>	Int32	Series line width (0 is Hidden): 0-6.
<b>lineType</b>	String	Line type (Line series only): <i>[NORMAL, STEP_LEFT, STEP_CENTER, STEP_RIGHT, SPLINE]</i>
<b>lineStyle</b>	String	Line style (Line series only): <i>[SOLID, SHORTDASH, SHORTDOT, SHORTDASHDOTDOT, DOT, SHORTDASHDOTDOT, DOT, DASH, LONGDASH, DASHDOT, LONGDASHDOT,</i>
<b>3.3. API Resources</b>		<i>459</i>

Example Line series:

```
{
  "chart": {
    "_class": "io.eagle.models.node.point.chart.LineChart",
    "stateThresholds": "ALL",
    "markerType": "AUTOMATIC",
    "markerSize": 2,
    "qualityStyle": "MARKER_HOVER",
    "shadow": false,
    "primaryColor": "AUTOMATIC",
    "fillStyle": "NONE",
    "fillOpacity": 75,
    "lineWidth": 1,
    "lineType": "NORMAL",
    "lineStyle": "SOLID",
    "aggregation": {
      "mode": "AUTOMATIC"
    }
  }
}
```

Example updating to stepped Line series with custom gradient fill:

```
{
  "chart": {
    "_class": "io.eagle.models.node.point.chart.LineChart",
    "primaryColor": "#33ff00",
    "secondaryColor": "#ff0000",
    "fillStyle": "LINEAR_TOP",
    "fillOpacity": 100,
    "lineType": "STEP_LEFT"
  }
}
```

Example Column series with hourly totals:

```
{
  "chart": {
    "_class": "io.eagle.models.node.point.chart.ColumnChart",
    "markerSize": 0,
    "primaryColor": "AUTOMATIC",
    "fillStyle": "SOLID",
    "fillOpacity": 100,
    "pointPlacementType": "ON",
    "groupType": "NORMAL",
    "aggregation": {
      "mode": "CUSTOM",
      "type": "TOTAL",
      "period": "FIXED",
      "interval": "1H",
      "baseTime": "D"
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    }
  }

```

## Rating configuration

Rating configuration can be included in the update request for *Rating Parameter* (page 156) nodes. An *inputNodeId* is required which specifies the Number Parameter to use as the input for the rating calculations. Multiple ratings can be added, with the *startTime* used to determine the data range each rating will apply.

**Note:** Rating configuration updates will **replace** any existing ratings and trigger historic data to be recalculated for this parameter.

Rating attributes	Type	Description
<b>inputNodeId</b>	ObjectId	Node <i>_id</i> to be used as the input for the rating calculations. Must be a Number parameter in the same Workspace as this Rating parameter.
<b>ratings</b>	Array	Rating table and rating equation objects (as documented below).

Rating table	Type	Description
<b>startTime</b>	Time	<i>Required.</i> <i>ISO8601</i> (page 581) timestamp of the start range this rating will be applied. End range is automatically set to the start-Time of the next most recent rating or will continue to apply to new data if no other ratings are specified.
<b>table</b>	Array	Lookup table for rating calculations with each row specified as an array of <i>input</i> and <i>result</i> . Values that fall between each input lookup will be derived by linear interpolation.
<b>[&lt;input&gt;, &lt;result&gt;]</b>	Array	<i>Required.</i> The numeric value of the input node to match and corresponding result.

Rating equation	Type	Description
<b>startTime</b>	Time	<i>Required.</i> <i>ISO8601</i> (page 581) timestamp of the start range this rating will be applied. End range is automatically set to the startTime of the next most recent rating or will continue to apply to new data if no other ratings are specified.
<b>equation-Type</b>	String	<i>POLYNOMIAL</i> is currently the only supported rating equation.
<b>coefficients</b>	Object	Map of <i>coefficient</i> key and corresponding <i>value</i> . The coefficient keys must be single letters, ordered alphabetically.

Example rating configuration:

```
{
  "rating": {
    "inputNodeId": "5ec206f9309acc31a896c3b9",
    "ratings": [
      {
        "startTime": "2020-01-01T00:00:00Z",
        "table": [
          [0.1, 9.8],
          [0.5, 40],
          [1.2, 120]
        ]
      },
      {
        "startTime": "2020-10-24T11:00:00Z",
        "table": [
          [0.1, 7.8],
          [0.2, 18.3],
          [0.6, 44.2],
          [1.1, 130.4]
        ]
      },
      {
        "startTime": "2020-11-01T11:00:00Z",
        "equationType": "POLYNOMIAL",
        "coefficients": {
          "a": 1.1,
          "b": 2.5,
          "c": 1.115,
          "d": 4.01,
          "e": 0.85
        }
      }
    ]
  }
}
```

In this example, Discharge is calculated using 2 rating lookup tables and 1 rating polynomial. The *inputNodeId* is set to the *\_id* of the *Water Level* parameter. A water level value of **0.1** will result in an output value of **9.8** for data at the beginning of 2020. Water level **0.1** will result in an output value of **7.8** for data in the range starting 24th October. From 1st November onwards, a polynomial equation will be applied using the coefficients **a, b, c, d, e**.

## Delete a node

Delete a node by its **id** including all child nodes. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'.

**Note:** This will permanently delete the node including all historic data and events. Required API key permission: *Modify*

## Request

```
DELETE /api/v1/nodes/:id
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 200,
    "message": "Success"
  }
}
```

## Clone a node or create instance from template

Clone a node or create an *instance* (page 62) from a *template* (page 60) by its *id*. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'.

When the node being cloned is not a Workspace you must specify a valid *parentId* as the destination for the clone/create operation. Optionally include *name* and *metadata* to be assigned to the newly created node.

**Note:** Not available for TRIAL accounts. Required API key permission: *Modify*. Cloning a workspace requires the API key to have 'All workspaces' access level.

## Arguments

Argument	Example	Description
<b>attr</b>	<code>_id,slug</code>	<i>Optional.</i> Comma delimited list of attributes to include in response
<b>type</b>	<code>TREE</code>	<i>Optional - Default is LIST</i> <sup>4</sup>   <b>LIST</b> returns a flat list of nodes   <b>TREE</b> returns hierarchical list of nodes
<b>expiry</b>	<code>60</code>	<i>Optional.</i> Expiry period in minutes for any included resource links. ie. <i>fileUrl</i> for Attachments. Default is 0 (no expiry). Expired resource links will return 403 Forbidden.

## Request

```
POST /api/v1/nodes/:id/clone
```

```
{
  "parentId": "5a554eed0b64aabe5738da86"
  "name": "Copy of Location 1",
  "metadata": [
    {
      "name": "Site ID",
      "value": "abc123"
    }
  ]
}
```

## Response

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
```

```
[
  {
    "_id": "5ae809d73549e867c71900db",
    "_class": "io.eagle.models.node.location.Location",
    "parentId": "5a554eed0b64aabe5738da86",
    "name": "Copy of Location 1",
    "metadata": [
      {
        "name": "Site ID",
        "value": "abc123"
      }
    ],
    "ownerId": "56d3a61a09c7aa9a82765540",
    "createdTime": "2018-05-01T06:31:51.907Z",
    "workspaceId": "5a554eed0b64aabe5738da86"
  }
]
```



## Acknowledge node alarms

Acknowledge active alarms for a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. Optionally provide a *comment* for the acknowledgement.

**Note:** Only available for Location, Source and Parameter nodes. Required API key permission: *Modify*

## Arguments

Argument	Example	Description
<b>alarmTypes</b>	stateAlarm, optionalAlarm	Optional. Default is ALL. Comma delimited list of specific alarms to acknowledge: [communicationsAlarm, configurationAlarm, controlAlarm, outdatedAlarm, overloadAlarm, processAlarm, qualityAlarm, stateAlarm]

## Request

```
POST /api/v1/nodes/:id/alarms/acknowledge
```

```
{
  "comment": "maintenance team investigating"
}
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Clear node alarms

Clear active and acknowledged alarms for a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'.

---

**Note:** Only available for Location, Source and Parameter nodes. Required API key permission: *Modify*

---

## Arguments

Argument	Example	Description
<b>alarmTypes</b>	stateAlarm,controlAlarm	Optional. Default is <i>ALL</i> . Comma delimited list of specific alarms to clear: [ <i>communicationsAlarm, configurationAlarm, controlAlarm, outdatedAlarm, overloadAlarm, processAlarm, qualityAlarm, stateAlarm</i> ]

## Request

```
POST /api/v1/nodes/:id/alarms/clear
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Trigger an acquisition

Trigger an acquisition (*Acquire Now*) for a Source node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'.

---

**Note:** Only available for Source nodes with Transports that allow for manual collection. Required API key permission: *Modify*

---

## Request

```
POST /api/v1/nodes/:id/operate/acquire
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Retrieve user and group notification preferences

Retrieve a list of all users and groups subscribed to receive notifications for a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'.

## Arguments

Argument	Example	Description
<b>subscribeType</b>	NODE	<i>Optional - Default is NODE.</i> Users and Groups to return in the response: <i>[NODE, BRANCH, ALL]</i> . NODE includes users & groups subscribed to this node. BRANCH includes users & groups subscribed to this node and any nodes it contains. ALL includes all users & groups that have access to the node (and can be subscribed) regardless of current subscription state. This can used to retrieve user notification preferences for a workspace prior to <i>subscribing to notifications</i> (page 471).

## Request

```
GET /api/v1/nodes/:id/notifications
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "users": [
    {
      "name": {
        "last": "Jones",
        "first": "Bob"
      },
      "contact": {
        "email": "bob@company.com"
      },
      "account": {
        "lastLoginTime": "2018-02-27T01:05:02.029Z"
      },
      "notificationCategories": [
        {
          "category": "Maintenance",
          "email": true,
          "sms": false
        },
        {
          "category": "Operations",
          "email": true,
          "sms": true
        },
        {
          "category": "Critical",
          "email": true,
          "sms": false
        }
      ]
    }
  ],
  "groups": [
    {
      "group": "Engineers",
      "users": [
        {
          "name": {
            "last": "Smith",
            "first": "Will"
          },
          "contact": {
            "email": "will@company.com"
          },
          "account": {
            "lastLoginTime": ""
          },
          "notificationCategories": [
            {
```

(continues on next page)

(continued from previous page)

```

        "category": "Maintenance",
        "email": true,
        "sms": true
      },
      {
        "category": "Operations",
        "email": true,
        "sms": true
      },
      {
        "category": "Critical",
        "email": true,
        "sms": true
      }
    ]
  }
]
}

```

## Update user notification preferences

Update user notification preferences for a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'.

This allows for *notificationCategories* to have *email* and *sms* enabled or disabled per *category*. These preferences **apply to all node notification subscriptions within the Workspace**. Only categories with changed attributes need to be included in the update. If the category specified does not exist it will be ignored.

When *subscribing groups to notifications* (page 471) you still need to update notification preferences for individual users of the group.

The response includes updated profiles for all *requested users*.

---

**Note:** The specified users (email addresses) must be valid *workspace users* (page 473). Required API key permission: *Modify*.

You must also *subscribe users and groups to notifications* (page 471) for the specific nodes they want to receive notifications from.

---

## Request

```
POST /api/v1/nodes/:id/notifications
```

```
{
  "users": [
    {
      "user": "bob@company.com",
      "notificationCategories": [
        {
          "category": "Critical",
          "email": true,
          "sms": true
        }
      ]
    },
    {
      "user": "will@company.com",
      "notificationCategories": [
        {
          "category": "Operations",
          "sms": false
        },
        {
          "category": "Critical",
          "email": false,
          "sms": false
        }
      ]
    }
  ]
}
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "users": [
    {
      "user": "bob@company.com",
      "notificationCategories": [
        {
          "category": "Maintenance",
          "email": false,
          "sms": false
        },
        {
          "category": "Operations",
          "email": false,
          "sms": false
        }
      ]
    },
    {

```

(continues on next page)

(continued from previous page)

```

        "category": "Critical",
        "email": true,
        "sms": true
      }
    ],
    },
    {
      "user": "will@company.com",
      "notificationCategories": [
        {
          "category": "Maintenance",
          "email": true,
          "sms": true,
        },
        {
          "category": "Operations",
          "email": true,
          "sms": false
        },
        {
          "category": "Critical",
          "email": false,
          "sms": false
        }
      ]
    }
  ]
}

```

## Subscribe users and groups to notifications

Subscribe a list of users and groups to receive notifications for a node by its **id**. You can use the automatically assigned `_id` or your own `customId` prepended with '@'.

---

**Note:** Only available for Location, Source and Parameter nodes. The specified users (email addresses) must be valid workspace users. Required API key permission: *Modify*

---

## Request

```
POST /api/v1/nodes/:id/notifications/subscribe
```

```

{
  "users": [
    {
      "user": "john@company.com"
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```
    },
    {
      "user": "bob@company.com"
    }
  ],
  "groups": [
    {
      "group": "Engineers"
    }
  ]
}
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Unsubscribe users and groups from notifications

Unsubscribe a list of users and groups from receiving notifications for a node by its **id**. You can use the automatically assigned `_id` or your own *customId* prepended with '@'.

---

**Note:** Only available for Location, Source and Parameter nodes. Required API key permission: *Modify*

---

## Request

```
POST /api/v1/nodes/:id/notifications/unsubscribe
```

```
{
  "users": [
    {
      "user": "john@company.com"
    },
    {
      "user": "bob@company.com"
    }
  ]
}
```

(continues on next page)



(continued from previous page)

```

    ],
    "groups": [
      {
        "group": "Engineers"
      }
    ]
  }

```

## Response

```

HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8

```

```

{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}

```

## Retrieve Workspace user and group permissions

Retrieve a list of all users and groups with access to a workspace (or node) by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. When a user is included in a group that has been provided access, the user entry may contain an *effectivePermissions* attribute which combines individual user and group(s) permissions (where different).

## Request

```
GET /api/v1/nodes/:id/security
```

## Response

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

```

```

{
  "users": [
    {
      "name": {
        "last": "Smith",
        "first": "John"
      }
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    },
    "contact": {
      "email": "john@company.com"
    },
    "account": {
      "lastLoginTime": "2018-05-01T01:21:59.735Z"
    },
    "role": "OWNER",
    "permissions": [
      "VIEW",
      "ATTACHMENT_READ",
      "EXPORT_DATA",
      "SEND_MESSAGE",
      "SUBSCRIBE_NOTIFICATIONS",
      "MANAGE_NOTIFICATIONS",
      "ACKNOWLEDGE_ALARMS",
      "EDIT_ALARMS",
      "CONTROL",
      "CONFIGURE",
      "SECURITY",
      "OWNER"
    ]
  },
  {
    "name": {
      "last": "Jones",
      "first": "Bob"
    },
    "contact": {
      "email": "bob@company.com"
    },
    "account": {
      "lastLoginTime": "2018-02-27T01:05:02.029Z"
    },
    "role": "",
    "permissions": []
    "effectivePermissions": [
      "VIEW"
    ]
  }
],
"groups": [
  {
    "group": "Engineers",
    "role": "View",
    "permissions": [
      "VIEW"
    ],
    "users": [
      {
        "user": "bob@company.com"
      }
    ]
  }
]

```

(continues on next page)

(continued from previous page)

```

    ]
  }
]
}

```

**Note:** When retrieving permissions for a node it will return permissions that have been directly set on this node (or the Workspace) and not inherited node permissions.

### Add or Update Workspace user and group permissions

Add a list of users and groups to a Workspace by its **id** and set or update user security roles. You can use the automatically assigned `_id` or your own *customId* prepended with '@'. You must specify a valid *role* name that has been preconfigured in *account settings* (page 270).

If the user (email address) does not already exist a new user profile will be created and a *profileActivateUrl* will be returned so the user profile can be finalized by navigating to the url in a web browser. When adding a new user (profile does not exist) you can optionally include attributes: *name*, *phone*, *timezone*, *timezoneAdjustForDst*, *timeFormat* and *notificationCategories*.

**Note:** Required API key permission: *Modify*

### Arguments

Argument	Example	Description
<b>notify</b>	TRUE	<i>Optional - Default is FALSE.</i> Notify users they have been granted workspace access

### Request

```
POST /api/v1/nodes/:id/security/subscribe
```

```

{
  "users": [
    {
      "user": "bob@company.com",
      "role": "View"
    },
    {
      "user": "jane@company.com",

```

(continues on next page)

(continued from previous page)

```

        "role": "Operate",
        "expiryTime": "2019-11-05T05:24:32.000+0000"
      },
      "name": {
        "first": "Jane",
        "last": "Smith"
      },
      "phone": "+61400000001",
      "timezone": "Australia/Sydney",
      "timezoneAdjustForDst": true,
      "timeFormat": "YYYY-MM-DD HH:mm:ss"
    },
    "groups": [
      {
        "group": "Engineers",
        "role": "View"
      }
    ]
  }
}

```

## Response

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

```

```

{
  "added": [
    {
      "user": "jane@company.com",
      "role": "Operate",
      "permissions": [
        "VIEW",
        "ATTACHMENT_READ",
        "EXPORT_DATA",
        "SEND_MESSAGE",
        "ACKNOWLEDGE_ALARMS",
        "EDIT_ALARMS",
        "CONTROL",
        "SUBSCRIBE_NOTIFICATIONS"
      ],
      "expiryTime": "2019-11-05T05:24:32.000+0000",
      "profileActivateUrl": "https://app.eagle.io/auth/setupprofile/
↪ca4d1da0-8231-46df-af69-df1b2f1a8b5d"
    },
    {
      "group": "Engineers",
      "role": "View",
      "permissions": [
        "VIEW"
      ]
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    ],
    "users": [
      {
        "user": "will@company.com",
        "profileActivateUrl": "https://app.eagle.io/auth/
↪setupprofile/bb2d3fc9-7399-26dc-ca22-ee2b6f0b6a0c"
      }
    ]
  },
  "updated": [
    {
      "user": "bob@company.com",
      "role": "View",
      "permissions": [
        "VIEW"
      ]
    }
  ]
]
}

```

## Remove users and groups from a Workspace

Remove a list of users and groups from a Workspace by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'.

**Note:** Owner and Administrator users can not be removed from an individual workspace. Required API key permission: *Modify*

## Arguments

Argument	Example	Description
<b>notify</b>	TRUE	<i>Optional - Default is FALSE.</i> Notify users they have had their workspace access revoked

## Request

```
POST /api/v1/nodes/:id/security/unsubscribe
```

```
{
  "users": [
```

(continues on next page)

(continued from previous page)

```
{
  "user": "jane@company.com"
},
"groups": [
  {
    "group": "Engineers"
  }
]
}
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "removed": [
    {
      "user": "jane@company.com"
    },
    {
      "group": "Engineers"
    }
  ]
}
```

## Retrieve node historic data

Retrieve historic data from a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. Data can be returned in JSON (*JTS* (page 572)) or CSV format. Use the *Historic resource* (page 494) for extracting historic data from multiple nodes in a single request.

---

**Note:** Only available for Location and Parameter nodes.

---

## Arguments

Argument	Example	Description
<b>format</b>	JSON	<i>Optional - Default is JSON.</i> Data format to return: <i>[JSON, CSV]</i>
<b>start-Time</b>	2014-08-16T02:00:00Z	<i>Required.</i> <sup>5</sup> <i>ISO8601</i> (page 581) url encoded timestamp
<b>endTime</b>	2014-08-16T02:20:43Z	<i>Required.</i> <sup>5</sup> <i>ISO8601</i> (page 581) url encoded timestamp
<b>limit</b>	100	<i>Optional.</i> Maximum number of historic records to be returned.
<b>time-zone</b>	Etc/UTC	<i>Optional - Default is Etc/UTC.</i> <i>Timezone</i> (page 581) applied to timestamps. Aggregate <i>interval</i> and <i>baseTime</i> calculations will also use this zone.
<b>time-zoneAdjust-ForDst</b>	FALSE	<i>Optional - Default is FALSE.</i> Flag to indicate if timestamps should be adjusted for DST in selected <i>timezone</i> .
<b>time-Query</b>	RECORD	<i>Optional - Default is RECORD.</i> Timestamp to query by. <i>MODIFIED</i> will query by modified timestamp (typically used to obtain records changed since a specific timestamp): <i>[RECORD, MODIFIED]</i>
<b>timeFormat</b>	YYYY-MM-DD HH:mm:ss	<i>Optional.</i> <i>Time format</i> (page 582) to use when <i>format</i> is 'CSV'.
<b>header</b>	TRUE	<i>Optional - Default is TRUE.</i> Flag to include header
<b>delimiter</b>	,	<i>Optional - Default is ','.</i> Delimiter character to use (in CSV format).
<b>textQualifier</b>	"	<i>Optional - Default is '"'</i> . Text qualifier character to use (in CSV format).
<b>quality</b>	NONE	<i>Optional - Default is NONE.</i> Specify how <i>quality</i> (page 577) should be output (in CSV format): <i>[NONE, DELIMITED_WITH_VALUE, SEPARATE_VALUE]</i>
<b>quality-Delimiter</b>	:	<i>Optional - Default is ':'.</i> Delimiter to use when <i>quality</i> is 'DELIMITED_WITH_VALUE'.
<b>qualityExcluded</b>	BAD,UNCERTAIN	<i>Optional - Default uses account settings</i> <i>Exclude quality</i> (page 266). Comma separated list of data point quality types to be excluded: <i>[GOOD, BAD, UNCERTAIN or NONE]</i>
<b>annotation</b>	NONE	<i>Optional - Default is NONE.</i> Specify how <i>annotations</i> (page 578) should be output (in CSV format): <i>[NONE, DELIMITED_WITH_VALUE, SEPARATE_VALUE]</i>
<b>annotationDelimiter</b>	;	<i>Optional - Default is ';'.</i> Delimiter to use when <i>annotation</i> is 'DELIMITED_WITH_VALUE'.
<b>render-Type</b>	VALUE	<i>Optional - Default is node displayType.</i> Rendering of value: <i>[VALUE, STATE]</i>
<b>render-Format</b>	0.000	<i>Optional - Default is node format.</i> <i>Format</i> (page 147) to apply when <i>renderType</i> is <i>VALUE</i> . '#' must be <i>URL Encoded</i> ( <a href="http://en.wikipedia.org/wiki/Percent-encoding">http://en.wikipedia.org/wiki/Percent-encoding</a> ) as '%23'.
<b>aggregate</b>	AVERAGE	<i>Optional - Default is NONE (raw).</i> Historic <i>aggregate</i> (page 569) to apply to extracted data.
<b>base-Time</b>	D	<i>Optional.</i> <i>OPC Base Time</i> (page 585) required for aggregation.
<b>interval</b>	3H	<i>Optional.</i> <i>OPC Interval</i> (page 585) required for aggregation.

## Request

```
GET /api/v1/nodes/:id/historic
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "docType": "jts",
  "version": "1.0",
  "header": {
    "startTime": "2014-08-16T02:00:00.000Z",
    "endTime": "2014-08-16T02:20:43.000Z",
    "recordCount": 5,
    "columns": {
      "0": {
        "id": "541a5a129bc9b4035f906d70",
        "name": "Temperature",
        "dataType": "NUMBER",
        "aggregate": "NONE"
      }
    }
  },
  "data": [
    {
      "ts": "2014-08-16T02:00:39.000Z",
      "f": { "0": { "v": 28.21 } }
    },
    {
      "ts": "2014-08-16T02:05:40.000Z",
      "f": { "0": { "v": 28.22 } }
    },
    {
      "ts": "2014-08-16T02:10:41.000Z",
      "f": { "0": { "v": 28.7 } }
    },
    {
      "ts": "2014-08-16T02:15:42.000Z",
      "f": { "0": { "v": 29.2 } }
    },
    {
      "ts": "2014-08-16T02:20:43.000Z",
      "f": { "0": { "v": 29.18 } }
    }
  ]
}
```

---

<sup>5</sup> startTime or endTime can be omitted when limit is specified.



## Retrieve node historic chart

Retrieve historic chart image from a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. Chart images can be returned in PNG, JPG, SVG and PDF format.

If a *chart node* (page 66) is specified, you can optionally specify: *format*, *width*, *height*, *title*, *subtitle*, *startTime*, *endTime*, *timezone*, *timezoneAdjustForDst*.

If a *parameter node* (page 147) is specified, the nodes' pre-configured (or default) chart settings will be used but you can optionally specify any of the arguments below. Use the *Historic resource* (page 494) chart endpoint for generating an adhoc chart using data from multiple parameters in a single request.

---

**Note:** Only available for Parameter and Custom Chart nodes.

---

## Arguments

Argument	Example	Description
<b>format</b>	PNG	<i>Optional - Default is PNG.</i> Image format to return: <i>[PNG, JPG, SVG, PDF]</i>
<b>width</b>	1200	<i>Optional - Default is 1200.</i> Width of image (png, jpg) in pixels
<b>height</b>	800	<i>Optional - Default is 800.</i> Height of image (png, jpg) in pixels
<b>scale</b>	2	<i>Optional - Default is 1.</i> Multiplier to adjust image dimensions (png, jpg) for higher resolution output: <i>1-4</i> .
<b>title</b>	My Chart	<i>Optional.</i> Title to display on chart
<b>subtitle</b>	Generated by eagle.io	<i>Optional.</i> Subtitle to display on chart
<b>start-Time</b>	2014-08-16T02:00:00Z	<i>Required.</i> <a href="#">ISO8601</a> (page 581) url encoded timestamp
<b>endTime</b>	2014-08-16T02:20:43Z	<i>Required.</i> <a href="#">ISO8601</a> (page 581) url encoded timestamp
<b>timezone</b>	Etc/UTC	<i>Optional - Default is Etc/UTC.</i> <a href="#">Timezone</a> (page 581) applied to timestamps. Aggregate <i>interval</i> and <i>baseTime</i> calculations will also use this zone
<b>time-zoneAdjust-ForDst</b>	FALSE	<i>Optional - Default is FALSE.</i> Flag to indicate if timestamps should be adjusted for DST in selected <i>timezone</i>
<b>render-Type<sup>6</sup></b>	VALUE	<i>Optional - Default is node displayType.</i> Rendering of value: <i>[VALUE, STATE]</i>
<b>aggregate<sup>6</sup></b>	AVERAGE	<i>Optional - Default is NONE (raw).</i> Historic <a href="#">aggregate</a> (page 569) to apply to extracted data
<b>base-Time<sup>6</sup></b>	D	<i>Optional.</i> <a href="#">OPC Base Time</a> (page 585) required for aggregation
<b>interval<sup>6</sup></b>	3H	<i>Optional.</i> <a href="#">OPC Interval</a> (page 585) required for aggregation
<b>interval-Include<sup>6</sup></b>	PARTIAL	<i>Optional - Default is PARTIAL.</i> COMPLETE will include aggregated values for complete intervals only. PARTIAL will also include values for non-complete intervals: <i>[PARTIAL, COMPLETE]</i>
<b>baseline-Type<sup>6</sup></b>	RELATIVE	<i>Optional - Default is ABSOLUTE.</i> Absolute will return data point values unmodified. Relative will subtract the first data point value from all subsequent data point values: <i>[ABSOLUTE, RELATIVE]</i>

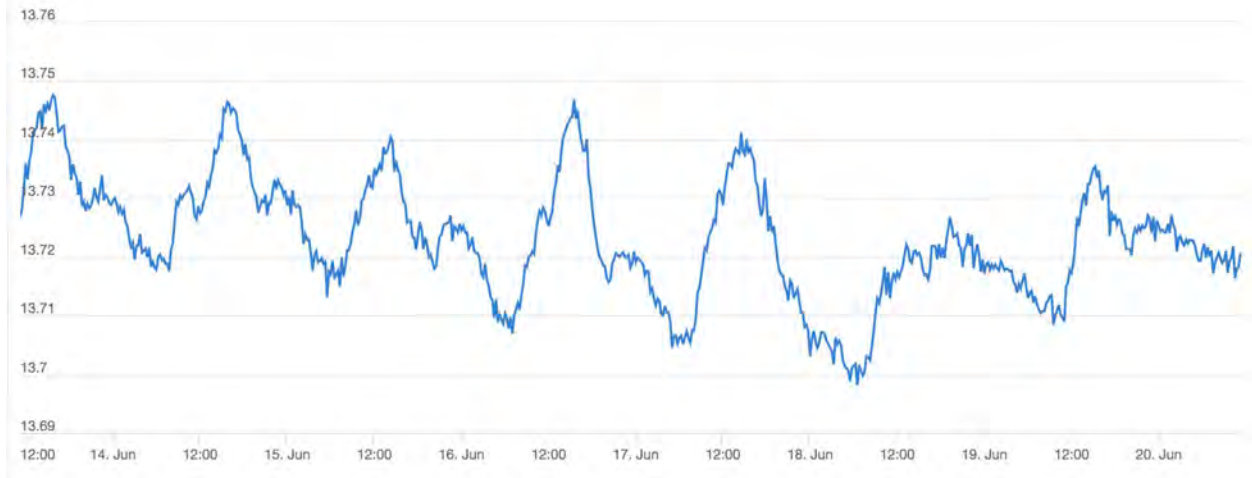
## Request

```
GET /api/v1/nodes/:id/historic/chart
```

<sup>6</sup> Only available when specifying a *parameter* node.

## Response

```
HTTP/1.1 200 OK
Content-Type: image/png;
```



## Retrieve node report

Generate and retrieve a PDF file from a [report node](#) (page 161) by its **id**. You can use the automatically assigned `_id` or your own `customId` prepended with '@'.

**Note:** The reporting feature is currently a **pre-release** version that is subject to change without notice and is only available to limited accounts.

## Arguments

Argument	Example	Description
<b>nowTime</b>	2021-06-01T00:00:00Z	<i>Optional.</i> Default is null (current time). <a href="#">ISO8601</a> (page 581) url encoded timestamp to use for report generation. All current values and historic records are relative to this timestamp.
<b>timezone</b>	Etc/UTC	<i>Optional - Default is report timezone.</i> <a href="#">Timezone</a> (page 581) applied to timestamps. Only specify to override the default report timezone.
<b>time-zoneAdjust-ForDst</b>	FALSE	<i>Optional - Default is report timezone adjust for dst.</i> Flag to indicate if timestamps should be adjusted for DST in selected <i>timezone</i> Only specify to override the default report timezone.

## Request

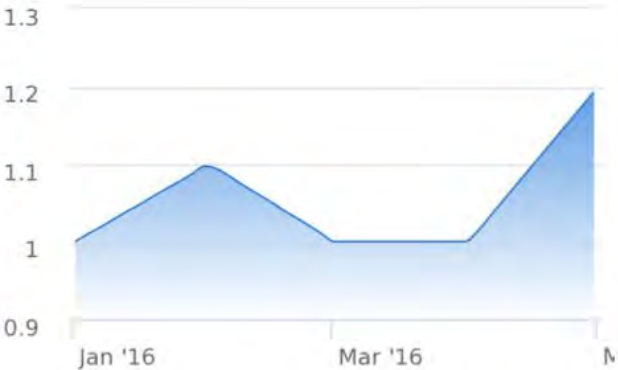
```
GET /api/v1/nodes/:id/report
```

## Response

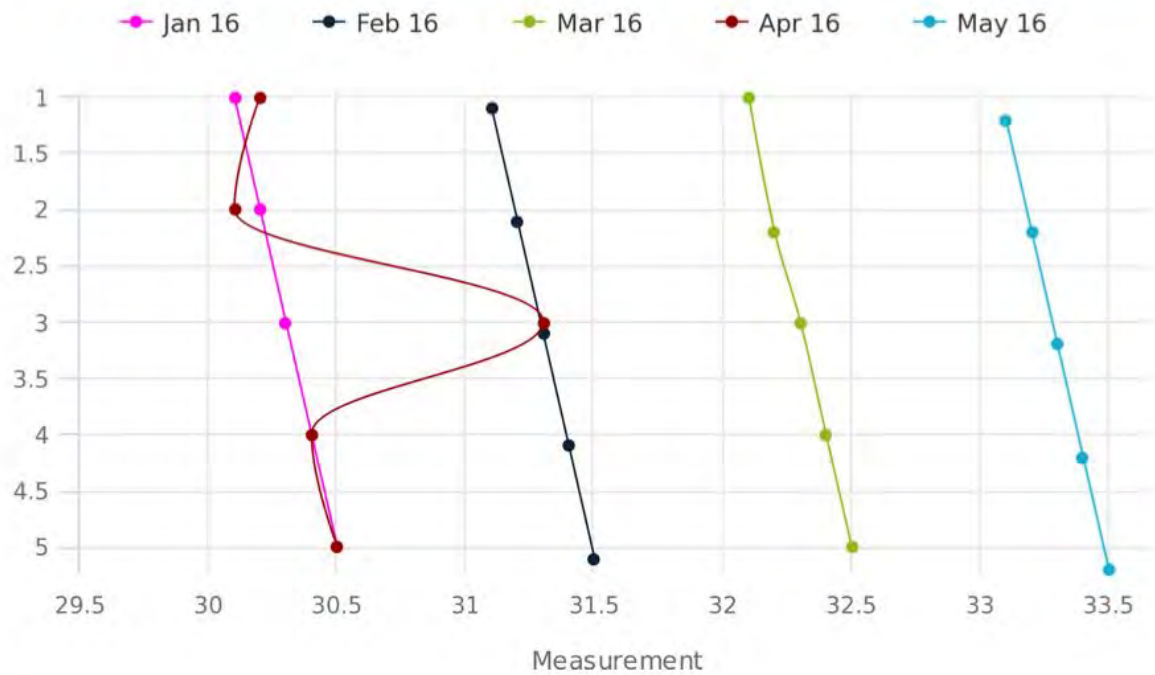
```
HTTP/1.1 200 OK
Content-Type: application/pdf;
```

### Example report

Report for 2021-06-11 11:09:18



Displacement chart showing variation over time



## Update Parameter or Location historic data

Update historic data for a Parameter or Location node by its **id**. You can use the automatically assigned `_id` or your own *customId* prepended with '@'. Data can be inserted in JSON (*JTS* (page 572)) or CSV format. Use the *Historic resource* (page 494) to update historic data for multiple nodes in a single request.

**Note:** Only available for Location and Parameter nodes. Required API key permission: *Modify*

## Arguments

Argument	Example	Description
<b>format</b>	JSON	<i>Optional - Default is JSON.</i> Data format being inserted: <i>[JSON]</i> .
<b>write-Mode</b>	MERGE_OVERWRITE_EXISTING	<i>Optional - Default is MERGE_OVERWRITE_EXISTING.</i> See all available <i>write mode</i> (page 217) options.
<b>notifyOn</b>	LATEST_ONLY	<i>Optional - Default is LATEST_ONLY.</i> When to generate events, raise alarms and send notifications: <i>[ALL_NEWER, LATEST_ONLY, NONE]</i> . ALL_NEWER: All events newer than parameter current value. LATEST_ONLY: Latest event newer than parameter current value.
<b>columnIndex</b>	0	<i>Optional - Default is 0.</i> Index of column in data to be associated with this parameter. Will use index specified in JTS Doc header if available or default to 0.

## Request

```
PUT /api/v1/nodes/:id/historic
```

```
{
  "docType": "jts",
  "version": "1.0",
  "data": [
    {
      "ts": "2014-09-17T07:30:00Z",
      "f": { "0": { "v": 25.05 } }
    },
    {
      "ts": "2014-09-17T07:40:00Z",
      "f": { "0": { "v": 25.20 } }
    },
    {
      "ts": "2014-09-17T07:50:00Z",
      "f": { "0": { "v": 25.14 } }
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
]
}
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Update Data Source historic data

Update historic data for multiple parameters via a Data Source node **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. Data can be inserted in JSON (*JTS* (page 572)) format. New parameters will be automatically created.

The JTS Document must contain header columns. Each column must either specify a **series** or parameter **id**. If **id** is provided it will be used to match to the associated parameter under the current Data Source, otherwise the associated parameter will be matched using the **series** attribute.

If a column specified in the header can not be matched to an existing parameter (and **series** was provided), a new parameter will be created automatically. Optionally specify **name**, **dataType** (NUMBER, TEXT, TIME. Default is NUMBER) and **units** in the column header which will be used when creating new parameters.

---

**Note:** Only available for Datasource nodes. Required API key permission: *Modify*

---

## Arguments

Argument	Example	Description
<b>format</b>	JSON	<i>Optional - Default is JSON.</i> Data format being inserted: <i>[JSON]</i> .
<b>write-Mode</b>	MERGE_OVERWRITE_EXISTING	<i>Optional - Default is MERGE_OVERWRITE_EXISTING.</i> See all available <a href="#">write mode</a> (page 217) options.
<b>notifyOn</b>	LATEST_ONLY	<i>Optional - Default is LATEST_ONLY.</i> When to generate events, raise alarms and send notifications: <i>[ALL_NEWER, LATEST_ONLY, NONE]</i> . ALL_NEWER: All events newer than parameter current value. LATEST_ONLY: Latest event newer than parameter current value.

## Request

```
PUT /api/v1/nodes/:id/historic
```

```
{
  "docType": "jts",
  "version": "1.0",
  "header" : {
    "columns": {
      "0": {
        "series": "temp1",
        "name": "Temperature",
        "dataType": "NUMBER",
        "units": "°C"
      },
      "1": {
        "id": "541a5a129bc9b4035f906d71"
      }
    }
  },
  "data": [
    {
      "ts": "2014-09-17T07:30:00Z",
      "f": { "0": { "v": 25.05 } }
    },
    {
      "ts": "2014-09-17T07:40:00Z",
      "f": { "0": { "v": 25.20 } }
    },
    {
      "ts": "2014-09-17T07:50:00Z",
      "f": { "0": { "v": 25.14 }, "1": { "v": "text data here" } }
    }
  ]
}
```

(continues on next page)



(continued from previous page)

```
]
}
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Update node historic data with a single value

Update historic data for a node with a single record. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. You must specify at least one record attribute: *value*, *quality*, *annotation*. If *timestamp* is omitted the time the request was made will be used. Existing records with the same timestamp will be overwritten.

---

**Note:** Only available for Location and Parameter nodes. Required API key permission: *Modify*

---

## Request

```
PUT /api/v1/nodes/:id/historic/now
```

```
{
  "value": 10,
  "quality": 156,
  "annotation": "maintenance performed at site",
  "timestamp": "2019-01-09T23:38:00Z"
}
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Delete node historic data

Delete all historic data from a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'.

---

**Note:** Only available for Location and Parameter nodes. Required API key permission: *Modify*

---

## Request

```
DELETE /api/v1/nodes/:id/historic
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Retrieve node events

Retrieve events related to a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. Use the [Events resource](#) (page 505) for retrieving events related to multiple nodes in a single request.

## Arguments

Argument	Example	Description
<b>start-Time</b>	2017-08-16T02:00:00Z	<i>Optional.</i> <a href="#">ISO8601</a> (page 581) url encoded timestamp
<b>end-Time</b>	2017-08-16T02:20:43Z	<i>Optional.</i> <a href="#">ISO8601</a> (page 581) url encoded timestamp
<b>attr</b>	event-Time,message	<i>Optional.</i> Comma delimited list of attributes to include in response
<b>filter</b>	level(\$eq:DEBUG)	<i>Optional.</i> <a href="#">Filter</a> (page 437) the records based on attribute value(s). <i>Note: event-Time can not be used as a filter</i>
<b>limit</b>	100	<i>Optional - Default is 1000 if both startTime &amp; endTime are not specified.</i> Maximum number of records to be returned.
<b>skip</b>	50	<i>Optional.</i> Skip the first <i>n</i> records returned. Can be used with <code>limit</code> to paginate results
<b>sort</b>	event-Time(DESC)	<i>Optional - Default is eventTime(ASC).</i> Comma delimited list of attributes to sort by. Optionally include sort direction in parentheses or default to ASC: <i>[ASC, DESC]</i>

## Request

```
GET /api/v1/nodes/:id/events
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
[
  {
    "_class": "io.eagle.models.event.Event",
    "_id": "58d3508c8d6a80f2bd7eaf2e",
    "alarmState": "NONE",
    "message": "Processed 60 records from sensors.dat [4 parameters_
↪updated]",
    "eventType": "ACQUISITION",
    "eventTime": "2017-03-23T04:35:24.000Z",
    "nodeId": "57d799d86a559fb79215b5c1",
    "nodeType": "io.eagle.models.node.source.data.TextParser",
    "level": "DEBUG",
    "workspaceId": "57e0a4d8a19b65042ad079c2",
    "username": "System"
```

(continues on next page)

(continued from previous page)

```

    },
    {
      "_class": "io.eagle.models.event.Event",
      "_id": "58d0b9df0bb36d2965e5b745",
      "alarmState": "NONE",
      "message": "Rename node 'Cond' to 'Conductivity'",
      "eventType": "CONFIGURATION",
      "eventTime": "2017-03-21T05:27:59.000Z",
      "nodeId": "58bf63d57426a2b19ea63f54",
      "nodeType": "io.eagle.models.node.point.NumberPoint",
      "level": "INFO",
      "workspaceId": "577b5a821f704bb5a3f1411b",
      "username": "user@company.com"
    }
  ]

```

## Create node events

Create new event(s) for a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. Multiple events can be created for the node in a single request by providing an Array of Events in the request body. The event requires a **message** and optional **eventTime**. Events created via the API are assigned *level: INFO, eventType: CONFIGURATION, alarmState: NONE*.

**Note:** Required API key permission: *Modify*

## Attributes

At-tribute	Type	Description
<b>mes- sage</b>	String	<i>Required.</i> Event description. 1-500 characters.
<b>event- Time</b>	Time	<i>Optional.</i> <i>ISO8601</i> (page 581) timestamp associated with the event. If not specified the time the request was made will be used.

## Request

```
POST /api/v1/nodes/:id/events
```

```

[
  {
    "message": "Configuration and Site Acceptance Test complete"
  },

```

(continues on next page)

(continued from previous page)

```
{
  "message": "New event created via API with optional timestamp",
  "eventTime": "2019-10-20T02:32:55.000Z"
}
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

## Delete node events

Delete all events related to a node by its **id**. You can use the automatically assigned *\_id* or your own *customId* prepended with '@'. A new event “*Cleared events*” will be created to indicate this action occurred.

---

**Note:** Required API key permission: *Modify*

---

## Request

```
DELETE /api/v1/nodes/:id/events
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

### 3.3.2 Historic

Historic data can be extracted or inserted into one or more nodes in a single request. You can also use the [Nodes resource](#) (page 442) to extract or insert data for a single node. Data can be extracted or inserted in JSON ([JTS](#) (page 572)) format.

We recommend you `limit` the size of requests to 1000 records or less and use [HTTP compression](#) (page 435) whenever possible. If the request or response is too large or exceeds our processing time threshold, the **413 Request Too Large** error response will be returned.

Successful extraction requests will result in a **200 OK** response with the data returned in the body of the response. Successful insertion requests always result in a **202 Accepted** response.

#### Quick Links

- [Retrieve historic data from multiple nodes](#) (page 494)
- [Retrieve historic chart with multiple nodes](#) (page 497)
- [Update historic data for multiple nodes](#) (page 499)

### Retrieve historic data from multiple nodes

Retrieve historic data from multiple nodes in a single request using **node id's**. You can use the automatically assigned `_id's` or your own `customId's` prepended with '@'.

---

**Note:** Only available for Location and Parameter nodes.

---

## Arguments

Argument	Example	Description
<b>format</b>	JSON	<i>Optional - Default is JSON.</i> Data format to return: <i>[JSON, CSV]</i>
<b>startTime</b>	2014-08-16T02:00:00Z	<i>Required.</i> <sup>1</sup> <i>ISO8601</i> (page 581) url encoded timestamp
<b>endTime</b>	2014-08-16T02:20:43Z	<i>Required.</i> <sup>1</sup> <i>ISO8601</i> (page 581) url encoded timestamp
<b>limit</b>	100	<i>Optional.</i> Maximum number of historic records to be returned
<b>timezone</b>	Etc/UTC	<i>Optional - Default is Etc/UTC.</i> <i>Timezone</i> (page 581) applied to timestamps. Aggregate <i>interval</i> and <i>baseTime</i> calculations will also use this zone.
<b>timezoneAdjustForDst</b>	FALSE	<i>Optional - Default is FALSE.</i> Flag to indicate if timestamps should be adjusted for DST in selected <i>timezone</i> .
<b>timeQuery</b>	RECORD	<i>Optional - Default is RECORD.</i> Timestamp to query by. MODIFIED will query by modified timestamp (typically used to obtain records changed since a specific timestamp): <i>[RECORD, MODIFIED]</i>
<b>timeFormat</b>	YYYY-MM-DD HH:mm:ss	<i>Optional.</i> <i>Time format</i> (page 582) to use when <i>format</i> is 'CSV'.
<b>header</b>	TRUE	<i>Optional - Default is TRUE.</i> Flag to include header
<b>delimiter</b>	,	<i>Optional - Default is ','.</i> Delimiter character to use (in CSV format).
<b>textQualifier</b>	"	<i>Optional - Default is ' '.</i> Text qualifier character to use (in CSV format).
<b>quality</b>	NONE	<i>Optional - Default is NONE.</i> Specify how <i>quality</i> (page 577) should be output (in CSV format): <i>[NONE, DELIMITED_WITH_VALUE, SEPARATE_VALUE]</i>
<b>qualityDelimiter</b>	:	<i>Optional - Default is ':'.</i> Delimiter to use when <i>quality</i> is 'DELIMITED_WITH_VALUE'.
<b>qualityExcluded</b>	BAD,UNCERTAIN	<i>Optional - Default uses account settings</i> <i>Exclude quality</i> (page 266). Comma separated list of data point quality types to be excluded: <i>[GOOD, BAD, UN-</i>

Include individual parameter arguments in parentheses with key/value separated by colons and multiple arguments semicolon delimited:

```
params=:id(key:value;key:value),:id
```

Example with optional parameter arguments:

```
params=541a5a129bc9b4035f906d71(aggregate:AVERAGE;baseTime:D;interval:3H),  
↪@sensor_254
```

## Request

```
GET /api/v1/historic
```

## Response

```
HTTP/1.1 200 OK  
Content-Type: application/json; charset=utf-8
```

```
{  
  "docType": "jts",  
  "version": "1.0",  
  "header": {  
    "startTime": "2014-08-16T02:00:00.000Z",  
    "endTime": "2014-08-16T02:20:43.000Z",  
    "recordCount": 5,  
    "columns": {  
      "0": {  
        "id": "541a5a129bc9b4035f906d70",  
        "name": "Temperature",  
        "dataType": "NUMBER",  
        "aggregate": "NONE"  
      }  
    }  
  },  
  "data": [  
    {  
      "ts": "2014-08-16T02:00:39.000Z",  
      "f": { "0": { "v": 28.21 } }  
    },  
    {  
      "ts": "2014-08-16T02:05:40.000Z",  
      "f": { "0": { "v": 28.22 } }  
    },  
    {  
      "ts": "2014-08-16T02:10:41.000Z",
```

(continues on next page)

---

<sup>1</sup> startTime or endTime can be omitted when limit is specified.



(continued from previous page)

```
    "f": { "0": { "v": 28.7 } }  
  },  
  {  
    "ts": "2014-08-16T02:15:42.000Z",  
    "f": { "0": { "v": 29.2 } }  
  },  
  {  
    "ts": "2014-08-16T02:20:43.000Z",  
    "f": { "0": { "v": 29.18 } }  
  }  
]  
}
```

### Retrieve historic chart with multiple nodes

Retrieve historic chart image with multiple nodes in a single request using **node id's**. You can use the automatically assigned *\_id's* or your own *customId's* prepended with '@'. Chart images can be returned in PNG, JPG, SVG and PDF format.

The node's pre-configured (or default) chart settings will be used but you can optionally specify any of the arguments below.

Alternatively use the [Nodes resource](#) (page 442) historic chart endpoint for generating an image from a [chart node](#) (page 66).

---

**Note:** Only available for Parameter nodes.

---

## Arguments

Argument	Example	Description
<b>format</b>	PNG	<i>Optional</i> - Default is PNG. Image format to return: [PNG, JPG, SVG, PDF]
<b>width</b>	1200	<i>Optional</i> - Default is 1200. Width of image (png, jpg) in pixels
<b>height</b>	800	<i>Optional</i> - Default is 800. Height of image (png, jpg) in pixels
<b>scale</b>	2	<i>Optional</i> - Default is 1. Multiplier to adjust image dimensions (png, jpg) for higher resolution output: 1-4.
<b>title</b>	My Chart	<i>Optional</i> . Title to display on chart
<b>subtitle</b>	Generated by eagle.io	<i>Optional</i> . Subtitle to display on chart
<b>startTime</b>	2014-08-16T02:00:00Z	<i>Optional</i> . <a href="#">ISO8601</a> (page 581) url encoded timestamp
<b>endTime</b>	2014-08-16T02:20:43Z	<i>Optional</i> . <a href="#">ISO8601</a> (page 581) url encoded timestamp
<b>timezone</b>	Etc/UTC	<i>Optional</i> - Default is Etc/UTC. <a href="#">Timezone</a> (page 581) applied to timestamps. Aggregate <i>interval</i> and <i>baseTime</i> calculations will also use this zone
<b>timezoneAdjustForDst</b>	FALSE	<i>Optional</i> - Default is FALSE. Flag to indicate if timestamps should be adjusted for DST in selected <i>timezone</i>
<b>params</b>	541a5a129bc9b4035f906d70@sensor_254	<i>Required</i> . Comma delimited list of parameter id's to include in extraction. Individual parameter options are specified in parentheses
<b>renderType</b>	VALUE	<i>Optional</i> - Default is <i>parameter displayType</i> . Rendering of value: [VALUE, STATE]
<b>aggregate</b>	AVERAGE	<i>Optional</i> . Historic <a href="#">aggregate</a> (page 569) to apply to extracted data
<b>baseTime</b>	D	<i>Optional</i> . <a href="#">OPC Base Time</a> (page 585) required for aggregation
<b>498 interval</b>	3H	<i>Optional</i> . <a href="#">Chapter 3. HTTP API</a> (page 585) required for aggregation
	RELATIVE	<i>Optional</i> . Relative time will use

Include individual parameter arguments in parentheses with key/value separated by colons and multiple arguments semicolon delimited:

```
params=:id(key:value;key:value),:id
```

Example with optional parameter arguments:

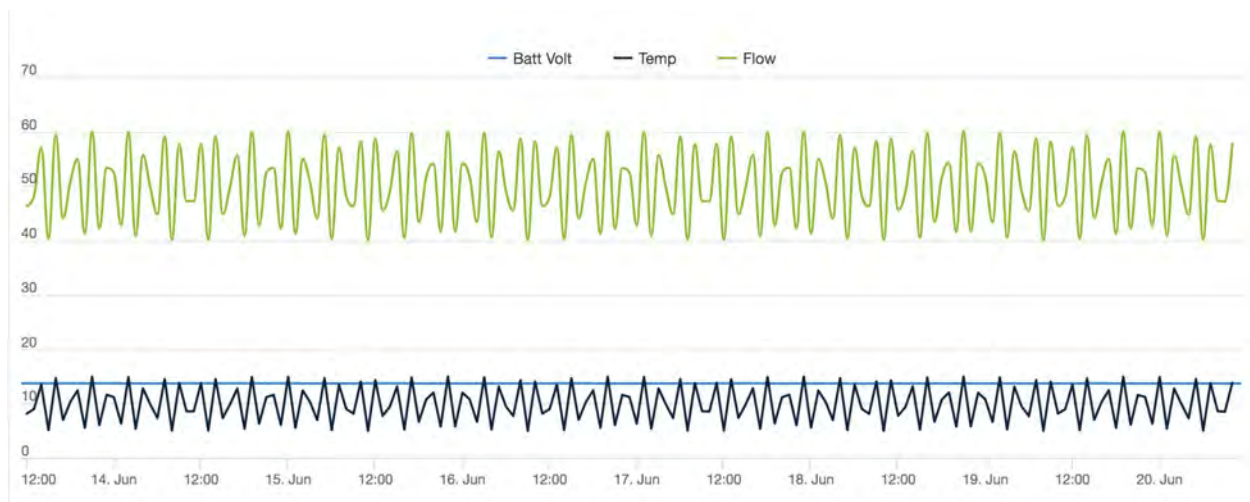
```
params=541a5a129bc9b4035f906d71(aggregate:AVERAGE;baseTime:D;interval:3H),  
↪@sensor_254,@sensor_255
```

## Request

```
GET /api/v1/historic/chart
```

## Response

```
HTTP/1.1 200 OK  
Content-Type: application/json; charset=utf-8
```



## Update historic data for multiple nodes

Update historic data for one or more nodes by **id**'s. You can use the automatically assigned *\_id*'s or your own *customId*'s prepended with '@'.

**Note:** Only available for Location and Parameter nodes. Required API key permission: *Modify*

## Arguments

Argument	Example	Description
<b>format</b>	JSON	<i>Optional</i> - Default is <i>JSON</i> . Data format being inserted: <i>[JSON]</i> . (CSV support coming soon)
<b>writeMode</b>	MERGE_OVERWRITE_EXISTING	<i>Optional</i> - Default is <i>MERGE_OVERWRITE_EXISTING</i> . See all available <i>write mode</i> (page 217) options.
<b>notifyOn</b>	LATEST_ONLY	<i>Optional</i> - Default is <i>LATEST_ONLY</i> . When to generate events, raise alarms and send notifications: <i>[ALL_NEWER, LATEST_ONLY, NONE]</i> . <i>ALL_NEWER</i> : All events newer than parameter current value. <i>LATEST_ONLY</i> : Latest event newer than parameter current value.
<b>params</b>	541a5a129bc9b4035f906d70@sensor_254	<i>Required.</i> <sup>2</sup> Comma delimited list of parameter <i>_id</i> 's to include in extraction. Individual parameter options are specified in parentheses.
<b>columnIndex</b>	0	<i>Required.</i> <sup>2</sup> Index of column in data to be associated with this parameter.

Include individual parameter arguments in parentheses with key/value separated by colons and multiple arguments semicolon delimited:

```
params=:id(key:value;key:value),:id
```

Example with required parameter arguments:

```
params=541a5a129bc9b4035f906d70(columnIndex:0),@sensor_254(columnIndex:1)
```

<sup>2</sup> params argument can be omitted if JTS Document contains headers

## Request

```
PUT /api/v1/historic
```

```
{
  "docType": "jts",
  "version": "1.0",
  "data": [
    {
      "ts": "2014-09-17T07:30:00Z",
      "f": { "0": { "v": 25.05 }, "1": { "v": 44.6 } }
    },
    {
      "ts": "2014-09-17T07:40:00Z",
      "f": { "0": { "v": 25.20 } }
    },
    {
      "ts": "2014-09-17T07:50:00Z",
      "f": { "0": { "v": 25.14 }, "1": { "v": 45.21 } }
    }
  ]
}
```

## Response

```
HTTP/1.1 202 Accepted
Content-Type: application/json; charset=utf-8
```

```
{
  "status": {
    "code": 202,
    "message": "Operation accepted but not yet complete"
  }
}
```

### 3.3.3 Devices

Devices contain details about modems and other communication hardware associated with a Source or Workspace.

#### Quick Links

- [Device Attributes](#) (page 502)
- [Retrieve all devices](#) (page 502)
- [Retrieve a device](#) (page 504)

## Device Attributes

Consumers of Devices should tolerate the addition of new attributes and variance in ordering of attributes with ease. Not all attributes appear in all contexts. It is generally safe to consider a nulled attribute, an empty set, and the absence of an attribute as the same thing.

Attribute	Type	Description
<b>_class</b>	String	Identifies resource type: <i>io.eagle.models.device.*</i>
<b>_id</b>	ObjectId	Unique identifier for this device
<b>config</b>	Object	Object containing device specific configuration
<b>createdTime</b>	Time	<i>ISO8601</i> (page 581) timestamp the device was created
<b>deviceKey</b>	String	IMEI or Unique identifier for this device
<b>isConnected</b>	Boolean	Flag to indicate if device is currently connected
<b>lastConnected</b>	Time	<i>ISO8601</i> (page 581) timestamp the device last established a connection
<b>phoneNumber</b>	String	Phone number associated with the device
<b>software</b>	Object	Object containing info about the current device software
<b>workspaceId</b>	ObjectId	Associated Workspace _id

## Retrieve all devices

Retrieve a list of all devices available to the authenticated API key. Optionally *filter* by attribute values.

## Arguments

Argument	Example	Description
<b>attr</b>	_id,phoneNumber	<i>Optional.</i> Comma delimited list of attributes to include in response
<b>filter</b>	isConnected(\$eq:true)	<i>Optional.</i> <i>Filter</i> (page 437) the records based on attribute value(s)
<b>limit</b>	100	<i>Optional.</i> Maximum number of records to be returned
<b>skip</b>	50	<i>Optional.</i> Skip the first <i>n</i> records returned. Can be used with <code>limit</code> to paginate results
<b>sort</b>	created-Time(DESC)	<i>Optional.</i> Comma delimited list of attributes to sort by. Optionally include sort direction in parentheses or default to ASC: <i>[ASC, DESC]</i>

## Request

```
GET /api/v1/devices
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
[
  {
    "_class": "io.eagle.models.device.Device",
    "_id": "53ce0b6a0eba8dc46900000f",
    "createdTime": "2014-09-08T04:13:32.488Z",
    "deviceKey": "123456789012347",
    "isConnected": true,
    "lastConnected": null,
    "workspaceId": "536884ecb5a76fd5d3000014"
  },
  {
    "_class": "io.eagle.models.device.ScoutDevice",
    "_id": "541b8d46b94f051649012fae",
    "config": {
      "utcOffset": "+00:00",
      "simPin": "",
      "schedule": {
        "scheduleType": "SERVER",
        "statusIntervalSeconds": 600,
        "dataIntervalSeconds": 3600,
        "dataIntervalOffsetSeconds": -1
      },
      "logLevel": "INFO",
      "interfaces": {
        "ETH0": {
          "gateway": "",
          "subnet": "",
          "ipAddress": "",
          "_class": "io.eagle.models.device.config.EthernetInterface"
        },
        "COM1": {
          "stopBits": 1,
          "speed": 9600,
          "rts": false,
          "parity": "none",
          "name": "UART1",
          "dataBits": 8,
          "cts": false,
          "_class": "io.eagle.models.device.config.SerialInterface"
        }
      },
      "gpsEnabled": true,
      "connection": {
        "retryMax": 3,
        "retryDelaySeconds": 30,
        "port": 4331,

```

(continues on next page)

(continued from previous page)

```
        "host": "device.eagle.io"
      },
      "apn": {
        "user": "",
        "pass": "",
        "host": ""
      }
    },
    "createdTime": "2014-09-08T05:24:15.461Z",
    "deviceKey": "123456789012313",
    "isConnected": false,
    "lastConnected": null,
    "phoneNumber": null,
    "software": {
      "appSentTime": null,
      "appVersion": null,
      "firmwareSentTime": null,
      "firmwareVersion": null,
      "bootloaderSentTime": null,
      "bootloaderVersion": null,
      "branch": "RELEASE"
    },
    "workspaceId": "536884ecb5a76fd5d3000014"
  }
]
```

## Retrieve a device

Retrieve a device by its **id**.

## Arguments

Argument	Example	Description
<b>attr</b>	<code>_id,_class</code>	<i>Optional.</i> Comma delimited list of attributes to include in response

## Request

```
GET /api/v1/devices/:id
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```



```
{
  "_class": "io.eagle.models.device.Device",
  "_id": "53ce0b6a0eba8dc46900000f",
  "datasourceId": "541a45c1d07ac441473b0fc3",
  "deviceKey": "123456789012347",
  "isConnected": true,
  "lastConnected": null,
  "phoneNumber": "+61400000001",
  "software": {
    "appSentTime": null,
    "appVersion": "0.4.1.20140714155804",
    "bootloaderSentTime": null,
    "bootloaderVersion": "SWI6200T_02.02.04.18BT",
    "branch": "RELEASE",
    "firmwareSentTime": null,
    "firmwareVersion": "FW_753_18.FX8Fx"
  },
  "workspaceId": "536884ecb5a76fd5d3000014"
}
```

### 3.3.4 Events

Events contain details about user interactions, alarm history and system generated notices.

#### Quick Links

- [Event Attributes](#) (page 505)
- [Retrieve events](#) (page 506)
- [Create or Delete events](#) (page 508)

### Event Attributes

Consumers of Events should tolerate the addition of new attributes and variance in ordering of attributes with ease. Not all attributes appear in all contexts. It is generally safe to consider a nulled attribute, an empty set, and the absence of an attribute as the same thing.

Attribute	Type	Description
<b>_class</b>	String	Identifies resource type: <i>io.eagle.models.event.*</i>
<b>_id</b>	ObjectId	Unique identifier for this event
<b>level</b>	String	Detail level of event: <i>[DEBUG, INFO, WARNING, ERROR]</i>
<b>event-Time</b>	Time	<i>ISO8601</i> (page 581) timestamp associated with the event
<b>event-Type</b>	String	Type of event: <i>[ACQUISITION, CONFIGURATION, STATE, NOTIFICATION, CONTROL, SECURITY]</i>
<b>message</b>	String	Event description
<b>alarm-State</b>	String	Indicate if event is associated with an alarm: <i>[NONE, ACTIVE, ACKNOWLEDGED, CLEARED]</i>
<b>nodeId</b>	ObjectId	Id of node associated with the event (where applicable)
<b>nodeType</b>	String	_class of node associated with the event (where applicable)
<b>workspaceId</b>	ObjectId	Associated Workspace _id
<b>ownerId</b>	ObjectId	Associated Owner (account) _id
<b>username</b>	String	User or process that triggered the event

## Retrieve events

Retrieve a list of events available to the authenticated API key. Optionally *filter* by attribute values.

## Arguments

Argument	Example	Description
<b>start-Time</b>	2017-08-16T02:00:00Z	<i>Optional.</i> <a href="#">ISO8601</a> (page 581) url encoded timestamp
<b>end-Time</b>	2017-08-16T02:20:43Z	<i>Optional.</i> <a href="#">ISO8601</a> (page 581) url encoded timestamp
<b>attr</b>	event-Time,message	<i>Optional.</i> Comma delimited list of attributes to include in response
<b>filter</b>	level(\$eq:DEBUG)	<i>Optional.</i> <a href="#">Filter</a> (page 437) the records based on attribute value(s). <i>Note: event-Time can not be used as a filter</i>
<b>limit</b>	100	<i>Optional - Default is 10000 when both startTime &amp; endTime are specified. 1000 when startTime or endTime is omitted.</i> Maximum number of records to be returned. 1-10000
<b>skip</b>	50	<i>Optional.</i> Skip the first <i>n</i> records returned. Can be used with <code>limit</code> to paginate results
<b>sort</b>	event-Time(DESC)	<i>Optional - Default is eventTime(ASC).</i> Comma delimited list of attributes to sort by. Optionally include sort direction in parentheses or default to ASC: <i>[ASC, DESC]</i>

## Request

```
GET /api/v1/events
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
[
  {
    "_class": "io.eagle.models.event.Event",
    "_id": "58d3508c8d6a80f2bd7eaf2e",
    "alarmState": "NONE",
    "message": "Processed 60 records from sensors.dat [4 parameters_
↪updated]",
    "eventType": "ACQUISITION",
    "eventTime": "2017-03-23T04:35:24.000Z",
    "nodeId": "57d799d86a559fb79215b5c1",
    "nodeType": "io.eagle.models.node.source.data.TextParser",
    "level": "DEBUG",
    "workspaceId": "57e0a4d8a19b65042ad079c2",
    "username": "System"
```

(continues on next page)

(continued from previous page)

```
    },
    {
      "_class": "io.eagle.models.event.Event",
      "_id": "58d0b9df0bb36d2965e5b745",
      "alarmState": "NONE",
      "message": "Rename node 'Cond' to 'Conductivity'",
      "eventType": "CONFIGURATION",
      "eventTime": "2017-03-21T05:27:59.000Z",
      "nodeId": "58bf63d57426a2b19ea63f54",
      "nodeType": "io.eagle.models.node.point.NumberPoint",
      "level": "INFO",
      "workspaceId": "577b5a821f704bb5a3f1411b",
      "username": "user@company.com"
    }
  ]
```

## Create or Delete events

You can *Create events* (page 492) or *Delete events* (page 493) for a specific node via the *Nodes Resource* (page 442).

### 3.3.5 Owners

Owners contain details about the eagle.io account associated with a Workspace or Device.

#### Quick Links

- *Owner Attributes* (page 508)
- *Retrieve owner* (page 509)
- *Update owner* (page 510)
- *Create owner groups* (page 515)
- *Update owner group* (page 516)
- *Remove owner group* (page 517)
- *Retrieve account usage* (page 517)

## Owner Attributes

Consumers of Owners should tolerate the addition of new attributes and variance in ordering of attributes with ease. Not all attributes appear in all contexts. It is generally safe to consider a nulled attribute, an empty set, and the absence of an attribute as the same thing. Restricted attributes are only available when using an API Key with **account** level access.

Standard attributes	Type	Update	Description
<b>_class</b>	String		Identifies resource type: <i>io.eagle.models.owner.Owner</i>
<b>_id</b>	ObjectId		Unique identifier for this owner
<b>accountName</b>	String	Yes	Name of the account
<b>accountType</b>	String		Type of account
<b>categories</b>	Array		Array of notification categories configured for this owner
<b>createdTime</b>	Time		<i>ISO8601</i> (page 581) timestamp the account was created.
<b>quality</b>	Object		Quality <i>codes</i> and configuration
<b>slug</b>	String		Used as part of Node full name path

Restricted attributes	Type	Update	Description
<b>administrators</b>	Array	Yes	Array of <i>administrator users</i> (page 512)
<b>groups</b>	Array	Yes	Array of <i>groups</i> (page 513) and associated users

## Retrieve owner

Retrieve the owner associated with the API key.

## Arguments

Argument	Example	Description
<b>attr</b>	_id,slug	<i>Optional.</i> Comma delimited list of attributes to include in response

## Request

```
GET /api/v1/owners/account
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "_class": "io.eagle.models.owner.Owner",
  "_id": "5424fbe96c4728a166f0d725",
  "accountName": "Environment Inc",
  "accountType": "STANDARD",
  "administrators": [
    {
      "user": "rlewis@company.com",
```

(continues on next page)

(continued from previous page)

```

        "name": {
            "first": "Rod",
            "last": "Lewis"
        },
        "lastLoginTime": "2021-12-16T03:17:35.170Z"
    },
    ],
    "categories": [
        {
            "name": "Maintenance",
            "index": 0,
            "_id": "52969367593a1a3a32000012"
        },
        {
            "name": "Operations",
            "index": 1,
            "_id": "52969367593a1a3a32000011"
        },
        {
            "name": "Critical",
            "index": 2,
            "_id": "52969367593a1a3a32000010"
        }
    ],
    "createdTime": "2022-01-26T05:38:49.167Z",
    "groups": [
        {
            "_id": "61bac74a84cac62fcbfe2c3c",
            "name": "Engineering team",
            "users": [
                {
                    "user": "jsmith@company.com",
                    "name": {
                        "first": "Jane",
                        "last": "Smith"
                    },
                    "lastLoginTime": "2022-01-04T11:30:22.310Z"
                }
            ]
        }
    ],
    "slug": "mypers42"
}

```

## Update owner

Update the owner associated with the API key. Updates are limited to the attributes listed with the *Update* flag. Multiple attributes can be updated in a single request.

The updated owner will be returned in the response if the request is successful. You can optionally limit the returned attributes by specifying the **attr** argument.

**Note:** Required API key permission: *Modify*. Required API key level: *Account*.

## Arguments

Argument	Example	Description
<b>attr</b>	<code>_id, _class</code>	<i>Optional</i> . Comma delimited list of attributes to include in successful response
<b>notify</b>	<code>TRUE</code>	<i>Optional - Default is FALSE</i> . Notify users when they have been granted/revoked administrator access or a new user profile created (as a result of this request)

## Request

```
PUT /api/v1/owners/account
```

```
{
  "accountName": "My Renamed Account"
}
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "_class": "io.eagle.models.owner.Owner",
  "_id": "5424fbe96c4728a166f0d725",
  "accountName": "My Renamed Account",
  ...
}
```

## Complex attribute updates

Certain attributes such as Administrator and Groups have specific update requirements:

- *Administrators* (page 512)
- *Groups* (page 513)

## Administrators

Administrators can be included in the update request. This will replace existing administrators.

If the user email address does not already exist in the system, a new user profile will be created and *profileActivateUrl* will be returned in the response. Use the *notify=TRUE* [argument](#) (page 511) to automatically send a welcome email to newly created users (or use the *profileActivateUrl* to manually complete the user profile setup).

When adding a new user (profile does not exist) you can optionally include attributes: *name*, *phone*, *time-zone*, *timezoneAdjustForDst* and *timeFormat*.

---

**Note:** Removing a user as an administrator will also remove their *user notification subscriptions* (page 471) for any workspace nodes they no longer have access to.

The account owner and account manager administrators (for managed accounts) automatically inherit implicit administrator access and are not included in the administrators array.

---

### Example replacing administrators

```
PUT /api/v1/owners/account
```

```
{
  "administrators": [
    {
      "user": "rlewis@company.com"
    },
    {
      "user": "new@company.com",
      "name": {
        "first": "New",
        "last": "User"
      },
      "phone": "+61000000001",
      "timezone": "Australia/Sydney",
      "timezoneAdjustForDst": true,
      "timeFormat": "YYYY-MM-DD HH:mm:ss"
    }
  ]
}
```

### Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```



```
{
  "administrators": [
    {
      "user": "rlewis@company.com",
      "name": {
        "first": "Rod",
        "last": "Lewis"
      },
      "lastLoginTime": "2021-12-16T03:17:35.170Z"
    },
    {
      "user": "new@company.com",
      "name": {
        "first": "New",
        "last": "User"
      },
      "profileActivateUrl": "https://app.eagle.io/auth/setupprofile/
↪ca4d1da0-8231-46df-af69-df1b2f1a8b5c"
    }
  ],
  ...
}
```

## Groups

Groups and associated users can be included in the update request. This will merge and replace existing groups.

For each group in the request, if an *\_id* attribute is provided it will be used to update an existing group (which allows the group name to be changed as part of the update). If the *\_id* attribute is not provided or does not match an existing group, the *name* attribute will be used to find an existing group to update.

---

**Note:** If an existing group is not found, a new group entity will be created. Any groups previously configured that do not match will be permanently removed.

Removing a group will also remove its *group notification subscriptions* (page 471).

---

When specifying the users associated with a group, if the user email address does not already exist in the system, a new user profile will be created and *profileActivateUrl* will be returned in the response. Use the *notify=TRUE* [argument](#) (page 511) to automatically send a welcome email to newly created users (or use the *profileActivateUrl* to manually complete the user profile setup).

When adding a new user (profile does not exist) you can optionally include attributes: *name*, *phone*, *time-zone*, *timezoneAdjustForDst* and *timeFormat*.

Group attributes	Type	Description
<b>_id</b>	ObjectId	<i>Optional.</i> Id of existing group to update.
<b>name</b>	String	<i>Required.</i> Name to assign to this group.
<b>users</b>	Array	<i>Required.</i> Array of users to assign to this group.

### Example updating groups

```
PUT /api/v1/owners/account
```

```
{
  "groups": [
    {
      "name": "New Group",
      "users": [
        {
          "user": "rlewis@company.com"
        },
        {
          "user": "new@company.com",
          "name": {
            "first": "New",
            "last": "User"
          },
          "phone": "+610000000001",
          "timezone": "Australia/Sydney",
          "timezoneAdjustForDst": true,
          "timeFormat": "YYYY-MM-DD HH:mm:ss"
        }
      ]
    },
    {
      "_id": "61bac74a84cac62fcbfe2c3c",
      "users": []
    }
  ]
}
```

### Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "groups": [
    {
      "_id": "61bac74a84cac62fcbfe47aa",
      "name": "New Group",
      "users": [
        {
          "user": "rlewis@company.com",
```

(continues on next page)

(continued from previous page)

```

        "name": {
            "first": "Rod",
            "last": "Lewis"
        },
        "lastLoginTime": "2021-12-16T03:17:35.170Z"
    },
    {
        "user": "new@company.com",
        "name": {
            "first": "New",
            "last": "User"
        },
        "profileActivateUrl": "https://app.eagle.io/auth/
↪setupprofile/ca4d1da0-8231-46df-af69-df1b2f1a8b5c"
    }
]
},
{
    "_id": "61bac74a84cac62fcbfe2c3c",
    "name": "Engineering team",
    "users": []
}
],
...
}

```

## Create owner groups

Create new owner group(s). The complete array of groups will be returned in the response if the request is successful.

Alternatively *update or replace multiple groups* (page 513) via owner update.

Group attributes	Type	Description
<b>name</b>	String	<i>Required.</i> Name to assign to this group.
<b>users</b>	Array	<i>Required.</i> Array of users to assign to this group.

## Request

```
POST /api/v1/owners/account/groups
```

```

[
  {
    "name": "New Group",
    "users": [
      {
        "user": "rlewis@company.com"
      }
    ]
  }
]

```

(continues on next page)

(continued from previous page)

```

    },
    {
      "user": "new@company.com",
      "name": {
        "first": "New",
        "last": "User"
      },
      "phone": "+61000000001",
      "timezone": "Australia/Sydney",
      "timezoneAdjustForDst": true,
      "timeFormat": "YYYY-MM-DD HH:mm:ss"
    }
  ]
},
{
  "name": "New Group 2",
  "users": []
},
]

```

## Response

```

HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8

```

## Update owner group

Update an existing owner group by its *\_id*. The updated group will be returned in the response if the request is successful.

Alternatively *update or replace multiple groups* (page 513) via owner update.

Group attributes	Type	Description
<b>name</b>	String	<i>Optional.</i> Name to assign to this group.
<b>users</b>	Array	<i>Optional.</i> Array of users to assign to this group.

## Request

```
PUT /api/v1/owners/account/groups/:id
```

```

{
  "name": "Updated Group",
  "users": [
    {

```

(continues on next page)

(continued from previous page)

```
      "user": "rlewis@company.com"
    }
  ]
}
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

## Remove owner group

Permanently delete an existing group by its *\_id*.

Alternatively *update or replace multiple groups* (page 513) via owner update.

## Request

```
DELETE /api/v1/owners/account/groups/:id
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

## Retrieve account usage

Retrieve account usage for an owner (including managed accounts) associated with the API key. TRIAL and DISABLED managed accounts are not included in Source usage.

---

**Note:** This resource is currently in development and may change prior to finalization. Required API key access level: *Account*. Not accessible using managed account API keys.

---

## Request

```
GET /api/v1/owners/account/usage
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "rates": {
    "attachmentsource": 0.7,
    "datasource": {
      "small": 0.3,
      "medium": 0.7,
      "large": 1
    },
    "sms": 1
  },
  "total": {
    "source": {
      "usage": 32.4,
      "usageLimit": 50
    },
    "sms": {
      "usage": 30,
      "usageLimit": 250
    }
  },
  "accounts": [
    {
      "_id": "5424fbe96c4728a166f0d725",
      "accountName": "My Personal Account",
      "accountType": "STANDARD",
      "usage": {
        "attachmentsource": 0.7,
        "datasource": {
          "small": 0.3,
          "medium": 1.4,
          "large": 20
        },
        "sms": 10
      }
    },
    {
      "_id": "5d7adc82e9e941211c57032c",
      "accountName": "Managed account",
      "accountType": "MANAGED",
      "usage": {
        "attachmentsource": 0,
        "datasource": {
          "small": 0,
          "medium": 7,
          "large": 3
        },
        "sms": 20
      }
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
}  
  }  
  ]  
}
```





## 4.1 Configuration

This section provides specifications for configuration file formats.

### 4.1.1 Ratings CSV

*Rating parameters* (page 156) support the **import** and **export** of rating configuration files from the user interface in a fixed CSV format as specified below.

The *HTTP API* (page 433) can additionally be used to read or update *rating parameter configuration* (page 461) in **JSON** (<http://json.org>) format.

Example *ratings.csv*

```
"2019-05-14T00:00:00.000Z", "table", 0:0, 1:10, 2:20  
"2020-09-25T13:30:00.000Z", "table", 0.1:15, 0.7:32.44, 1.22:122.91, 1.5:200  
"2020-10-10T00:00:00.000Z", "polynomial", a:1.1, b:2.5, c:1.115, d:4.01, e:0.85
```

### Rating CSV line specification

Every line in the ratings.csv file defines a single rating. The file can contain multiple lines/ratings. Columns (separated by comma ',') must use this strict format.

Column	Example	Description
<b>1</b>	“2020-09-25T13:30:00.000Z”	<i>ISO8601</i> (page 581) start timestamp this rating applies. <i>Note the double quotes around the timestamp.</i>
<b>2</b>	“table”	Rating type: [ <i>“table”</i> , <i>“polynomial”</i> ] <i>Note the double quotes around the text.</i>
<b>3+</b>	0.1:15	<b>Table:</b> Columns 3 onwards represent the rows in the rating table. Each column must use the format <b>&lt;input&gt;:&lt;result&gt;</b> which contains an <i>input</i> lookup value and <i>result</i> value separated by ‘:’. In the example shown <i>0.1</i> is the input lookup value and <i>15</i> is the result.
	a:1.1	<b>Polynomial:</b> Columns 3 onwards specifies the coefficients of a polynomial equation. Each column must use the format <b>&lt;coefficient&gt;:&lt;value&gt;</b> . The coefficient must be a single letter, ordered alphabetically. In the example shown <i>a</i> is the coefficient and <i>1.1</i> is its value.

---

**Note:** Rating configuration files require the ‘.csv’ file extension for import.

---

## 4.2 Expressions

Expressions provide a simple method to describe dynamic text content.

Our expressions are based on the open-source [Liquid](https://shopify.github.io/liquid/) (<https://shopify.github.io/liquid/>) template language with some custom filter additions.

Expressions can be categorized into *objects* (page 566), *filters* (page 566) and *tags* (page 567).

### 4.2.1 Filters

Liquid expressions implement business-logic independent filters that are typically implemented in [shopify/liquid](https://shopify.github.io/liquid/) (<https://shopify.github.io/liquid/>). This section contains the specification and examples for all the filters implemented by Liquid.

There's 40+ filters supported by Liquid. These filters can be categorized into these groups:

Cate- gories	Filters
Math	plus, minus, modulo, times, floor, ceil, round, divided_by, abs, decimal
String	append, prepend, capitalize, upcase, downcase, strip, lstrip, rstrip, strip_newlines, split, replace, replace_first, remove, remove_first, truncate, truncatewords
HTML/URL	escape, escape_once, url_encode, url_decode, strip_html, newline_to_br, link_to
Array	slice, map, sort, sort_natural, uniq, where, first, last, join, reverse
Time	time, time_plus, time_minus

#### abs

Returns the absolute value of a number.

Input

```
{{ -17 | abs }}
```

Output

```
17
```

Input

```
{{ 4 | abs }}
```

Output

```
4
```

`abs` will also work on a string that only contains a number:

Input

```
{{ "-19.86" | abs }}
```

Output

```
19.86
```

## append

Concatenates two strings and returns the concatenated value.

Input

```
{{ "/my/fancy/url" | append: ".html" }}
```

Output

```
/my/fancy/url.html
```

`append` can also be used with variables:

Input

```
{% assign filename = "/index.html" %}  
{{ "website.com" | append: filename }}
```

Output

```
website.com/index.html
```

## at\_least

Limits a number to a minimum value.

Input

```
{{ 4 | at_least: 5 }}
```

Output

```
5
```

Input

```
{{ 4 | at_least: 3 }}
```

Output

```
4
```

### **at\_most**

Limits a number to a maximum value.

Input

```
{{ 4 | at_most: 5 }}
```

Output

```
4
```

Input

```
{{ 4 | at_most: 3 }}
```

Output

```
3
```

### **capitalize**

Makes the first character of a string capitalized.

Input

```
{{ "title" | capitalize }}
```

Output

```
Title
```

capitalize only capitalizes the first character of a string, so later words are not affected:

Input

```
{{ "my great title" | capitalize }}
```

Output

```
My great title
```

## ceil

Rounds the input up to the nearest whole number. Liquid tries to convert the input to a number before the filter is applied.

Input

```
{{ 1.2 | ceil }}
```

Output

```
2
```

Input

```
{{ 2.0 | ceil }}
```

Output

```
2
```

Input

```
{{ 183.357 | ceil }}
```

Output

```
184
```

Here the input value is a string:

Input

```
{{ "3.5" | ceil }}
```

Output

```
4
```

## compact

Removes any `nil`/`null` values from an array.

For this example, assume `site.pages` is an array of content pages for a website, and some of these pages have an attribute called `category` that specifies their content category. If we map those categories to an array, some of the array items might be `nil` if any pages do not have a `category` attribute.

Input

```
{% assign site_categories = site.pages | map: "category" %}

{% for category in site_categories %}
- {{ category }}
{% endfor %}
```

### Output

```
- business
- celebrities
-
- lifestyle
- sports
-
- technology
```

By using `compact` when we create our `site_categories` array, we can remove all the `nil` values in the array.

### Input

```
{% assign site_categories = site.pages | map: "category" | compact %}

{% for category in site_categories %}
- {{ category }}
{% endfor %}
```

### Output

```
- business
- celebrities
- lifestyle
- sports
- technology
```

## concat

Concatenates (joins together) multiple arrays. The resulting array contains all the items from the input arrays.

### Input

```
{% assign fruits = "apples, oranges, peaches" | split: ", " %}
{% assign vegetables = "carrots, turnips, potatoes" | split: ", " %}

{% assign everything = fruits | concat: vegetables %}

{% for item in everything %}
- {{ item }}
{% endfor %}
```

### Output

```
- apples
- oranges
- peaches
- carrots
- turnips
- potatoes
```

You can string together `concat` filters to join more than two arrays:

### Input

```
{% assign furniture = "chairs, tables, shelves" | split: ", " %}

{% assign everything = fruits | concat: vegetables | concat: furniture %}

{% for item in everything %}
- {{ item }}
{% endfor %}
```

### Output

```
- apples
- oranges
- peaches
- carrots
- turnips
- potatoes
- chairs
- tables
- shelves
```

### Output

```
- business
- celebrities
-
- lifestyle
- sports
-
- technology
```

By using `compact` when we create our `site_categories` array, we can remove all the `nil` values in the array.

### Input

```
{% assign site_categories = site.pages | map: "category" | compact %}

{% for category in site_categories %}
- {{ category }}
{% endfor %}
```



## Output

```
- business
- celebrities
- lifestyle
- sports
- technology
```

## date

Converts a timestamp into another date format using `strftime` (<http://www.ruby-doc.org/core/Time.html#method-i-strftime>) syntax.

**Note:** The *time* (page 545) filter is recommended to be used instead of *date* it is more versatile and correctly applies timezone and DST to the formatted output.

## Input

```
{{ article.published_at | date: "%a, %b %d, %y" }}
```

## Output

```
Fri, Jul 17, 15
```

## Input

```
{{ article.published_at | date: "%Y" }}
```

## Output

```
2015
```

`date` works on strings if they contain well-formatted dates:

## Input

```
{{ "March 14, 2016" | date: "%b %d, %y" }}
```

## Output

```
Mar 14, 16
```

To get the current time, pass the special word `"now"` (or `"today"`) to `date`:

## Input

```
This page was last updated at {{ "now" | date: "%Y-%m-%d %H:%M" }}.
```

## Output

This page was last updated at 2020-03-25 15:57.

## decimal

Decimal uses tokens to format the number to a set number of decimal places. 0 will always display a digit. # will only display a digit if it exists. Liquid tries to convert the input to a number before the filter is applied.

Input

```
{{ 1.23 | decimal: "0.#" }}
```

Output

```
1.2
```

Input

```
{{ 1000000 | decimal: "#,###.#" }}
```

Output

```
1,000,000
```

Input

```
{{ "2" | decimal: "0.00" }}
```

Output

```
2.00
```

## default

Allows you to specify a fallback in case a value doesn't exist. default will show its value if the left side is false or empty (string or Array).

In this example, product\_price is not defined, so the default value is used.

Input

```
{{ product_price | default: 2.99 }}
```

Output

```
2.99
```

In this example, product\_price is defined, so the default value is not used.

Input

```
{% assign product_price = 4.99 %}  
{{ product_price | default: 2.99 }}
```

Output

```
4.99
```

In this example, `product_price` is empty, so the default value is used.

Input

```
{% assign product_price = "" %}  
{{ product_price | default: 2.99 }}
```

Output

```
2.99
```

## divided\_by

Divides a number by another number.

Input

```
{{ 16 | divided_by: 4 }}
```

Output

```
4
```

Input

```
{{ 5 | divided_by: 3 }}
```

Output

```
1.6666666666666667
```

## downcase

Makes each character in a string lowercase. It has no effect on strings which are already all lowercase.

Input

```
{{ "Parker Moore" | downcase }}
```

Output

```
parker moore
```

Input

```
{{ "apple" | downcase }}
```

Output

```
apple
```

## escape

Escapes a string by replacing HTML special characters with escape sequences. It doesn't change strings that don't have anything to escape.

Input

```
{{ "Have you read 'James & the Giant Peach'?" | escape }}
```

Output

```
Have you read &#39;James &amp; the Giant Peach&#39;?
```

Input

```
{{ "Tetsuro Takara" | escape }}
```

Output

```
Tetsuro Takara
```

## escape\_once

Escapes a string without changing existing escaped entities. It doesn't change strings that don't have anything to escape.

Input

```
{{ "1 < 2 & 3" | escape_once }}
```

Output

```
1 &lt; 2 &amp; 3
```

Input

```
{{ "{ \"1 < 2 & 3\" | escape }" | escape_once }}
```

Output

```
{{ "1 < 2 & 3" | escape }}
```

## first

Returns the first item of an array.

Input

```
{{ "Ground control to Major Tom." | split: " " | first }}
```

Output

```
Ground
```

Input

```
{% assign my_array = "zebra, octopus, giraffe, tiger" | split: ", " %}
{{ my_array.first }}
```

Output

```
zebra
```

You can use `first` with dot notation when you need to use the filter inside a tag:

```
{% if my_array.first == "zebra" %}
  Here comes a zebra!
{% endif %}
```

## floor

Rounds the input down to the nearest whole number. Liquid tries to convert the input to a number before the filter is applied.

Input

```
{{ 1.2 | floor }}
```

Output

```
1
```

Input

```
{{ 2.0 | floor }}
```

Output

```
2
```

Input

```
{{ 183.357 | floor }}
```

Output

```
183
```

Here the input value is a string:

Input

```
{{ "3.5" | floor }}
```

Output

```
3
```

## join

Combines the items in an array into a single string using the argument as a separator.

Input

```
{% assign beatles = "John, Paul, George, Ringo" | split: ", " %}  
{{ beatles | join: " and " }}
```

Output

```
John and Paul and George and Ringo
```

## json

Convert variables to JSON text.

Input

```
{% assign arr = "foo bar coo" | split: " " %}  
{{ arr | json }}
```

Output

```
["foo","bar","coo"]
```

## last

Returns the last item of an array.

Input

```
{{ "Ground control to Major Tom." | split: " " | last }}
```

Output

```
Tom.
```

Input

```
{% assign my_array = "zebra, octopus, giraffe, tiger" | split: ", " %}  
{{ my_array.last }}
```

Output

```
tiger
```

You can use `last` with dot notation when you need to use the filter inside a tag:

```
{% if my_array.last == "tiger" %}  
  There goes a tiger!  
{% endif %}
```

## link\_to

Generates an HTML link for HTML content.

Input

```
{{ "Ground control to Major Tom." | link_to: "http://majortom.com" }}
```

Output

```
<a href="http://majortom.com">Ground control to Major Tom.</a>
```

## lstrip

Removes all whitespace (tabs, spaces, and newlines) from the left side of a string. It does not affect spaces between words.

Input

```
BEGIN{{ "      So much room for activities!      " | lstrip }}END
```

Output

```
BEGINSo much room for activities!          END
```

## map

Creates an array of values by extracting the values of a named property from another object.

In this example, assume the object `site.pages` contains all the metadata for a website. Using `assign` with the `map` filter creates a variable that contains only the values of the `category` properties of everything in the `site.pages` object.

Input

```
{% assign all_categories = site.pages | map: "category" %}  
  
{% for item in all_categories %}  
- {{ item }}  
{% endfor %}
```

Output

```
- business  
- celebrities  
- lifestyle  
- sports  
- technology
```

## minus

Subtracts a number from another number.

Input

```
{{ 4 | minus: 2 }}
```

Output

```
2
```

Input

```
{{ 16 | minus: 4 }}
```

Output

```
12
```

Input



```
{{ 183.357 | minus: 12 }}
```

Output

```
171.357
```

## modulo

Returns the remainder of a division operation.

Input

```
{{ 3 | modulo: 2 }}
```

Output

```
1
```

Input

```
{{ 24 | modulo: 7 }}
```

Output

```
3
```

Input

```
{{ 183.357 | modulo: 12 }}
```

Output

```
3.3569999999999993
```

## newline\_to\_br

Replaces every newline (`\n`) in a string with an HTML line break (`<br />`).

Input

```
{% capture string_with_newlines %}
Hello
there
{% endcapture %}

{{ string_with_newlines | newline_to_br }}
```

Output

```
<br/>Hello<br/>there<br/>
```

## plus

Adds a number to another number.

Input

```
{{ 4 | plus: 2 }}
```

Output

```
6
```

Input

```
{{ 16 | plus: 4 }}
```

Output

```
20
```

Input

```
{{ 183.357 | plus: 12 }}
```

Output

```
195.357
```

## prepend

Adds the specified string to the beginning of another string.

Input

```
{{ "apples, oranges, and bananas" | prepend: "Some fruit: " }}
```

Output

```
Some fruit: apples, oranges, and bananas
```

prepend can also be used with variables:

Input

```
{% assign url = "example.com" %}  
{{ "/index.html" | prepend: url }}
```

Output

```
example.com/index.html
```

## remove

Removes every occurrence of the specified substring from a string.

Input

```
{{ "I strained to see the train through the rain" | remove: "rain" }}
```

Output

```
I strained to see the t through the
```

## remove\_first

Removes only the first occurrence of the specified substring from a string.

Input

```
{{ "I strained to see the train through the rain" | remove_first: "rain" }}
```

Output

```
I strained to see the t through the rain
```

## replace

Replaces every occurrence of the first argument in a string with the second argument.

Input

```
{{ "Take my protein pills and put my helmet on" | replace: "my", "your" }}
```

Output

```
Take your protein pills and put your helmet on
```

## replace\_first

Replaces only the first occurrence of the first argument in a string with the second argument.

Input

```
{{ "Take my protein pills and put my helmet on" | replace_first: "my", "your"
↪ }}
```

## Output

```
Take your protein pills and put my helmet on
```

## reverse

Reverses the order of the items in an array. `reverse` cannot reverse a string.

### Input

```
{% assign my_array = "apples, oranges, peaches, plums" | split: ", " %}  
{{ my_array | reverse | join: ", " }}
```

### Output

```
plums, peaches, oranges, apples
```

Although `reverse` cannot be used directly on a string, you can split a string into an array, reverse the array, and rejoin it by chaining together filters:

### Input

```
{{ "Ground control to Major Tom." | split: "" | reverse | join: "" }}
```

### Output

```
.moT roJaM ot lortnoc dnuoRg
```

## round

Rounds a number to the nearest integer or, if a number is passed as an argument, to that number of decimal places.

### Input

```
{{ 1.2 | round }}
```

### Output

```
1
```

### Input

```
{{ 2.7 | round }}
```

### Output

```
3
```

### Input

```
{{ 183.357 | round: 2 }}
```

### Output

```
183.36
```

## rstrip

Removes all whitespace (tabs, spaces, and newlines) from the right side of a string. It does not affect spaces between words.

### Input

```
BEGIN{{ "          So much room for activities!          " | rstrip }}END
```

### Output

```
BEGIN          So much room for activities!END
```

## size

Returns the number of characters in a string or the number of items in an array.

### Input

```
{{ "Ground control to Major Tom." | size }}
```

### Output

```
28
```

### Input

```
{% assign my_array = "apples, oranges, peaches, plums" | split: ", " %}
{{ my_array.size }}
```

### Output

```
4
```

You can use `size` with dot notation when you need to use the filter inside a tag:

```
{% if site.pages.size > 10 %}
  This is a big website!
{% endif %}
```

## slice

Returns a substring of 1 character beginning at the index specified by the first argument. An optional second argument specifies the length of the substring to be returned.

String indices are numbered starting from 0.

Input

```
{{ "Liquid" | slice: 0 }}
```

Output

```
L
```

Input

```
{{ "Liquid" | slice: 2 }}
```

Output

```
q
```

Input

```
{{ "Liquid" | slice: 2, 5 }}
```

Output

```
quid
```

If the first argument is a negative number, the indices are counted from the end of the string:

Input

```
{{ "Liquid" | slice: -3, 2 }}
```

Output

```
ui
```

## sort

Sorts items in an array in case-sensitive order.

Input

```
{% assign my_array = "zebra, octopus, giraffe, Sally Snake" | split: ", " %}  
{{ my_array | sort | join: ", " }}
```

Output

```
Sally Snake, giraffe, octopus, zebra
```

An optional argument specifies which property of the array's items to use for sorting.

```
{% assign products_by_price = collection.products | sort: "price" %}
{% for product in products_by_price %}
  <h4>{{ product.title }}</h4>
{% endfor %}
```

## sort\_natural

Sorts items in an array in case-insensitive order.

Input

```
{% assign my_array = "zebra, octopus, giraffe, Sally Snake" | split: ", " %}
{{ my_array | sort_natural | join: ", " }}
```

Output

```
giraffe, octopus, Sally Snake, zebra
```

An optional argument specifies which property of the array's items to use for sorting.

```
{% assign products_by_company = collection.products | sort_natural: "company"
↪ %}
{% for product in products_by_company %}
  <h4>{{ product.title }}</h4>
{% endfor %}
```

## split

Divides a string into an array using the argument as a separator. `split` is commonly used to convert comma-separated items from a string to an array.

Input

```
{% assign beatles = "John, Paul, George, Ringo" | split: ", " %}

{% for member in beatles %}
  {{ member }}
{% endfor %}
```

Output

```
John
```

```
Paul
```

(continues on next page)

(continued from previous page)

```
George
```

```
Ringo
```

## strip

Removes all whitespace (tabs, spaces, and newlines) from both the left and right sides of a string. It does not affect spaces between words.

Input

```
BEGIN{{ "          So much room for activities!          " | strip }}END
```

Output

```
BEGINSo much room for activities!END
```

## strip\_html

Removes any HTML tags from a string.

Input

```
{{ "Have <em>you</em> read <strong>Ulysses</strong>?" | strip_html }}
```

Output

```
Have you read Ulysses?
```

## strip\_newlines

Removes any newline characters (line breaks) from a string.

Input

```
{% capture string_with_newlines %}  
Hello  
there  
{% endcapture %}  
  
{{ string_with_newlines | strip_newlines }}
```

Output

```
Hellothere
```



## time

Converts an *ISO8601 timestamp* (page 581) or *OPC Relative time* (page 585) into another time format using *time format tokens* (page 583).

Input

```
{{ node.createdTime | time }}
```

Output

```
2021-06-01 10:50:00
```

Input

```
{{ "2021-06-01T10:50:00.000Z" | time }}
```

Output

```
2021-06-01 10:50:00
```

Here the input value is an *OPC Relative time* (page 585)

Input

```
{{ "NOW" | time }}
```

Output

```
2021-07-18 14:30:00
```

## Format

Format the output using *time format tokens* (page 583). The default format is YYYY-MM-DD HH:mm:ss.

Input

```
{{ node.createdTime | time: "DD MMMM YYYY" }}
```

Output

```
01 June 2021
```

## Timezone and DST

Provide *timezone* (page 581) for output in regional time. The default timezone is `Etc/UTC`.

Input

```
{{ "2021-01-01T01:50:00.000Z" | time: "YYYY-MM-DD HH:mm:ss", "Australia/Sydney"
↪" }}
```

Output

```
2021-01-01 12:50:00
```

The timezone will always be adjusted to DST (Daylight Savings Time) but this can be disabled to ensure a fixed UTC offset by including `false` as an additional option.

Input

```
{{ "2021-01-01T01:50:00.000Z" | time: "YYYY-MM-DD HH:mm:ss", "Australia/Sydney"
↪", false }}
```

Output

```
2021-01-01 11:50:00
```

---

**Note:** Reports will default to the *Timezone* and *DST* specified in its properties dialog.

---

## time\_minus

Subtracts an interval of time from an *ISO8601 timestamp* (page 581).

Interval units include: `years`, `months`, `weeks`, `days`, `hours`, `minutes`, `seconds`, `milliseconds`

Input

```
{{ node.createdTime | time_minus: 1, "years" }}
```

Output

```
2020-06-01T10:50:00.000Z
```

The `time_minus` filter can be applied multiple times

Input

```
{{ "2021-06-01T10:50:00.000Z" | time_minus: 1, "years" | time_minus: 2, "days"
→ }}
```

Output

```
2020-05-30T10:50:00.000Z
```

Format the timestamp by applying the [time](#) (page 545) filter last

Input

```
{{ "2021-06-01T10:50:00.000Z" | time_minus: 1, "years" | time: "MMMM YYYY" }}
```

Output

```
June 2020
```

## time\_plus

Adds interval of time to an *ISO8601 timestamp* (page 581).

Interval units include: years, months, weeks, days, hours, minutes, seconds, milliseconds

Input

```
{{ node.createdTime | time_plus: 1, "years" }}
```

Output

```
2022-06-01T10:50:00.000Z
```

The `time_plus` filter can be applied multiple times

Input

```
{{ "2021-06-01T10:50:00.000Z" | time_plus: 1, "years" | time_plus: 2, "days" }}
```

Output

```
2022-06-03T10:50:00.000Z
```

Format the timestamp by applying the *time* (page 545) filter last

Input

```
{{ "2021-06-01T10:50:00.000Z" | time_plus: 1, "years" | time: "MMMM YYYY" }}
```

Output

```
June 2022
```

## times

Multiplies a number by another number.

Input

```
{{ 3 | times: 2 }}
```

Output

```
6
```

Input

```
{{ 24 | times: 7 }}
```

Output

```
168
```

Input

```
{{ 183.357 | times: 12 }}
```

Output

```
2200.284
```

## truncate

Shortens a string down to the number of characters passed as an argument. If the specified number of characters is less than the length of the string, an ellipsis (...) is appended to the string and is included in the character count.

### Input

```
{{ "Ground control to Major Tom." | truncate: 20 }}
```

### Output

```
Ground control to...
```

## Custom ellipsis

`truncate` takes an optional second argument that specifies the sequence of characters to be appended to the truncated string. By default this is an ellipsis (...), but you can specify a different sequence.

The length of the second argument counts against the number of characters specified by the first argument. For example, if you want to truncate a string to exactly 10 characters, and use a 3-character ellipsis, use **13** for the first argument of `truncate`, since the ellipsis counts as 3 characters.

### Input

```
{{ "Ground control to Major Tom." | truncate: 25, ", and so on" }}
```

### Output

```
Ground control, and so on
```

## No ellipsis

You can truncate to the exact number of characters specified by the first argument and avoid showing trailing characters by passing a blank string as the second argument:

### Input

```
{{ "Ground control to Major Tom." | truncate: 20, "" }}
```

### Output

```
Ground control to Ma
```

## truncatewords

Shortens a string down to the number of words passed as an argument. If the specified number of words is less than the number of words in the string, an ellipsis (...) is appended to the string.

Input

```
{{ "Ground control to Major Tom." | truncatewords: 3 }}
```

Output

```
Ground control to...
```

## Custom ellipsis

truncatewords takes an optional second argument that specifies the sequence of characters to be appended to the truncated string. By default this is an ellipsis (...), but you can specify a different sequence.

Input

```
{{ "Ground control to Major Tom." | truncatewords: 3, "--" }}
```

Output

```
Ground control to--
```

## No ellipsis

You can avoid showing trailing characters by passing a blank string as the second argument:

Input

```
{{ "Ground control to Major Tom." | truncatewords: 3, "" }}
```

Output

```
Ground control to
```

## uniq

Removes any duplicate elements in an array.

### Input

```
{% assign my_array = "ants, bugs, bees, bugs, ants" | split: ", " %}
{{ my_array | uniq | join: ", " }}
```

### Output

```
ants, bugs, bees
```

## upcase

Makes each character in a string uppercase. It has no effect on strings which are already all uppercase.

### Input

```
{{ "Parker Moore" | upcase }}
```

### Output

```
PARKER MOORE
```

### Input

```
{{ "APPLE" | upcase }}
```

### Output

```
APPLE
```

## url\_decode

Decodes a string that has been encoded as a URL.

### Input

```
{{ "%27Stop%21%27+said+Fred" | url_decode }}
```

### Output

```
'Stop!' said Fred
```

## url\_encode

Converts any URL-unsafe characters in a string into percent-encoded characters.

### Input

```
{{ "john@liquid.com" | url_encode }}
```

### Output

```
john%40liquid.com
```

### Input

```
{{ "Tetsuro Takara" | url_encode }}
```

### Output

```
Tetsuro+Takara
```

## where

Creates an array including only the objects with a given property value, or any `truthy` value by default.

In this example, assume you have a list of products and you want to show your kitchen products separately. Using `where`, you can create an array containing only the products that have a `"type"` of `"kitchen"`.

### Input

```
All products:
{% for product in products %}
- {{ product.title }}
{% endfor %}

{% assign kitchen_products = products | where: "type", "kitchen" %}

Kitchen products:
{% for product in kitchen_products %}
- {{ product.title }}
{% endfor %}
```

### Output

```
All products:
- Vacuum
- Spatula
- Television
- Garlic press

Kitchen products:
- Spatula
- Garlic press
```

Say instead you have a list of products and you only want to show those that are available to buy. You can `where` with a property name but no target value to include all products with a **truthy** `"available"` value.

### Input



```
All products:
{% for product in products %}
- {{ product.title }}
{% endfor %}

{% assign available_products = products | where: "available" %}

Available products:
{% for product in available_products %}
- {{ product.title }}
{% endfor %}
```

### Output

```
All products:
- Coffee mug
- Limited edition sneakers
- Boring sneakers

Available products:
- Coffee mug
- Boring sneakers
```

The `where` filter can also be used to find a single object in an array when combined with the `first` filter. For example, say you want to show off the shirt in your new fall collection.

### Input

```
{% assign new_shirt = products | where: "type", "shirt" | first %}

Featured product: {{ new_shirt.title }}
```

### Output

```
Featured product: Hawaiian print sweater vest
```

## 4.2.2 Tags

Liquid expressions implement business-logic independent tags that are typically implemented in [shopify/liquid](https://shopify.github.io/liquid/) (<https://shopify.github.io/liquid/>). This section contains the specification and examples for all the tags implemented by Liquid.

There're are dozens of tags supported by Liquid, with all tags described at [shopify/liquid](https://shopify.github.io/liquid/) (<https://shopify.github.io/liquid/>). These tags can be categorized into these groups:

Category	Purpose	Tags
Iteration	iterate over a collection	for, cycle, tablerow
Control Flow	control the execution branch of template rendering	if, unless, elsif, else, case, when
Variable	define and alter variables	assign, increment, decrement
Language	temporarily disable Liquid syntax	raw, comment

## assign

Creates a new variable.

### Input

```
{% assign my_variable = false %}  
{% if my_variable != true %}  
  This statement is valid.  
{% endif %}
```

### Output

```
This statement is valid.
```

Wrap a variable value in quotations " to save it as a string.

### Input

```
{% assign foo = "bar" %}  
{{ foo }}
```

### Output

```
bar
```

## capture

Captures the string inside of the opening and closing tags and assigns it to a variable. Variables created through `capture` are strings.

### Input

```
{% capture my_variable %}I am being captured.{% endcapture %}  
{{ my_variable }}
```

### Output

```
I am being captured.
```

Using `capture`, you can create complex strings using other variables created with `assign`:

### Input

```
{% assign favorite_food = "pizza" %}  
{% assign age = 35 %}  
  
{% capture about_me %}  
I am {{ age }} and my favorite food is {{ favorite_food }}.  
{% endcapture %}  
  
{{ about_me }}
```

## Output

```
I am 35 and my favourite food is pizza.
```

## case

Creates a switch statement to compare a variable with different values. `case` initializes the switch statement, and `when` compares its values.

## Input

```
{% assign handle = "cake" %}
{% case handle %}
  {% when "cake" %}
    This is a cake
  {% when "cookie" %}
    This is a cookie
  {% else %}
    This is not a cake nor a cookie
{% endcase %}
```

## Output

```
This is a cake
```

## comment

Allows you to leave un-rendered code inside a Liquid template. Any text within the opening and closing `comment` blocks will not be printed, and any Liquid code within will not be executed.

## Input

```
Anything you put between {% comment %} and {% endcomment %} tags
is turned into a comment.
```

## Output

```
Anything you put between tags is turned into a comment.
```

## cycle

Loops through a group of strings and prints them in the order that they were passed as arguments. Each time `cycle` is called, the next string argument is printed.

## Input

```
{% cycle "one", "two", "three" %}
{% cycle "one", "two", "three" %}
{% cycle "one", "two", "three" %}
{% cycle "one", "two", "three" %}
```

### Output

```
one
two
three
one
```

Uses for `cycle` include:

- applying odd/even classes to rows in a table
- applying a unique class to the last product thumbnail in a row

## Parameters

`cycle` accepts a “cycle group” parameter in cases where you need multiple `cycle` blocks in one template. If no name is supplied for the cycle group, then it is assumed that multiple calls with the same parameters are one group.

### Input

```
{% cycle "first": "one", "two", "three" %}
{% cycle "second": "one", "two", "three" %}
{% cycle "second": "one", "two", "three" %}
{% cycle "first": "one", "two", "three" %}
```

### Output

```
one
one
two
two
```

## decrement

Creates a new number variable, and decreases its value by one every time it is called. The first value is `-1`.

### Input

```
{% decrement variable %}
{% decrement variable %}
{% decrement variable %}
```

### Output

```
-1
-2
-3
```

Like *increment* (page 562), variables declared inside `decrement` are independent from variables created through *assign* (page 554) or *capture* (page 554).

## for

Iteration tags run blocks of code repeatedly.

### Basic Usage

#### for...in

Repeatedly executes a block of code. For a full list of attributes available within a `for` loop, see *forloop* (page 559).

### Input

```
{% for product in collection.products %}
  {{ product.title }}
{% endfor %}
```

### Output

```
hat shirt pants
```

For loops can iterate over arrays, hashes, and *ranges of integers* (page 560).

When iterating a hash, `item[0]` contains the key, and `item[1]` contains the value:

### Input

```
{% for item in hash %}
  * {{ item[0] }}: {{ item[1] }}
{% endfor %}
```

### Output

```
* key1: value1
* key2: value2
```

## else

Specifies a fallback case for a `for` loop which will run if the loop has zero length.

### Input

```
{% for product in collection.products %}
  {{ product.title }}
{% else %}
  The collection is empty.
{% endfor %}
```

### Output

```
The collection is empty.
```

## break

Causes the loop to stop iterating when it encounters the `break` tag.

### Input

```
{% for i in (1..5) %}
  {% if i == 4 %}
    {% break %}
  {% else %}
    {{ i }}
  {% endif %}
{% endfor %}
```

### Output

```
1 2 3
```

## continue

Causes the loop to skip the current iteration when it encounters the `continue` tag.

### Input

```
{% for i in (1..5) %}
  {% if i == 4 %}
    {% continue %}
  {% else %}
```

(continues on next page)

(continued from previous page)

```

    {{ i }}
  {% endif %}
{% endfor %}

```

**Output**

```
1 2 3 5
```

**forloop**

There's a `forloop` object available inside `for` loops. It's used to indicate the current state of `for` loop.

The `forloop.first`, `forloop.last` and `forloop.length` property:

**Input**

```

{% for i in (1..5) %}
  {%- if forloop.first == true -%} First
  {%- elsif forloop.last == true -%} Last
  {%- else -%} {{ forloop.length }}
  {%- endif %}
{% endfor -%}

```

**Output**

```

First
5
5
5
Last

```

The `forloop.index`, `forloop.index0`, `forloop.rindex` and `forloop.rindex0` property:

**Input**

```

index index0 rindex rindex0
{% for i in (1..5) %}
  {{- forloop.index }}      {{ forloop.index0 }}      {{ forloop.rindex }}
  ↪ {{ forloop.rindex0 }}
{% endfor -%}

```

**Output**

```

index index0 rindex rindex0
1      0      5      4
2      1      4      3
3      2      3      2
4      3      2      1
5      4      1      0

```

## Parameters

### limit

Limits the loop to the specified number of iterations.

#### Input

```
<!-- if array = [1,2,3,4,5,6] -->
{% for item in array limit:2 %}
  {{ item }}
{% endfor %}
```

#### Output

```
1 2
```

### offset

Begins the loop at the specified index.

#### Input

```
<!-- if array = [1,2,3,4,5,6] -->
{% for item in array offset:2 %}
  {{ item }}
{% endfor %}
```

#### Output

```
3 4 5 6
```

### range

Defines a range of numbers to loop through. The range can be defined by both literal and variable numbers.

#### Input

```
{% for i in (3..5) %}
  {{ i }}
{% endfor %}

{% assign num = 4 %}
{% for i in (1..num) %}
  {{ i }}
{% endfor %}
```



### Output

```
3 4 5
1 2 3 4
```

### reversed

Reverses the order of the loop. Note that this flag's spelling is different from the filter `reverse`.

### Input

```
<!-- if array = [1,2,3,4,5,6] -->
{% for item in array reversed %}
  {{ item }}
{% endfor %}
```

### Output

```
6 5 4 3 2 1
```

### if

Executes a block of code only if a certain condition is `true`.

### Input

```
{% if product.title == "Awesome Shoes" %}
  These shoes are awesome!
{% endif %}
```

### Output

```
These shoes are awesome!
```

### elsif / else

Adds more conditions within an `if` or `unless` block.

### Input

```
{% if customer.name == "kevin" %}
  Hey Kevin!
{% elsif customer.name == "anonymous" %}
  Hey Anonymous!
```

(continues on next page)

(continued from previous page)

```
{% else %}  
  Hi Stranger!  
{% endif %}
```

### Output

```
Hey Anonymous!
```

## increment

Creates a new number variable, and increases its value by one every time it is called. The first value is 0.

### Input

```
{% increment my_counter %}  
{% increment my_counter %}  
{% increment my_counter %}
```

### Output

```
0  
1  
2
```

Variables created through the `increment` tag are independent from variables created through `assign` or `capture`.

In the example below, a variable named “var” is created through `assign`. The `increment` tag is then used several times on a variable with the same name. Note that the `increment` tag does not affect the value of “var” that was created through `assign`.

### Input

```
{% assign var = 10 %}  
{% increment var %}  
{% increment var %}  
{% increment var %}  
{{ var }}
```

### Output

```
0  
1  
2  
10
```

## raw

Raw temporarily disables tag processing. This is useful for generating content (eg, Mustache, Handlebars) which uses conflicting syntax.

### Input

```
{% raw %}
In Handlebars, {{ this }} will be HTML-escaped, but
{{{ that }}} will not.
{% endraw %}
```

### Output

```
In Handlebars, {{ this }} will be HTML-escaped, but {{{ that }}} will not.
```

## tablerow

Generates an HTML table. Must be wrapped in opening <table> and closing </table> HTML tags.

### Basic Usage

#### Input

```
<table>
{% tablerow product in collection.products %}
  {{ product.title }}
{% endtablerow %}
</table>
```

#### Output

```
<table>
  <tr class="row1">
    <td class="col1">
      Cool Shirt
    </td>
    <td class="col2">
      Alien Poster
    </td>
    <td class="col3">
      Batman Poster
    </td>
    <td class="col4">
      Bullseye Shirt
    </td>
    <td class="col5">
      Another Classic Vinyl
    </td>
```

(continues on next page)

(continued from previous page)

```
<td class="col6">
  Awesome Jeans
</td>
</tr>
</table>
```

## Parameters

### cols

Defines how many columns the tables should have.

#### Input

```
{% tablerow product in collection.products cols:2 %}
  {{ product.title }}
{% endtablerow %}
```

#### Output

```
<table>
  <tr class="row1">
    <td class="col1">
      Cool Shirt
    </td>
    <td class="col2">
      Alien Poster
    </td>
  </tr>
  <tr class="row2">
    <td class="col1">
      Batman Poster
    </td>
    <td class="col2">
      Bullseye Shirt
    </td>
  </tr>
  <tr class="row3">
    <td class="col1">
      Another Classic Vinyl
    </td>
    <td class="col2">
      Awesome Jeans
    </td>
  </tr>
</table>
```

## limit

Exits the tablerow after a specific index.

```
{% tablerow product in collection.products cols:2 limit:3 %}
  {{ product.title }}
{% endtablerow %}
```

## offset

Starts the tablerow after a specific index.

```
{% tablerow product in collection.products cols:2 offset:3 %}
  {{ product.title }}
{% endtablerow %}
```

## range

Defines a range of numbers to loop through. The range can be defined by both literal and variable numbers.

```
<!--variable number example-->

{% assign num = 4 %}
<table>
{% tablerow i in (1..num) %}
  {{ i }}
{% endtablerow %}
</table>

<!--literal number example-->

<table>
{% tablerow i in (3..5) %}
  {{ i }}
{% endtablerow %}
</table>
```

## unless

The opposite of *if* (page 561) – executes a block of code only if a certain condition is **not** met.

Input

```
{% unless product.title == "Awesome Shoes" %}
  These shoes are not awesome.
{% endunless %}
```

Output

```
These shoes are not awesome.
```

This would be the equivalent of doing the following:

```
{% if product.title != "Awesome Shoes" %}  
  These shoes are not awesome.  
{% endif %}
```

## 4.2.3 Overview

### Objects

Objects describe where to insert dynamic content in your expression and are denoted by double curly braces: `{{ and }}`.

The following objects are available to your expression:

<b>NOW</b>	Current time (in timezone of Source)
<b>THIS</b>	Reference to Node of current expression
<b>SOURCE</b>	Reference to Source of current expression
<b>LOCATION</b>	Reference to Location of current expression
<b>WORKSPACE</b>	Reference to Workspace of current expression

In the following example, the expression refers to an object called `WORKSPACE`, which has an attribute called `name`, and that object attribute contains the text `Workspace 01`.

Input

```
{{ WORKSPACE.name }}
```

Output

```
Workspace 01
```

---

**Note:** A full list of available object attributes are described [here](#) (page 444).

---

### Filters

Filters change the output of an object. They are used within an output and are separated by a `|`.

### Input

```
{{ "/my/fancy/url" | append: ".html" }}
```

### Output

```
/my/fancy/url.html
```

Multiple filters can be used on one output. They are applied from left to right.

### Input

```
{{ "adam!" | capitalize | prepend: "Hello " }}
```

### Output

```
Hello Adam!
```

---

**Note:** A full list of available filters are described [here](#) (page 523)

---

## Tags

Tags create the logic and control flow for templates. They are denoted by curly braces and percent signs: {% and %}.

The markup used in tags does not produce any visible text. This means that you can assign variables and create conditions and loops without showing any of the logic in the content.

### Input

```
{% if user %}  
  Hello {{ user.name }}!  
{% endif %}
```

### Output

```
Hello Adam!
```

---

**Note:** A full list of available tags are described [here](#) (page 553)

---

## Operators

Liquid operators are very simple and different. There're 2 types of operators supported:

- Comparison operators: `==`, `!=`, `>`, `<`, `>=`, `<=`
- Logic operators: `or`, `and`, `contains`

Thus numerical operators are not supported and you cannot even plus two numbers like this `{{ a + b }}`, instead we need a filter `{{ a | plus: b }}`.

## Precedence

1. Comparison operators. All comparison operations have the same precedence and higher than logic operators.
2. Logic operators. All logic operators have the same precedence.
3. Logic operators are evaluated from right to left, see [shopify docs](#) (<https://help.shopify.com/en/themes/liquid/basics/operators#order-of-operations>).

## Truthy and Falsy

According to [shopify documentation](#) (<https://shopify.github.io/liquid/basics/truthy-and-falsy/>) everything other than `false` and `nil/null` is truthy. Our implementation is slightly different:

value	truthy	falsy
true	✓	
false		✓
nil/null		✓
undefined		✓
string	✓	
empty string	✓	
0	✓	
integer	✓	
float	✓	
array	✓	
empty array	✓	



## 4.3 Historic Data

When working with *time series* or historic data there are several important topics to understand:

### 4.3.1 Aggregates

Historic Data Aggregates provide a method to resample raw data over a specified period into Intervals and then perform a calculation on the data in each interval.

#### Advantages of aggregation:

- Reduces the size of the dataset. Charts and tabular data can be loaded faster
- Easy generation of advanced statistical reports and charts. eg. Daily totals

The following aggregates are available depending on the *Display type* (page 147) setting of the Parameter:

Aggregate	VALUE	STATE
NONE (raw)	x	x
INTERPOLATED	x	
AVERAGE	x	
MEDIAN	x	
TOTAL	x	
MIN	x	
MAX	x	
RANGE	x	
DELTA	x	
CHANGE	x	x
START	x	x
END	x	x
COUNT	x	x

---

**Note:** Text and Time Parameters only support *State* aggregates.

---

NONE — No Aggregation is applied. Raw data (as collected by the Source) is returned.

### INTERPOLATED

The Interpolated aggregate either finds or calculates a value at each interval boundary. For each interval boundary timestamp, if there is already good raw data at that timestamp, then the raw data is used. Otherwise, the two closest bounding values must be found, one each in the intervals before and after the boundary. The value at the boundary is then interpolated between these two bounding values. In either case, the raw values involved must be good values to be used. If no good value can be found in the previous or next interval, then the aggregate will have a quality of *No Data*. If any non-good values are skipped in order to find the closest good bounding values, the aggregate quality will be *Data Subnormal*.

## AVERAGE

The average aggregate adds up the values of all good raw data for each interval, and divides the sum by the number of good values producing a mean average. If any non-good values are ignored in the computation, the aggregate quality will be *Data Subnormal*.

## MEDIAN

The median aggregate sorts the good values for each interval and selects the middle value as the median. If any non-good values are ignored in the computation, the aggregate quality will be *Data Subnormal*.

## TOTAL

The total aggregate adds up all the values of all good raw values for each interval. If any non-good values are ignored in the computation, the aggregate quality will be *Data Subnormal*.

## MIN

The minimum aggregate retrieves the minimum good raw value within the interval.

## MAX

The maximum aggregate retrieves the maximum good raw value within the interval.

## RANGE

The range aggregate finds the difference between the raw maximum and raw minimum values in the interval. If only one value exists in the interval, the range is zero. Note that the range is always zero or positive.

## DELTA

The delta aggregate retrieves the difference between the earliest and latest good raw values in an interval. If the last value is less than the first value, the result will be negative. If the last value is the same as the first value, or if the last value is also the first value at the same timestamp, the result will be zero. If the last value is greater than the first value, the result will be positive. If any non-good values exist earlier or later than the earliest and latest good values, respectively, the aggregate quality is *Data Subnormal*.

## CHANGE

The change aggregate retrieves raw values which have been filtered to remove any duplicate sequential values.

## START

The start aggregate retrieves the first raw value within the interval. If the value is non-good, than the quality of the aggregate will be *Data Subnormal*.

## END

The end aggregate retrieves the last raw value within the interval. If the value is non-good, than the quality of the aggregate will be *Data Subnormal*.

## COUNT

This aggregate retrieves a count of all the raw values within an interval. If one or more raw values are non-good, they are not included in the count and the aggregate quality will be *Data Subnormal*.

---

**Note:** Aggregates use the OPC-UA Historical Access v1.0 Specification

---

### 4.3.2 JSON Time Series

**JSON** (<http://json.org>) Time Series (JTS) is a lightweight data-interchange format for *time series* data. The specification is provided with an **MIT license** for free use. It has been designed to be highly readable, parsable and extendable.

The JTS document format is natively supported by [eagle.io](https://eagle.io) (<https://eagle.io>) which can be easily imported and exported via the [HTTP API](#) (page 433).

---

**Note:** The [Time Series](https://github.com/eagle-io/timeseries) (<https://github.com/eagle-io/timeseries>) JavaScript library is provided for free and can easily output time series data in JTS format.

---

Example JTS document:

```
{
  "docType": "jts",
  "version": "1.0",
  "header": {
    "startTime": "2018-08-16T02:00:00.000Z",
    "endTime": "2018-08-16T02:20:43.000Z",
    "recordCount": 5,
    "columns": {
      "0": {
        "id": "541a5a129bc9b4035f906d70",
        "name": "Temperature",
        "dataType": "NUMBER",
        "renderType": "VALUE",
        "format": "0.###",
        "aggregate": "NONE"
      }
    }
  },
  "data": [
    {
      "ts": "2018-08-16T02:00:39.000Z",
      "f": { "0": { "v": 99, "q": 100, "a": "site maintenance" } }
    },
    {
      "ts": "2018-08-16T02:05:40.000Z",
      "f": { "0": { "v": 28.22 } }
    },
    {
      "ts": "2018-08-16T02:10:41.000Z",
      "f": { "0": { "a": "sensor recalibrated" } }
    },
    {
      "ts": "2018-08-16T02:15:42.000Z",
      "f": { "0": { "v": 29.2, "q": 100 } }
    },
    {
      "ts": "2018-08-16T02:20:43.000Z",
```

(continues on next page)

(continued from previous page)

```

    "f": { "0": { "v": 29.18 } }
  }
]
}

```

## JTS Document specification

Attribute	Example	Description
<b>docType</b>	jts	<i>Required.</i> Must be 'jts'
<b>version</b>	1.0	<i>Required.</i> Version '1.0' supported
<b>header</b>		<i>Optional.</i> Object describing document data
<b>data</b>		<i>Required.</i> Array of historic records in chronological order

```

{
  "docType": "jts",
  "version": "<version>",
  "header": {},
  "data": []
}

```

## Header

The header is optional and is used to describe the data contained within the document.

Header Attribute	Example	Description
<b>startTime</b>	2018-08-16T02:00:00.000Z	<a href="#">ISO8601</a> (page 581) timestamp of earliest record in document
<b>endTime</b>	2018-08-16T02:20:43.000Z	<a href="#">ISO8601</a> (page 581) timestamp of latest record in document
<b>recordCount</b>	5	Total record count (number of array items in data)
<b>columns</b>		Object describing columns in document

```

{
  "startTime": "<startTime>",
  "endTime": "<endTime>",
  "recordCount": 0,
  "columns": {}
}

```

## Header Columns

The header *columns* object contains *column index* keys which map to the corresponding *column index* in the record fields object. Column index assignment is generally zero-based (first index is 0).

Column Attribute	Example	Description
<b>id</b>	541a5a129bc9b40354906d70	Unique identifier of resource (usually its <code>_id</code> )
<b>name</b>	Temperature	Name of resource or node
<b>dataType</b>	NUMBER	<i>Data type</i> (page 575) of associated node. <i>[NUMBER, TEXT, TIME, COORDINATES]</i> Note: if ‘renderType’ is STATE the data in this document will be String.
<b>render-Type</b>	VALUE	Rendering of ‘v’ attribute in records. <i>[VALUE, STATE]</i>
<b>format</b>	0.###	<i>Format</i> (page 147) of the value.
<b>aggregate</b>	NONE	Historic <i>aggregate</i> (page 569) applied to data. Default is NONE
<b>base-Time</b>	D	<i>OPC Base Time</i> (page 585) used for aggregation
<b>interval</b>	1H	<i>OPC Interval</i> (page 585) used for aggregation

```
{
  "0": {
    "id": "<id>",
    "name": "<name>",
    "dataType": "<dataType>",
    "format": "<format>",
    "aggregate": "<aggregate>",
    "baseTime": "<baseTime>",
    "interval": "<interval>"
  },
  "1": {},
  "2": {}
}
```

## Data

The data attribute contains an array of records. Each record contains a ‘ts’ *ISO8601* (page 581) timestamp and an ‘f’ fields object which can contain value, quality and annotation attributes for one or more columns (using column index as key). The records should be sorted chronologically by timestamp.

Record Attribute	Example	Description
<b>ts</b>	2018-08-16T02:00:00.000Z	<i>Required.</i> <a href="#">ISO8601</a> (page 581) timestamp of data point(s)
<b>f</b>		<i>Required.</i> Object containing data for 1 or more columns (using column index as key)
<b>v</b>	10.4	<i>Optional.</i> Value of column for the corresponding record timestamp <a href="#">Data type</a> (page 575) should match <i>dataType</i> option in header
<b>q</b>	100	<i>Optional.</i> <a href="#">Quality</a> (page 577) code associated with data value for this column
<b>a</b>	site maintenance	<i>Optional.</i> <a href="#">Annotation</a> (page 578) text associated with data point

```
[
  {
    "ts": "<ts>",
    "f": {
      "0": {"v": 10.4, "q": 100, "a": "site maintenance"},
      "1": {"v": 55}
      "2": {"a": "sensor recalibrated"}
    }
  },
  {
    "ts": "<ts>",
    "f": { "0": {"v": 12, "q": 100}, "1": {"v": 55, "q": 100} }
  }
]
```

**Note:** The ‘f’ fields object can be sparsely populated by only including columns that have data for the associated record timestamp.

## Data Types

JTS supports both simple and complex data types. Complex data types are always encapsulated in a JSON object with its key prefixed with ‘\$’.

When inserting historic data, the data type *MUST* match the parameter type. For example, TIME data is only accepted on Time parameters. Data that has been exported using renderType ‘STATE’ will always have a *String* format.

The following data types are supported in the ‘v’ value field of a record column:

Data Type	Accepted Parameter Type	Format	Example
<b>NUMBER</b>	Number parameters	Number	24.5
<b>TEXT</b>	Text parameters	String	“valve open”
<b>TIME</b>	Time parameters	{“\$time”: <i>“ISO8601 (page 581)”</i> }	{“\$time”: “2014-08-16T02:00:00Z”}
<b>COORDINATES</b>	Locations	{“\$coords”: [<latitude>, <longitude>]}	{“\$coords”: [-33.86785, 151.20732]}

---

**Note:** The record ‘ts’ field can be expressed as a simple *ISO8601* (page 581) timestamp instead of the complex TIME data type.

---



### 4.3.3 Quality

☆ Historic data can optionally contain quality information. Each data point in the time series can store a quality code that represents the quality of the value captured. System quality codes may also be applied automatically when displaying or exporting the historic data (usually when aggregate calculations are applied).

The account settings [Quality Codes](#) (page 266) section allows Account Owners and Administrators to add custom quality codes or modify the built-in system quality codes.

Quality codes are assigned a Type which represents the validity of the data: GOOD, BAD or UNCERTAIN.

#### System Qualities

System qualities can have their source and quality codes re-assigned but they can not be removed.

**Data Subnormal [149]** Non-Good values have been excluded in the aggregate calculation so the quality of the data is uncertain.

**Data Unavailable [158]** There is no data available for this period and has been specifically indicated as a gap by the user.

**No Bounds [156]** Bounding values determine the start and end points when requesting raw data. If a raw data value exists at the start or end point, it is considered the bounding value. If no raw data value exists at the start or end point, the next data point outside of the range is considered the bounding value and the quality will be shown as No Bounds.

**No Data [155]** There is no data available for the aggregation calculation period after non-good values have been excluded.

**No Good Data [165]** There is no data available for the aggregate calculation period.

**Entry Inserted [162]** Data has been inserted (e.g. via interpolation)

#### 4.3.4 Annotations



Historic data can optionally contain annotations. Each data point in the time series can store a textual comment to provide users with additional context to the value or quality.

Annotations are visible on charts and tabular data and can be included in historic exports.

---

**Note:** The maximum length of an annotation is 500 characters.

---

### 4.3.5 Limits

Eagle.io enforces some basic historic data limits to ensure the platform remains performant and provides the best user experience.

---

**Note:** If you believe your application may exceed our historic limits, please contact us to discuss your requirements.

---

### Acquisition and Storage

Eagle.io supports the acquisition and storage of up to **20000 records per [Data Source](#) (page 134) per day**. Exceeding the limit will trigger an *Overload Alarm* (page 173) on the Source.

Daily intervals are defined in the UTC time zone. The acquisition limit applies to the number of records acquired in one UTC day, regardless of the record timestamps. The storage limit applies to the number of records that have timestamps occurring on the same UTC day, regardless of when they are acquired.

The limit of 20000 records is sufficient for Data Sources that generate data continuously at 5-second intervals. While eagle.io supports smaller data intervals (down to 1 millisecond), such resolutions cannot be generated continuously. For example, if a Data Source generated data at 1-second intervals, the limit of 20000 records would be reached in under 6 hours.

Under some circumstances, a Data Source will be exempt from the acquisition limit (but will always be subject to the storage limit of 20000 records per day):

1. Data Sources that have never been acquired before (because they have just been created)
2. Data Sources that do not have current data (defined as a record that occurs on the current day in the UTC time zone)
3. Data Sources that have had all their historical records deleted since the last acquisition

This exemption from the acquisition limit allows a large number of historical records to be backfilled at one time; however, each individual day of data must still not have more than 20000 records. For example, when connecting a new data logger that has been logging data at 5-second intervals for 1 month, all the data on the device can be acquired by eagle.io in a single attempt, even though this is over 500000 records. Note that each individual day of data still does not contain more than 20000 records, and so does not exceed the storage limit.

The acquisition limit and storage limit have different effects, highlighted by the following examples.

### Acquisition limit example

Consider a data set spanning 1 week at 5-second resolution, i.e. 17280 records per day, for a total of 120960 records. If you tried to acquire all these records in one attempt, you would hit the daily limit of 20000 records acquired per day, even though each individual day of records is under the storage limit. The following day, you would be able to acquire a further 20000 records. In total, you would need to perform daily acquisitions for a week, adding only 20000 records per day, to acquire the entire data set. However, if

any of the previously listed exemptions applied to the Data Source, you would be able to acquire all 120960 records in a single attempt.

### Storage limit example

Consider a data set spanning 1 day at 1-second resolution, i.e. a total of 86400 records with timestamps all occurring on the same day. If you only acquired 1000 of these records each day, you would not hit the daily acquisition limit. However, after 20 days, you would hit the storage limit of 20000 records with timestamps occurring on the same day. You would then be permanently restricted from storing any more records with timestamps occurring on that day. There are no exemptions to the storage limit, so the only way to store additional records on that day would be to delete some or all of the existing records.

### Data editing

Eagle.io limits the number of raw records that are loaded on a *historic chart* (page 201) to **50000 records per parameter while editing**. If your time range selection exceeds the record limit a notice will be shown in the top-left corner of the chart with an option to temporarily ignore the warning and load the full dataset (please be aware this may result in poor browser performance and stability issues).

## 4.4 Time

When working with *time series* data from multiple geographic locations its important to understand how timezones and date formatting affects the display of the data.

### 4.4.1 Timezones

Timezones allow for data to be collected from various geographical locations and stored on the server in *UTC*. Refer to the Source configuration docs for information on assigning a timezone.

Users can assign a timezone to their User profile. All data displayed to the user including current, historic, notifications and events will be displayed in the selected timezone.

Example:

```
(GMT-08:00) Pacific Time
```

The *HTTP API* (page 433) requires all timezones to be specified using the *TZ* naming convention as specified in [this list](http://en.wikipedia.org/wiki/List_of_tz_database_time_zones) ([http://en.wikipedia.org/wiki/List\\_of\\_tz\\_database\\_time\\_zones](http://en.wikipedia.org/wiki/List_of_tz_database_time_zones)). *Etc/UTC* should be used for UTC.

Example:

```
America/Los_Angeles
```

### 4.4.2 Timestamps

*Timestamps* represent the date and time of an event and can be represented in different formats.

#### ISO8601

The *HTTP API* (page 433) requires all timestamps to use the *ISO8601* ([http://en.wikipedia.org/wiki/ISO\\_8601](http://en.wikipedia.org/wiki/ISO_8601)) standard format for date and time display. Timestamps in this format can be in UTC with the 'Z' (Zulu) zone designator applied or include a '+' offset from UTC. Timestamps can optionally specify fractional seconds using 3 decimal places.

Example

```
2014-10-09T22:38:10Z
2014-10-09T22:38:10.000Z

2014-10-09T20:38:10+0200
2014-10-09T20:38:10+02:00
```

## Customize time format

Users can assign a preset or custom Time format to their User profile using the time format tokens below. All timestamps displayed to the user will be in their preferred format (where applicable).

Default format

```
YYYY-MM-DD HH:mm:ss
```

Example output

```
2021-06-01 22:38:00
```

ISO8601 timestamp format

```
YYYY-MM-DDTHH:mm:ssZ
```

Example output

```
2021-06-01T22:38:00+1000
```

## Time format tokens

Time Unit	Token	Output
<b>Month</b>	M	1 2 ... 11 12
	Mo <sup>1</sup>	1st 2nd ... 11th 12th
	MM	01 02 ... 11 12
	MMM	Jan Feb ... Nov Dec
	MMMM	January February ... November December
<b>Day of Month</b>	D	1 2 ... 30 31
	Do <sup>1</sup>	1st 2nd ... 30th 31st
	DD	01 02 ... 30 31
<b>Day of Year</b>	DDD	1 2 ... 364 365
	DDDo	1st 2nd ... 364th 365th
	DDDD	001 002 ... 364 365
<b>Day of Week</b>	d <sup>1</sup>	0 1 ... 5 6
	do <sup>1</sup>	0th 1st ... 5th 6th
	dd	Su Mo ... Fr Sa
	ddd	Sun Mon ... Fri Sat
	dddd	Sunday Monday ... Friday Sat- urday
<b>Day of Week (Locale)</b>	e <sup>1</sup>	0 1 ... 5 6
<b>Day of Week (ISO)</b>	E	1 2 ... 6 7
<b>Week of Year</b>	w	12 .. 52 53
	wo <sup>1</sup>	1st 2nd ... 52nd 53rd
	ww	01 02 ... 52 53
<b>Week of Year (ISO)</b>	W	1 2 ... 52 53
	Wo <sup>1</sup>	1st 2nd ... 52nd 53rd
	WW	01 02 ... 52 53
<b>Year</b>	YY	70 71 ... 29 30
	YYYY	1970 1971 ... 2029 2030
<b>Week Year</b>	gg	70 71 ... 29 30
	gggg	1970 1971 ... 2029 2030
<b>Week Year (ISO)</b>	GG	70 71 ... 29 30
	GGGG	1970 1971 ... 2029 2030
<b>AM/PM</b>	A	AM PM
	a	am pm
<b>Hour</b>	H	0 1 ... 22 23
	HH	00 01 ... 22 23
	h	1 2 ... 11 12
	hh	01 02 ... 11 12
<b>Minute</b>	m	0 1 ... 58 59
	mm	00 01 ... 58 59
<b>Second</b>	s	0 1 ... 58 59
	ss	00 01 ... 58 59

Continued on next page

Table 1 – continued from previous page

Time Unit	Token	Output
<b>Fractional Second</b>	S	0 1 ... 8 9
	SS	0 1 ... 98 99
	SSS	0 1 ... 998 999
<b>Timezone</b>	Z	-07:00 -06:00 ... +06:00 +07:00
	ZZ	-0700 -0600 ... +0600 +0700
<b>Unix Timestamp</b>	X	1360013296
<b>Unix MS Timestamp</b>	x	1360013296000

---

<sup>1</sup> Tokens not available for Delimited Text Source.



### 4.4.3 Relative Time

OPC Relative Time is used to represent date and time in an easy to read format that allows users to navigate or select a period of time without explicitly specifying a date. An example use is NOW-5M which simply translates to the current time minus 5 minutes. All times are relative to the user profile timezone.

OPC Relative Time syntax:

```
<basetime> [ ( + | - ) <interval> ]
```

**Base Time** A time that is relative to ‘now’, the present time. You can enter a base time in either short or long format.

**Interval** An optional further offset that you can either add or subtract from the base time. You specify this offset in the OPC Time Format.

Example usage:

```
H (start of the current hour)
D+9H (9am current day)
NOW-7D+6H (6 days and 18hours ago)
```

### OPC Base Time and Interval Keywords

Keyword Long (short)	Base Time	Interval
NOW	The current time	N/A
FIRST	Earliest data value time	N/A
LAST	Most recent data value time	N/A
SECOND (S)	The start of the current second	Offset from time in seconds
MINUTE (M)	The start of the current minute	Offset from time in minutes
HOUR (H)	The start of the current hour	Offset from time in hours
DAY (D)	The start of the current day	Offset from time in days
WEEK (W)	The start of the current week	Offset from time in weeks
MONTH (MO)	The start of the current month	Offset from time in months
YEAR (Y)	The start of the current year	Offset from time in years

**Note:** You can specify both the <basetime> and <interval> in either short or long format.



---

## Frequently Asked Questions

---

### 5.1 General

#### 5.1.1 What browsers do you support?

We support all modern browsers: Chrome, Edge, Safari, Firefox and Opera.

Please ensure you have both **Cookies** and **Javascript enabled** and are using the **latest version** of your preferred browser.



We recommend [Google Chrome](https://www.google.com/intl/en/chrome/browser/) (<https://www.google.com/intl/en/chrome/browser/>) for the best [eagle.io](https://eagle.io) (<https://eagle.io>) experience.

#### 5.1.2 How do I access an existing users data?

Account access is managed by the owner of the account and any nominated account administrators. Please request access directly from the appropriate persons in charge of the account.

New users will receive an email invitation with a link to enter their details and access the site.

#### 5.1.3 What languages do you support?

English is currently the only supported language. We recommend you use the [Google Translate extension for Chrome](https://chrome.google.com/webstore/search/google%20translate%20extension?hl=en) (<https://chrome.google.com/webstore/search/google%20translate%20extension?hl=en>) for other languages. Users can set their own locale and timezone display preferences.

### 5.1.4 What types of charts do you support?

We offer a variety of highly customizable interactive charts including *Historic* (page 67), *Windrose* (page 74) and *Displacement* (page 81) charts. Check out our *charts documentation* (page 66) for full usage details.

### 5.1.5 How much historic data can I store?

We support the acquisition and storage of up to **20000 records per Data Source** (page 134) **per day**. Refer to Historic Data *Limits* (page 579) for more information about storage limits.

### 5.1.6 Where is my data stored and is it secure?

We store all customer data securely on Amazon Web Services data hosting facilities located in Australia. Data is strictly segregated so that no user may ever view, tamper with, or become aware of the data of any other user. Refer to our *Compliance* (<https://eagle.io/policies/compliance>) section for full details.

### 5.1.7 Do you have a service level guarantee?

We provide a full service level guarantee (99.9% up-time commitment) to clients subscribed to our Standard (or larger) pricing plans. Refer to our *SLA* (<https://eagle.io/policies/sla>) for full details.

## 5.2 Data Loggers

### 5.2.1 What types of data loggers, PLCs and RTUs are supported?

We support the following range of data loggers:

**Campbell Scientific** Campbell PakBus loggers including CR200, CR300, CR800, CR1000, CR3000, CR9000, CRVW & CR6 series.



**CellVisor** E State Automation CellVisor 3000/6000 series.



**dataTaker** Thermo Fisher Scientific DT80 range.



**Generic** Virtually any data logger that can *Email* or *FTP* data in CSV format.



**Libelium** Waspote and Plug & Sense! range with support for *Direct Connect* or *Meshlium*



### Worldsensing Loadsensing



### OTT netDL OTT netDL 500 and 1000.



**Point Orange 3G** Metasphere Point Orange 3G RTU.



**Senquip** ORB-X1.



**Spider** Halytech Spider SMS and microSpider range.



**YDOC** YDOC ML-315 range of data loggers.



**WaterLog** WaterLog Storm3 series.



Support for a wide range of additional data loggers and devices is coming soon. [Contact Us](https://eagle.io/contact/) (<https://eagle.io/contact/>) to find out how to get support for your device.

## 5.2.2 What connection options are supported?

You can use a variety of transport options to connect your data logger to [eagle.io](https://eagle.io) (<https://eagle.io>) including:

**TCP Client** Connect to any device with a Public IP Address or Domain name (including DDNS).

**TCP Server** WaterLog, SensorLog, Waspnotes and other devices can establish a direct connection to the cloud to stream data on schedule or event-based.

**TCP Callback** Allow any remote device to establish an IP connection to [eagle.io](https://eagle.io) using a special identification number. You can use your modem or data logger to send the Id on initial connection, or include a unique identifier in the data stream.

**Scout** The *Scout modem* is the easiest way to connect your data logger to [eagle.io](https://eagle.io). The Scout takes any standard internet ready SIM Card without the need for fixed IP addresses and has a range of smart features including inbuilt GPS, signal strength monitoring, event based reporting and low power mode.

**Meshlium** Meshlium devices provide efficient routing of data collected from Waspnote devices in large Libelium sensor networks.



**FTP/SFTP, Email, MQTT** Send your data to the cloud via FTP/SFTP, Email or MQTT.

### 5.2.3 What level of functionality do you provide for connected data loggers?

- Upload programs and firmware (including network wide updates in one click)
- Perform Control operations (setting of public table variables in Campbell Scientific data loggers)
- Create/update/merge parameters automatically as programs change
- Synchronize device clock with timezone and daylight savings support

### 5.2.4 Can I connect to my data logger using 3rd party software?

Yes! You can enable *Remote Access* to your data logger at any time. An Internet address and Port will be allocated to allow you to immediately connect via TCP/IP. You can specify a session timeout and restrict access based on IP Address or allow connections from anywhere. Want to use LoggerNet or other tools for your low level operations - no problem. See [Data Source](#) (page 134) configuration for full details.

## 5.3 Files

### 5.3.1 What types of time series data files do you support?

We support most types of delimited text files such as csv, loggernet and other standard formats. See [Data Source](#) (page 114) configuration for full details.

### 5.3.2 What file transports do you support for automatic collection?

We support automatic collection via the following transports:

- Download from FTP site
- Download from Web site
- Email to eagle.io
- Publish using MQTT
- Read from [Amazon S3](https://aws.amazon.com/s3) (<https://aws.amazon.com/s3>)
- Read from [Dropbox](http://www.dropbox.com) (<http://www.dropbox.com>)
- Upload to ftp.eagle.io
- Upload to sftp.eagle.io



You can also manually import and export data files directly from the user interface or use our [HTTP API](#) (page 433) for automation.

### 5.3.3 Can I upload other types of files?

Files of any type can be uploaded through the interface as an [Attachment](#) (page 105) or use an [Attachment Source](#) (page 109) for automated uploads. All content and files uploaded must comply with our [Terms of Use](#) (<https://eagle.io/policies/terms>).

## 5.4 Cameras

### 5.4.1 Do you support cameras sending images or video files to eagle.io?

Yes, you can create an [Attachment Source](#) (page 109) which accepts files of any type. The files are stored as Attachments and you can set your Locations to display the most recent images uploaded. Images and video can be reviewed in the media gallery or downloaded for offline viewing.

We support Email and FTP as the Attachment Source transport. To send via Email, your camera must have in-built support for emailing images/files with custom subject text. For uploads via FTP/SFTP you can simply enter the credentials provided on Source creation.

Note - live streaming video is not directly supported.

**data logger** A data logger is an electronic device that records data over time or in relation to location either with a built in instrument or sensor or via external instruments and sensors.

**delimiter** A delimiter is a sequence of one or more characters used to specify the boundary between separate, independent regions in plain text or other data streams. An example of a delimiter is the comma character, which acts as a field delimiter in a sequence of comma-separated values.

**time series** A time series is a sequence of data points, measured typically at successive points in time spaced at uniform time intervals or at irregular intervals (event based).

**timestamp** A timestamp is a sequence of characters or encoded information identifying when a certain event occurred, usually giving date and time of day.

**UTC** Coordinated Universal Time is the primary time standard by which the world regulates clocks and time. UTC is synonymous with GMT.



## D

data logger, [595](#)  
delimiter, [595](#)

## T

time series, [595](#)  
timestamp, [595](#)

## U

UTC, [595](#)