



Advanced Sandbox Walk-through Guide

Table of Contents

Overview.....	1
Component Studio	2
Explore Auto Registration and Notifications.....	3
Examples.....	4
Creating a Dashboard – Monitor Refrigerator Current	4
Creating a Dashboard – Monitor RSS Feeds	9
Defining an Expression Derived Stream.....	12
Defining an RSS Derived Stream.....	14
Creating an Event with Notifications.....	20

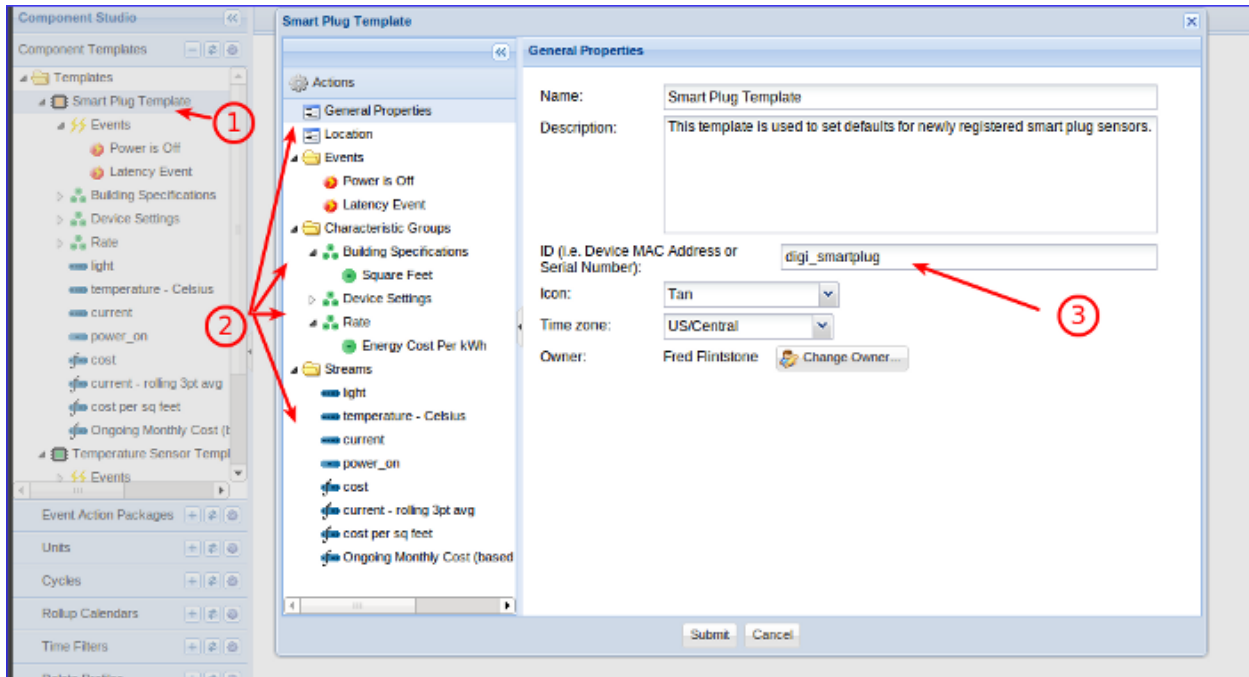
Overview

This document will walk you through creating a dashboard, a derived stream and adding an event to a component.

Modeling items in GroveStreams can be avoided by using GroveStreams Blueprints. More information about GroveStreams blueprints can be found at the [Help Center](#).

Component Studio

Next we will explore “Component Studio”. Click on the **Component Studio** button at the top of Observation Studio.

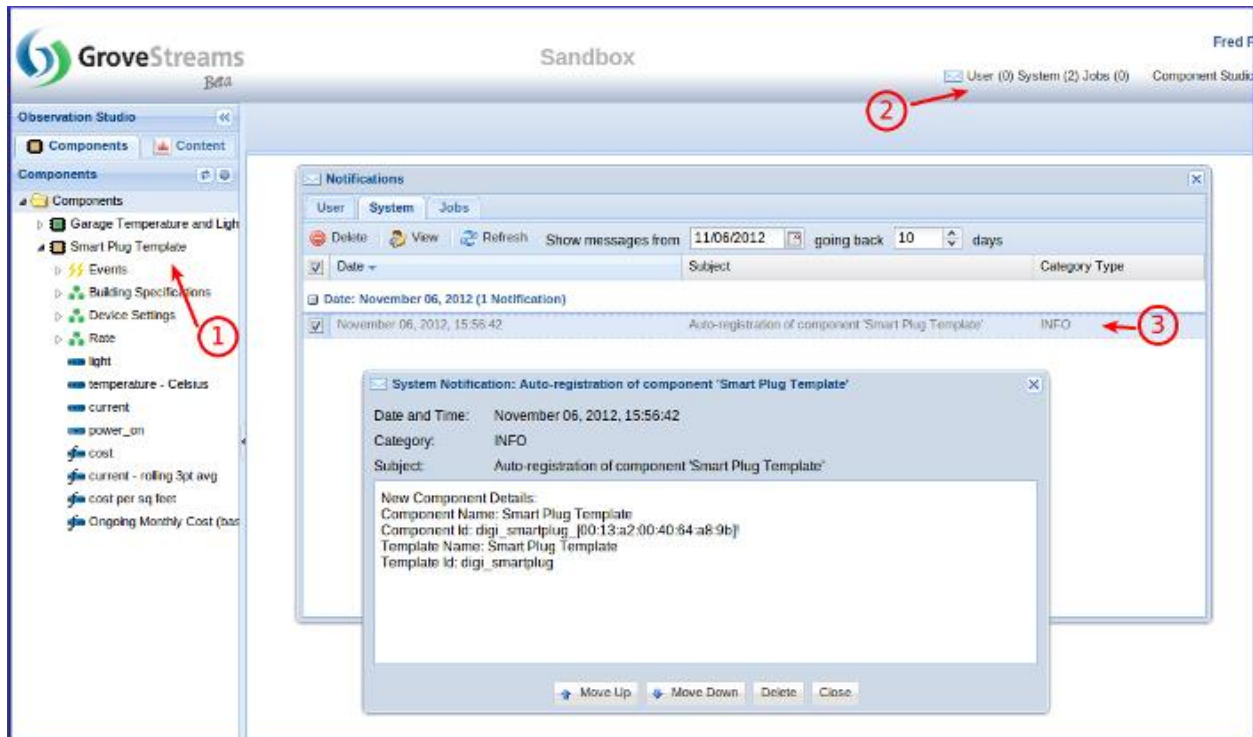


Component Studio is where most modeling takes place with the exception of component definitions and dashboards. Part of Component Studio is the modeling of Component Templates. Components can be created from Component Templates. Templates are useful if there are a lot similar components registering with GroveStreams. Templates avoid the requirement to model each component separately.

Let's look at the Smart Plug Component Template:

1. Right click on “Smart Plug” and choose **Edit Component Template**
2. Select each object in the left pane that makes up a Component Template to see how it is modeled.
3. A component feed can include the Template ID in its feed JSON body. The component will automatically be created during a feed upload based on the template ID in the Feed JSON body if the component does not already exist. The sandbox feeds include component template IDs so that the sensor can automatically register itself and be created from this template during a feed upload.
4. Close the template edit window by clicking on the **Cancel** button.

Explore Auto Registration and Notifications



A GroveStreams sandbox organization will reregister components if they are deleted.

The next steps will make current dashboards that are referencing the smart plug sensor invalid. Do not perform the next steps if you want to preserve your existing smart plug dashboards.

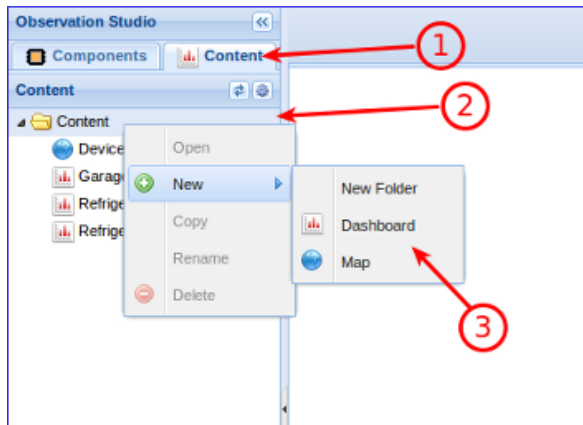
To demonstrate auto-registration of a component that is based on a component template:

1. Right click on the Refrigerator component and choose **Delete** and answer **Yes**.
2. You will receive a notification whenever a new component auto-registers. Wait a few seconds until you see a new System notification appear by watching for the number of unread System notifications to increase on the notifications button located at the top of the studio. Click the notifications button when this happens.
3. Double click the notification to view it. Select the Components folder located on the left and then select the **Refresh** button located above it. The newly registered component will appear in the **Components** tab under the **Components** folder. You can now create new folders and drag your component(s) to any folder. Note that when you delete a component, all of its data, including characteristic and stream data, is deleted and the properties of the newly registered component will match the properties of the template it was created from, including its name. You can edit the component to change the name and other properties once it has registered. Also, default component, stream and many of their properties can be passed with a Feed such as the component name and fixed location information. The sandbox feed does not pass any default information, only the componentTemplateID.

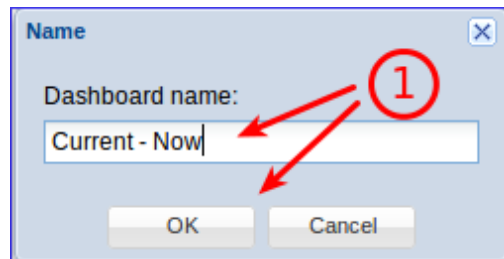
Examples

Creating a Dashboard – Monitor Refrigerator Current

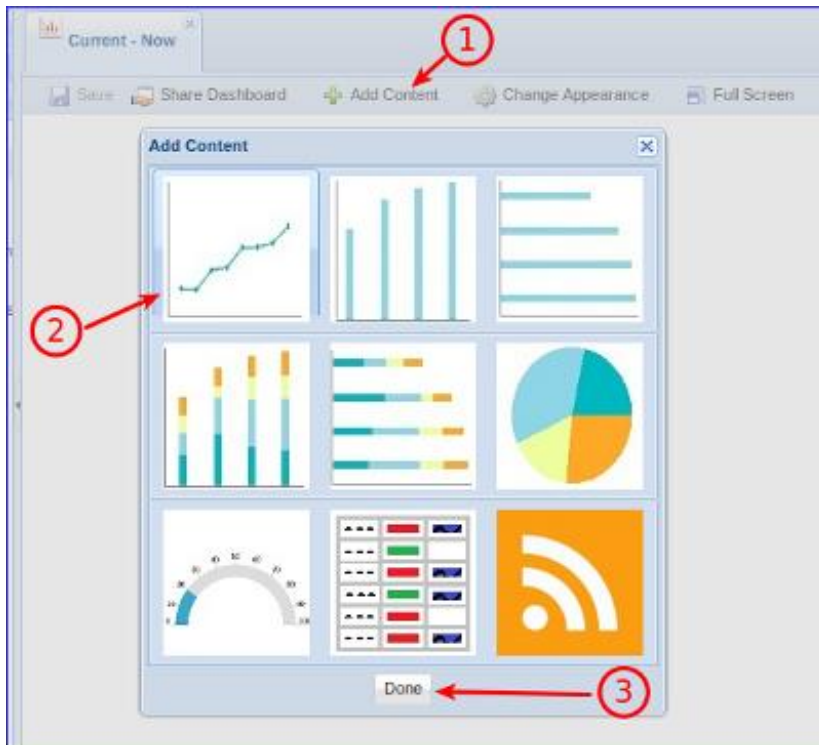
Dashboards allow many different streams to be viewed from many different components. This section will walk you through creating a new dashboard.



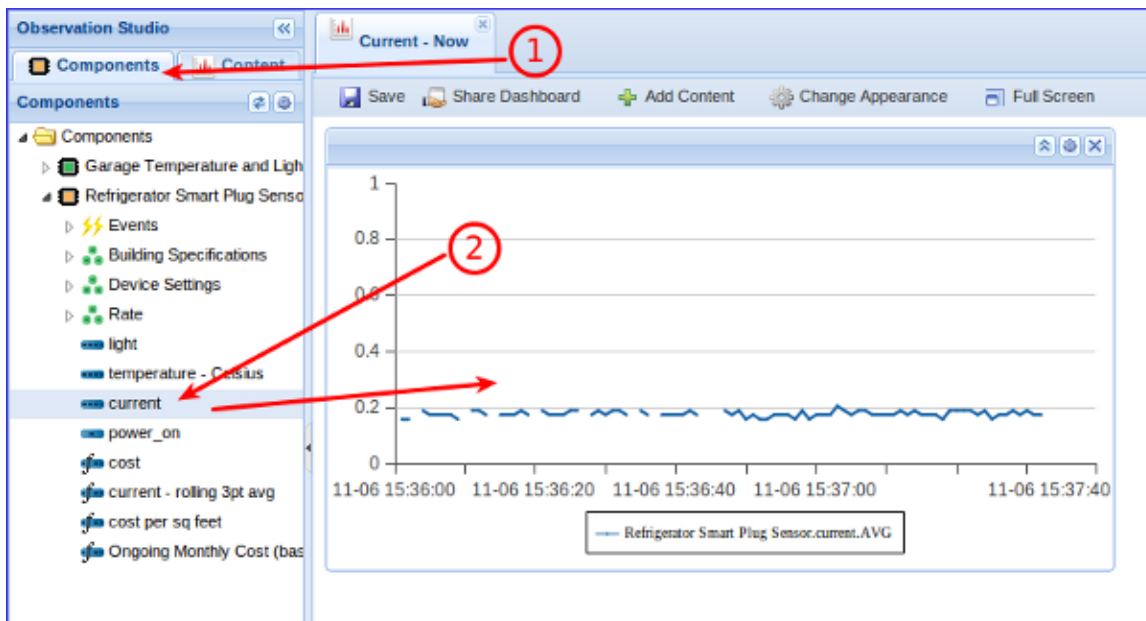
1. Choose the **Dashboards** tab.
2. Expand the **Dashboards** folder.
3. Right click on the **Dashboards** folder and choose **New – Dashboard**.



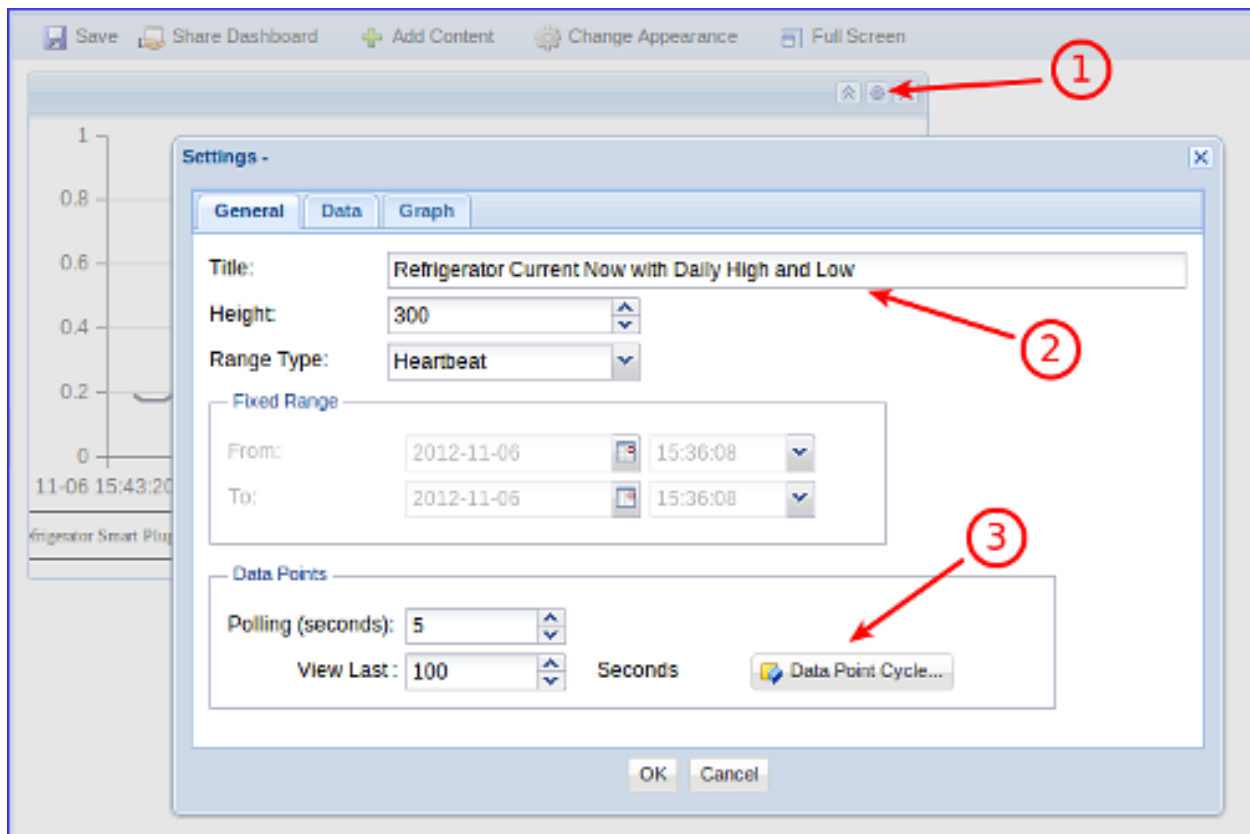
- a.
- b. Give your new dashboard a name and click **OK**.



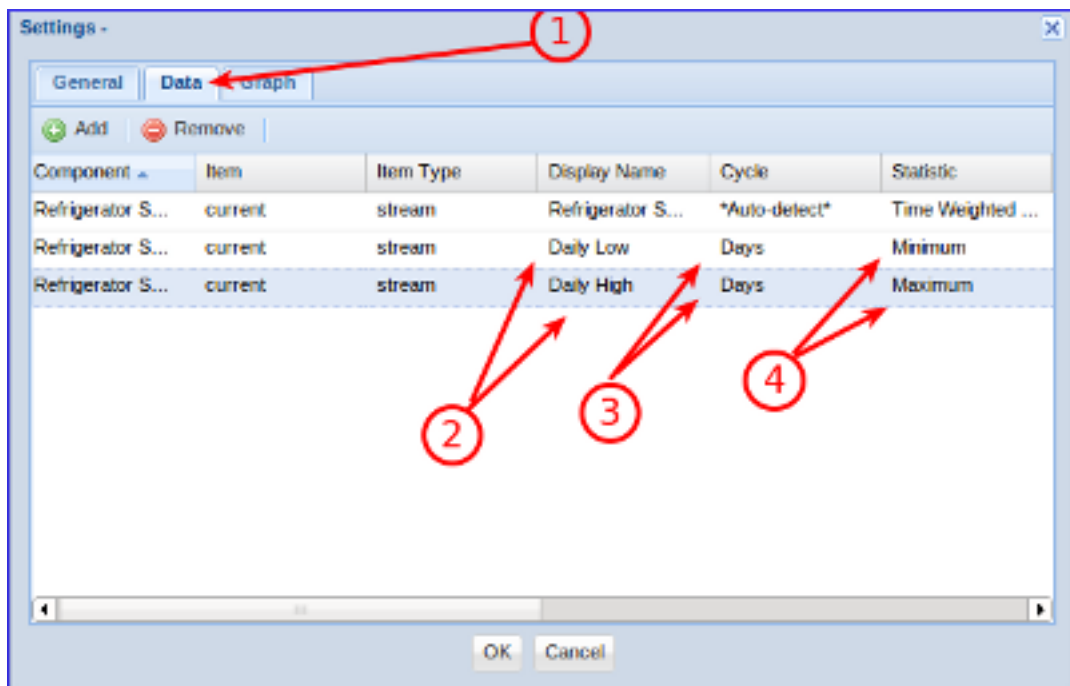
1. Click the **Add Content** button.
2. Click one time on the line chart.
3. Select **Done**.



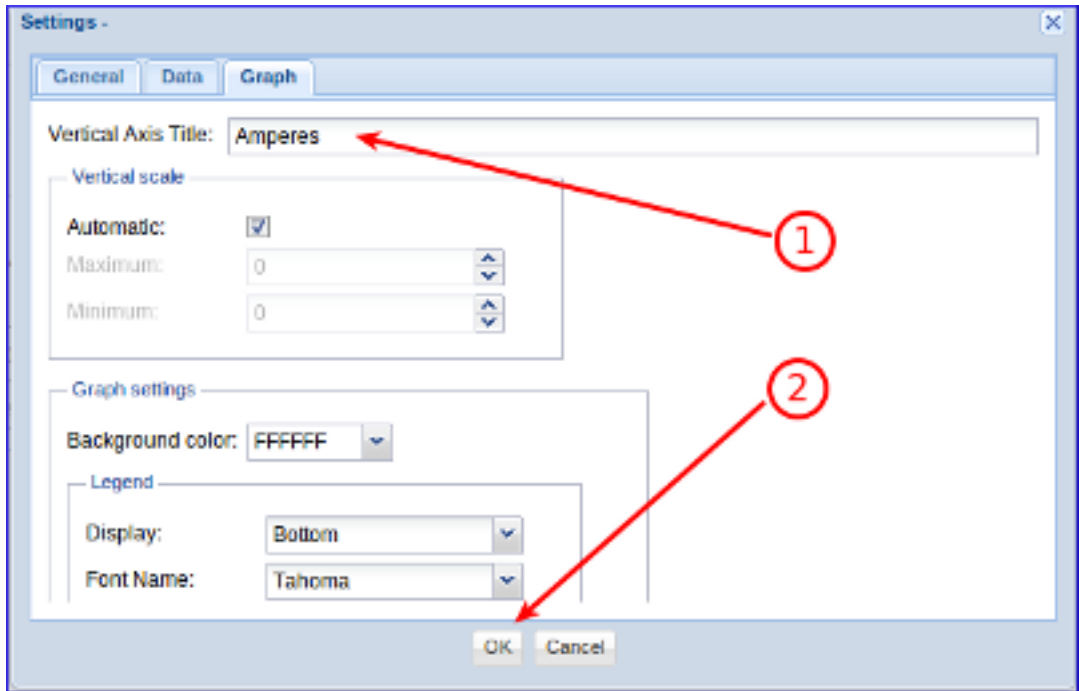
1. Select the **Components** tab.
2. Drag the “current” stream onto your line chart widget **three times**. Yes, three times! We will be graphing three separate current streams below. You should now see current being graphed.



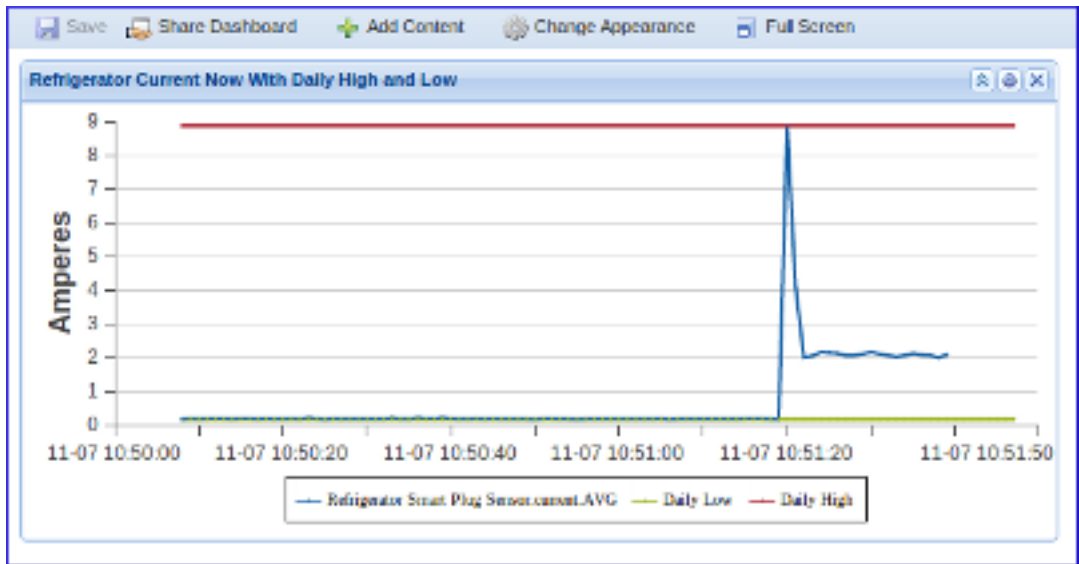
1. Click on the widget's cog button, located at the top of the widget, to edit its settings.
2. Enter a name for your dashboard widget. It can be any name.
3. Select **Seconds** for your Data Point Cycle. This widget will display the last 100 seconds for the selected streams.



1. Select the **Data** tab. You should see current listed three times, once for each time you dragged it onto the chart widget. Instead of dragging and dropping a stream, you could have clicked the Add button located on this view to add streams to the widget.
2. We want the chart to display current and the daily high and lows for the current. Change the display names to match what's in the view above.
3. Set the Cycle to **Days** for the last two data streams.
4. Set the statistic to **Minimum** and **Maximum** for the last two data streams.



1. Select the **Graph** tab and enter “Amperes” for your axis title.
2. Click **OK**.



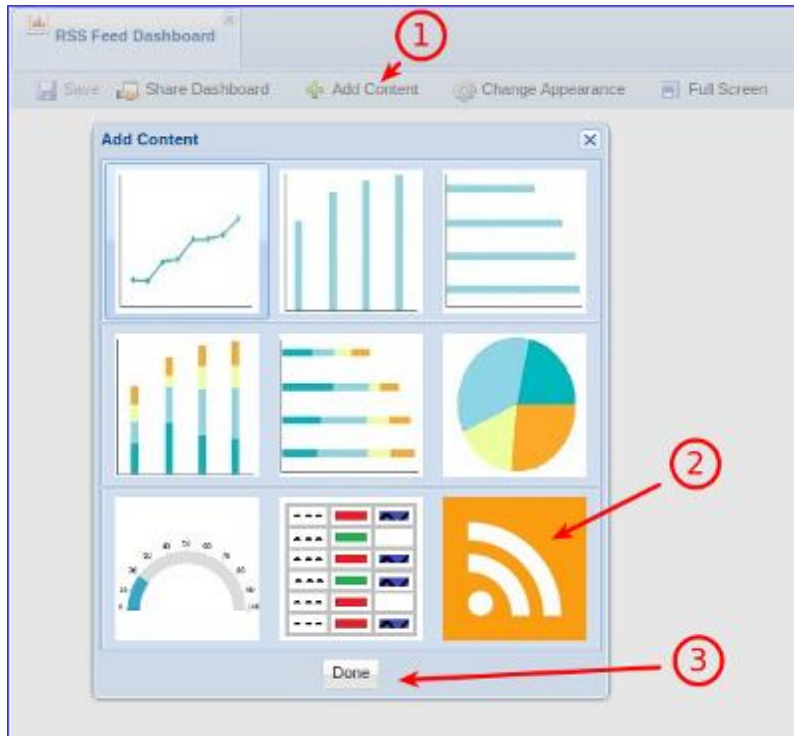
You should see a chart similar to the above image.

Congratulations! You’ve created a dashboard that shows one second actuals with daily highs and lows in the same chart! Click the save button to save your changes.

Play with the widget settings to learn more about how GroveStreams dashboards give you access to near real-time statistics for many different virtual streams (rollup cycle functions).

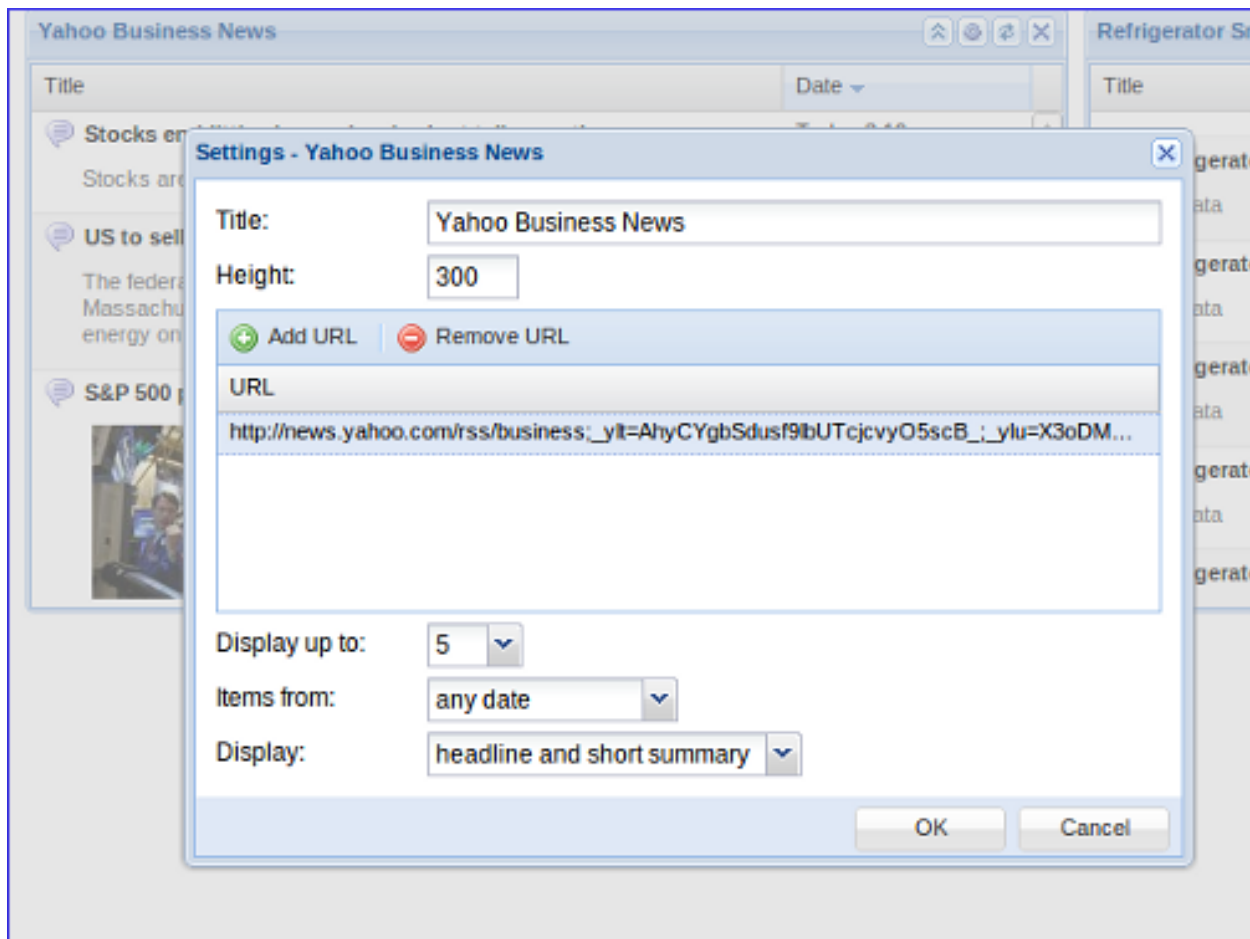
Creating a Dashboard – Monitor RSS Feeds

This section will walk you through creating a dashboard that monitors RSS feeds

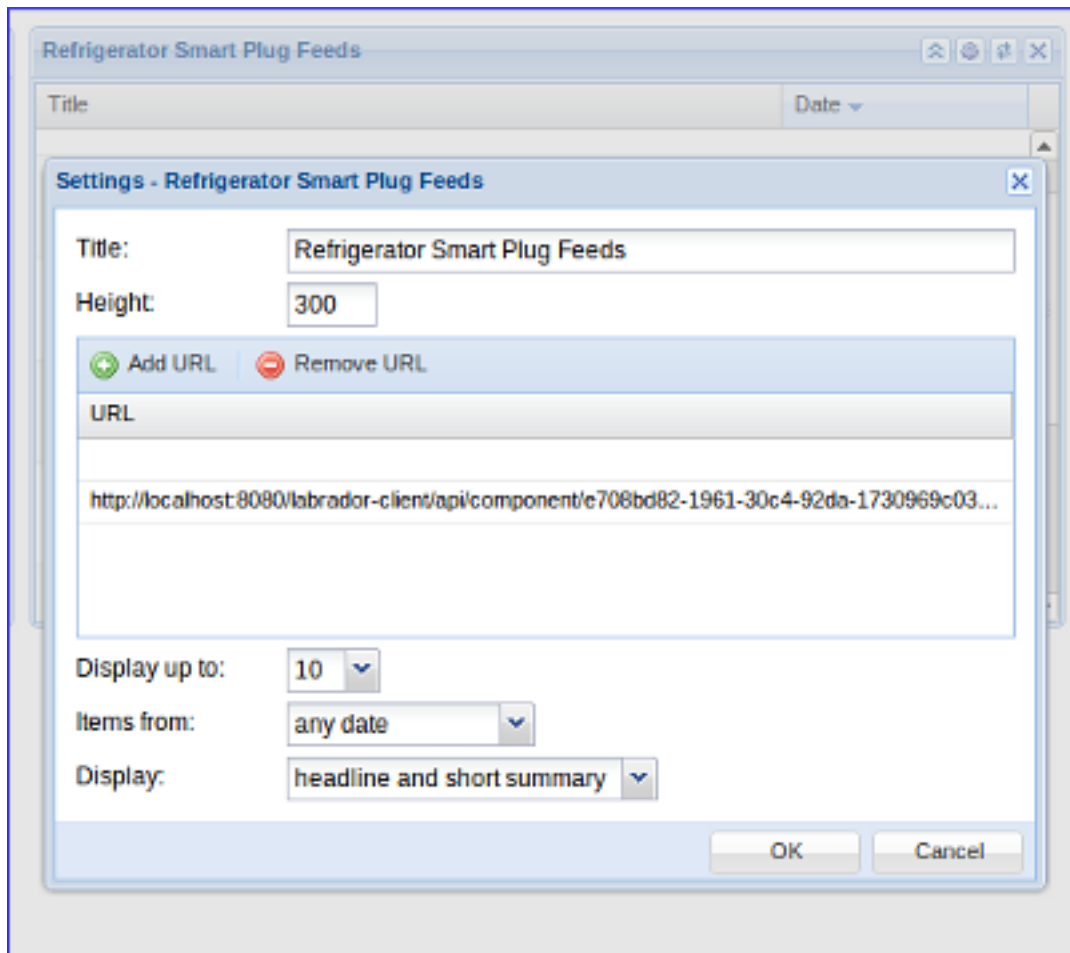


Click on the **Dashboards** tab within Observation Studio. Right click on the **Dashboards** folder and create a new dashboard.

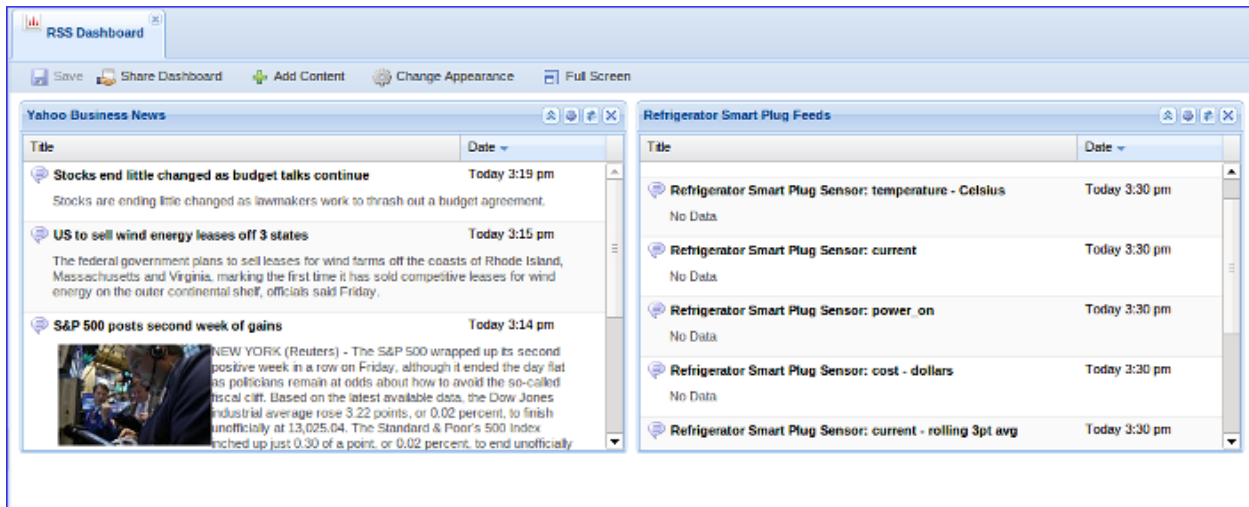
1. Click **Add Content** within the new dashboard.
2. Click the RSS Feed widget **two** times so that two dashboard widgets are added.



1. Edit the first widget's settings (click the cog icon in its title bar)
2. Set similar settings to the image above. Set the URL to Yahoo's Business RSS Feed which can be found [here](#):
 - a. http://news.yahoo.com/rss/business;_ylt=Ase6xNYRmudcaQJIEUJ76se5scB;_ylu=X3oDMTFtdDBydWjKBG1pdANSU1MgU2l0ZUluZGV4IEJ1c2luZXNzBHBvcwMOBHNIYwNNZWRpYVJTU0VkaXRvcmlhbaA--;_ylg=X3oDMTFIamZvM2ZlBGluGwDdXMEbGFuZwNlbi11cwRwc3RhaWQDBHBzdGNhdAMEchQDc2VjdGlbnM-;_ylv=3
3. Click **OK**



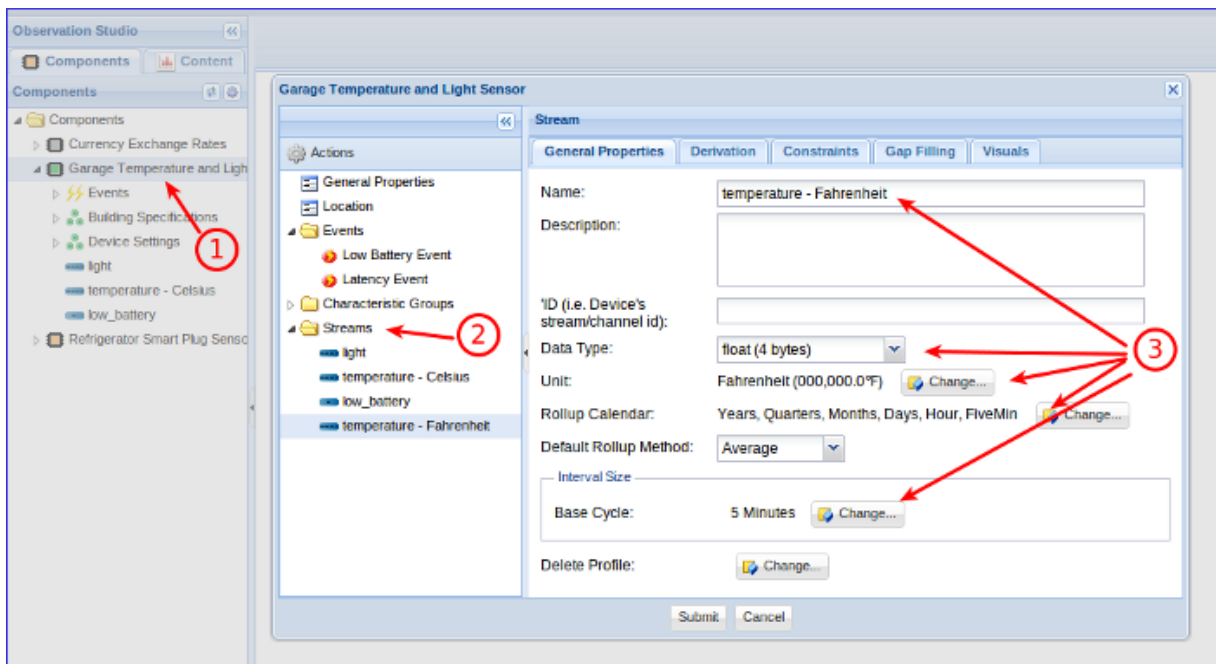
1. Drag the Refrigerator Smart Plug Sensor onto your second widget. This will automatically add its GroveStreams RSS feed to the widget
2. Edit the widget settings and set the same properties as in the image above (except for the URL since that was set when you dragged the component onto the widget)
3. Click **OK**



Congratulations! You have created a dashboard that displays RSS feeds from outside of GroveStreams and from within your GroveStreams organization.

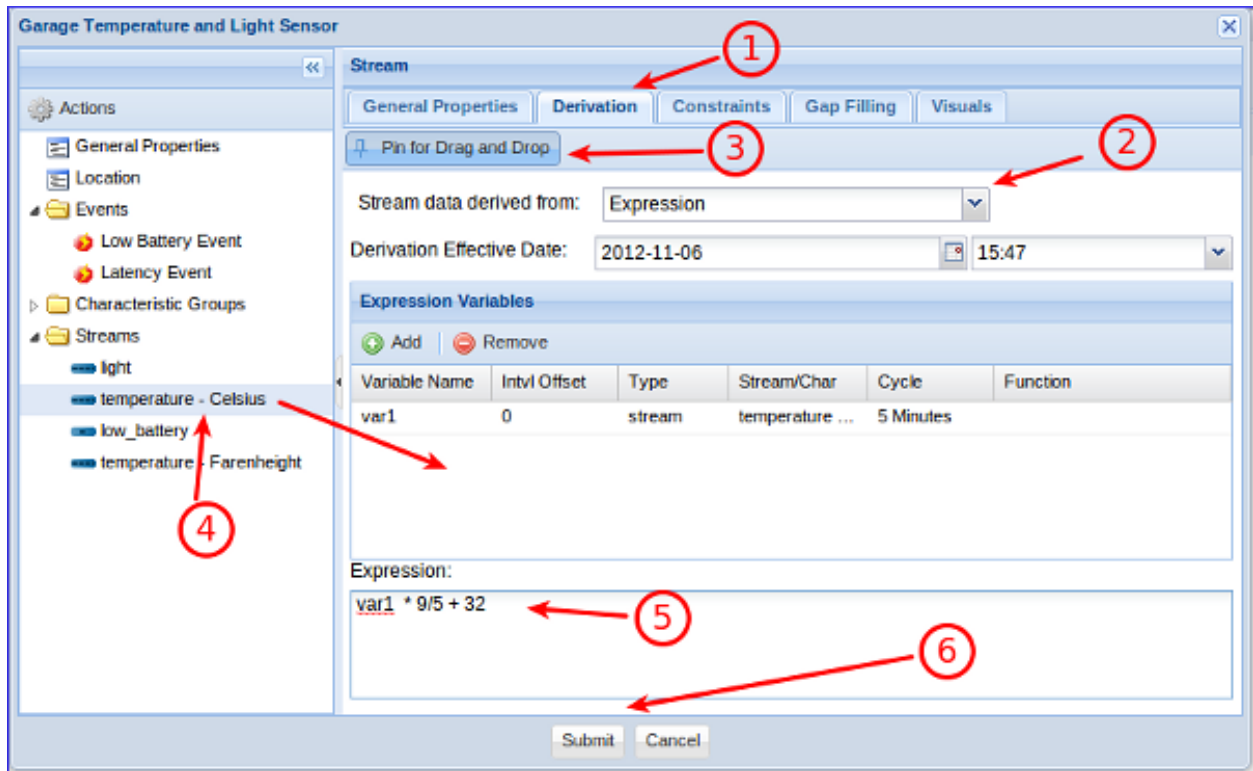
Defining an Expression Derived Stream

This section will walk you through creating a simple expression derived stream. It will derive a Fahrenheit stream based on an actual Celsius stream. This example will create the derived stream right on the existing component. If you want all newly registered components to have the new derived stream, then create the stream on the component template instead of the component.



1. Right click on the garage temperature sensor and choose **Edit Component**.

- Right click on the Streams folder and choose **Add Interval Stream**. Ensure the new stream is selected.
- Set the following attributes to the values in the above image.



- Select the **Derivation** tab on the new stream.
- Set Stream data derived from to **Expression**.
- You can click the **Add** button to add variables, but an easier way to add a stream variable is to click the **Pin for Drag and Drop** button and then...
- Drag “temperature – Celsius” onto the expression variables grid.
- Type the following formula into the Expression box: $var1 * 9/5 + 32$
- Click **Submit**. This stream will now be derived from the Celsius stream every few seconds as Celsius data comes in and the GroveStreams derivation engine runs.
 - Derived stream variables are only calculated when all dependent intervals have been uploaded into GroveStreams. If any dependent variable has a value of null (is a gap), then the derived value will also be null (a gap).

If an error occurs during derivation, possibly caused by a malformed expression, the error will appear as a Job error since derivation occurs as an asynchronous job that runs every few seconds. Check for derivation errors by clicking on the **Notification** button and selecting the **Jobs** tab if your derived stream is not producing any values.

Defining an RSS Derived Stream

This section describes how to create a component with a stream that is derived from information contained within an RSS ([Really Simple Syndication](#)) feed. The World Wide Web is full of information obtainable via RSS feeds. Most feeds are free. Some have subscriptions.

RSS feeds are a great way to import near real-time data into GroveStreams from many different sources. RSS feeds exist for things like:

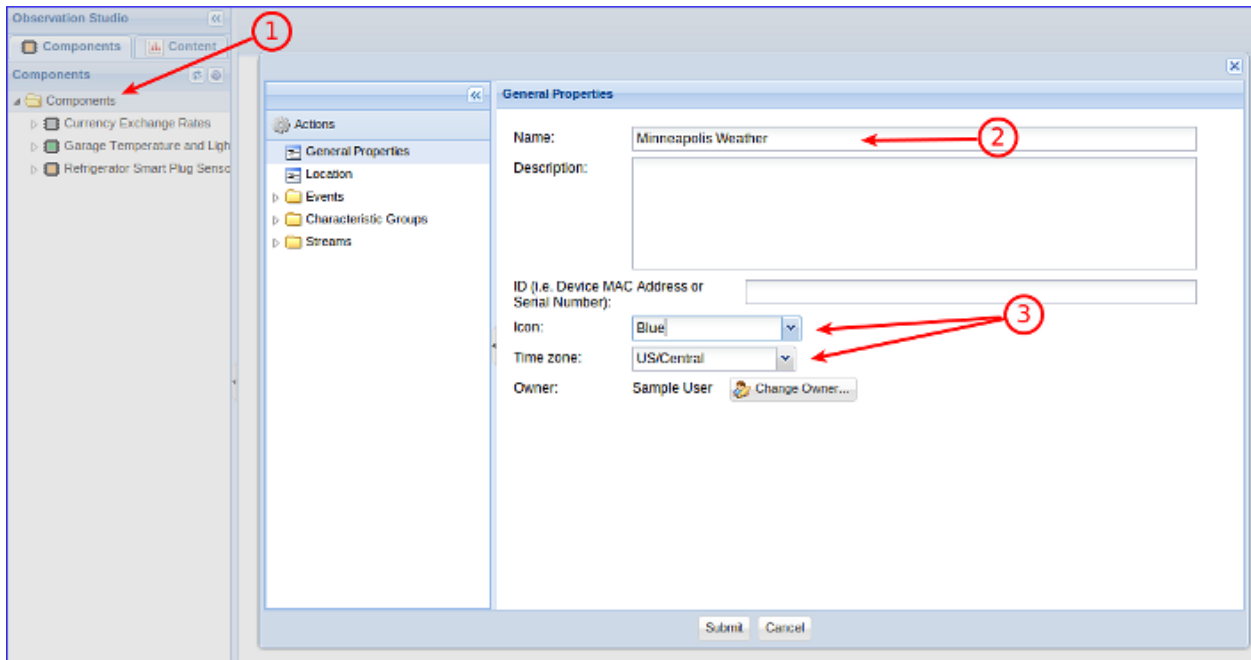
- Stock quotes
- Weather information
- Currency exchange rates
- Latest news articles

Once RSS data is placed within a GroveStreams stream, it can be used throughout GroveStreams for things like:

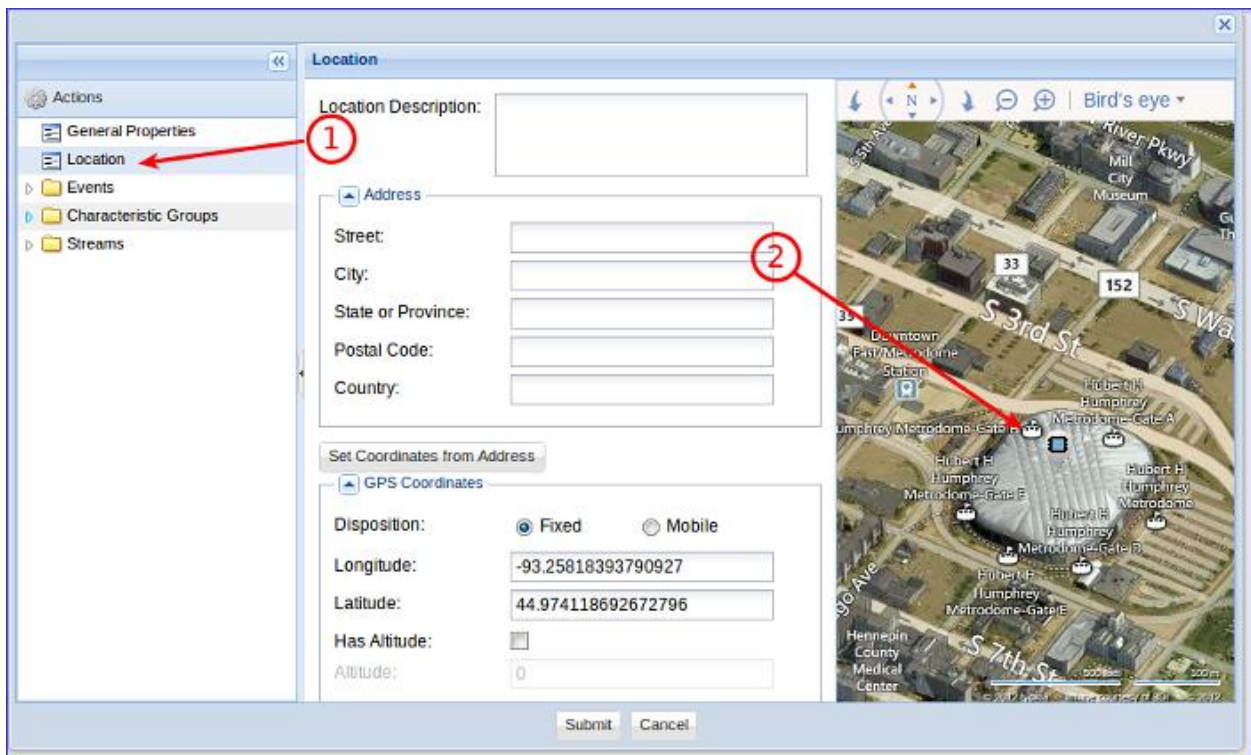
- Viewing in dashboards including mapping and graphing
- Used as a parameter for other expression derived streams
- Rollup calendar statistics
- Alerting
- Comparing to other streams
- Exposing the RSS filtered result as a GroveStreams RSS feed

Your sandbox organization was created with a EUR/USD exchange rate component and stream that gathers the exchange rate once an hour and uses it to calculate Refrigerator costs in Euros.

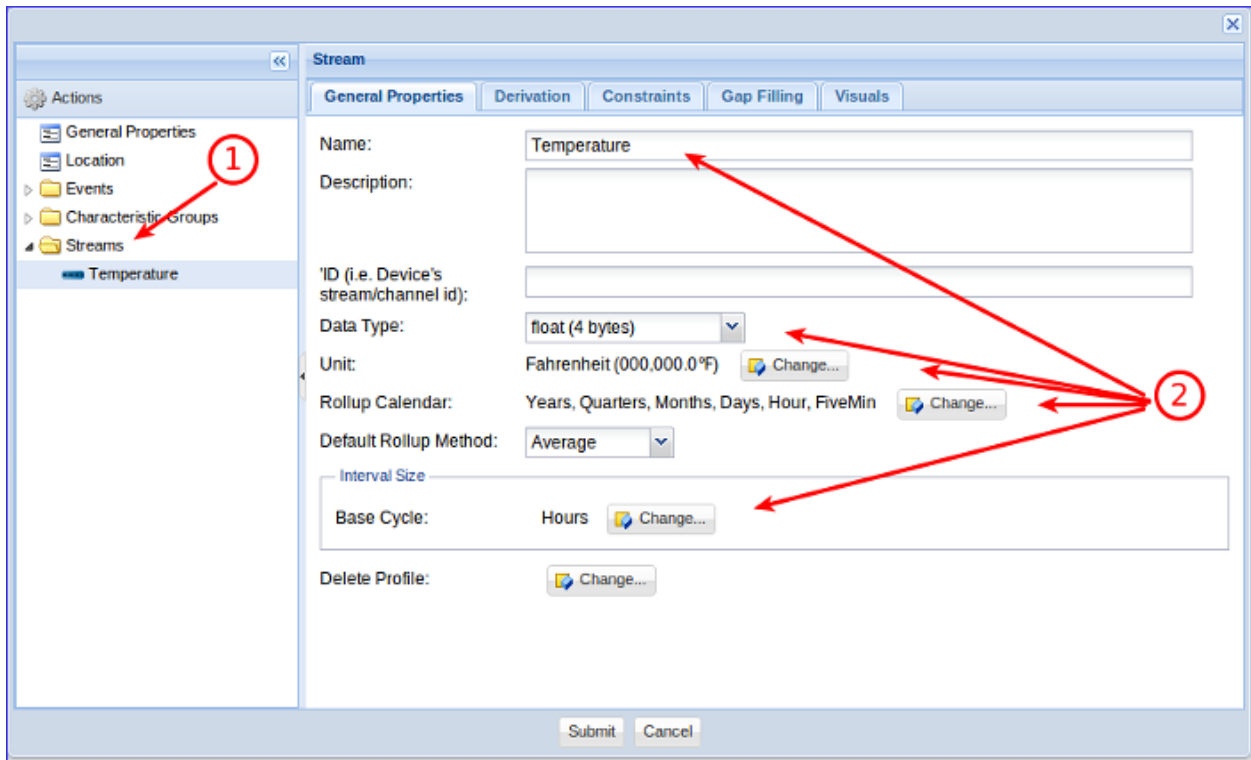
For this exercise, we will extract out an hourly temperature value from a free [NOAA](#) RSS feed and place the result into an hourly interval stream.



1. Right click on the **Components** folder and choose **New - Component**
2. Enter a name for your new component. Since we will be extracting the last observed Minneapolis temperature, we will name the component “Minneapolis Weather”.
3. Select an Icon color and the component’s time zone.



1. Select Location
2. Navigate within the map and click where the component is located to set its geo coordinates. Since the component represents Minneapolis weather, we'll choose downtown Minneapolis

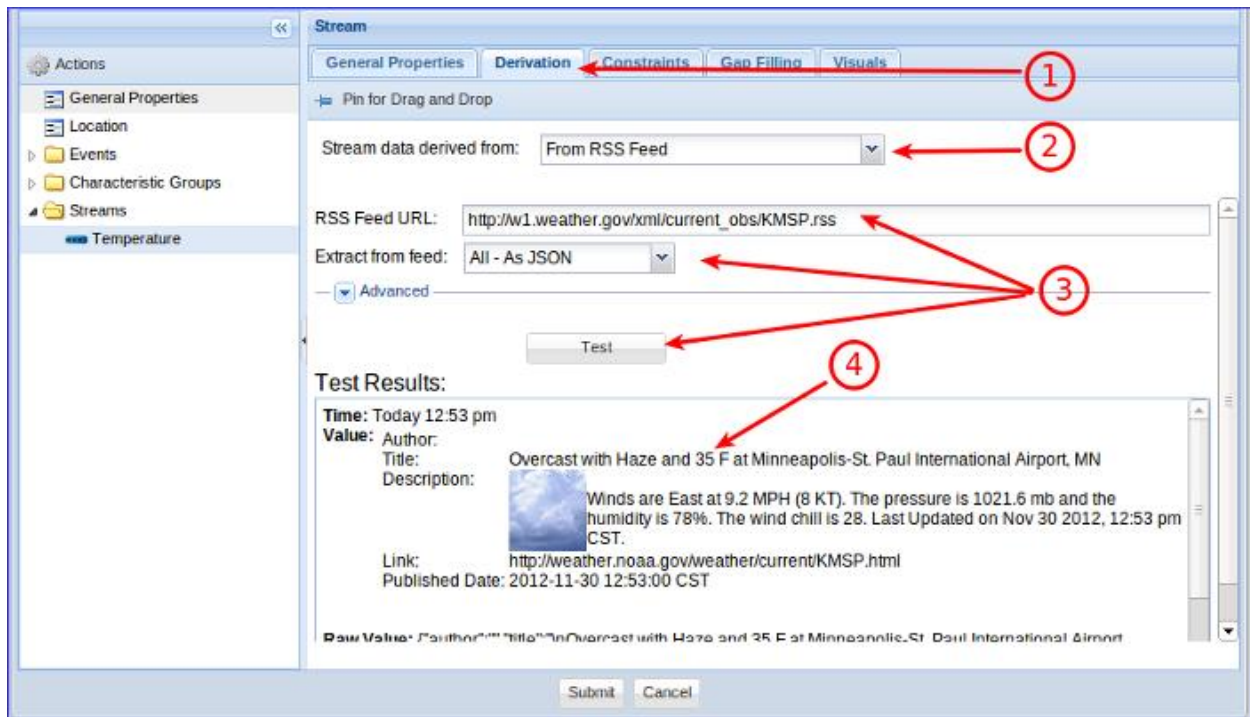


1. Right click on the streams folder and choose **Add Interval Stream**.
2. Give the stream a name. It can be any name. Set the rest of the fields to what is in the above image.

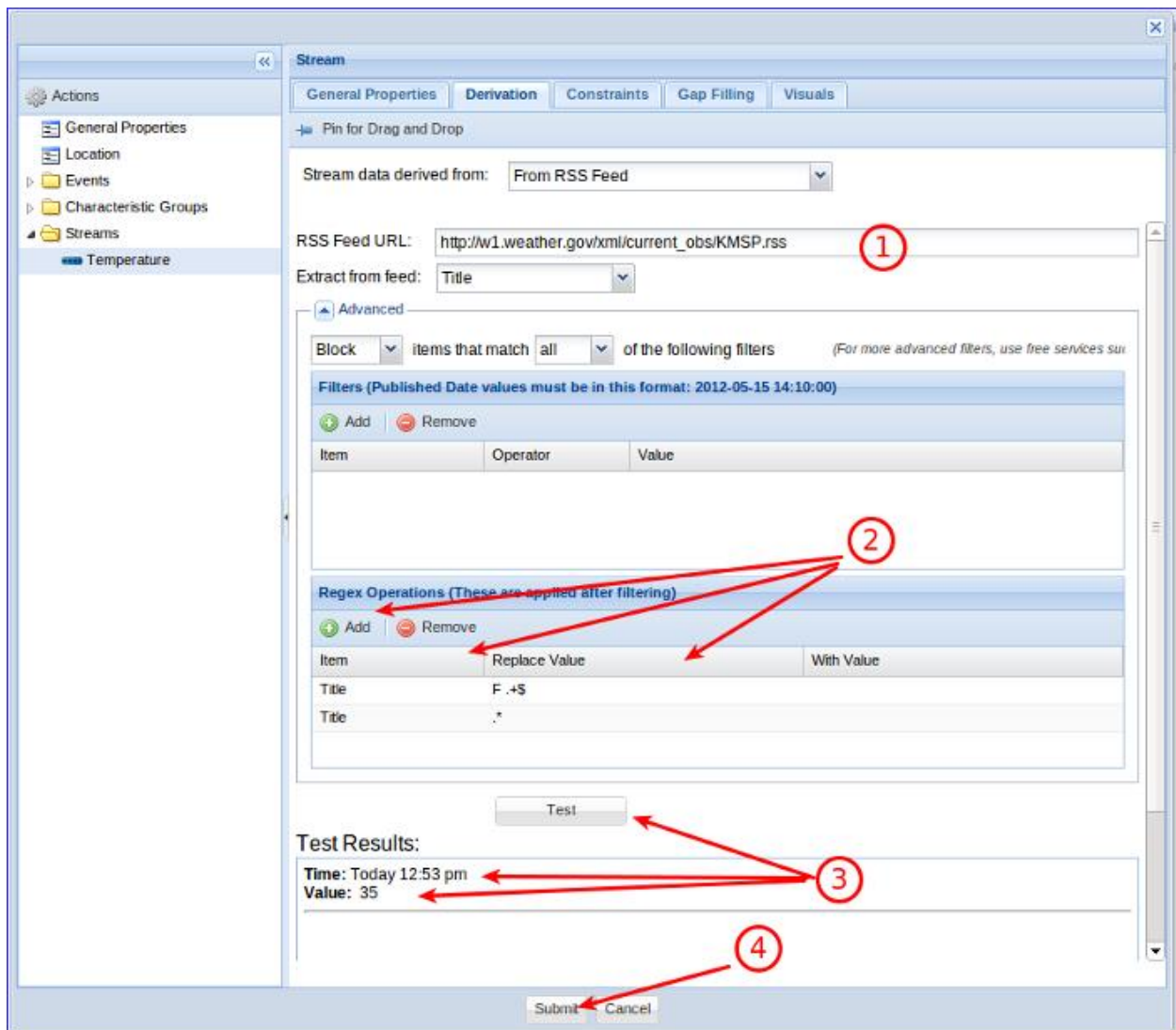


For this example, we will extract data from a NOAA RSS feed. NOAA's RSS feed library can be found here: <http://www.nws.noaa.gov/rss> .

1. I have selected a feed from Observed Conditions by opening my browser and going to this page [National Hourly Aviation Weather Observations](#) and selecting Minnesota.
2. Copy the RSS URL to your clipboard: Do this by right clicking on the RSS button next to Minneapolis-St. Paul International Airport (KMSP) and selecting:
 - a. Copy Shortcut (for Internet Explorer)
 - b. Copy Link Address (for Chrome)
 - c. Copy Link Location (for Firefox)



1. Choose the **Derivation** tab
2. Select **From RSS Feed**
3. Paste the RSS feed URL you copied to the clipboard above into the RSS Feed URL field. The URL should be: http://w1.weather.gov/xml/current_obs/KMSP.rss . Set **Extract from feed** to All-As JSON and click the **Test** button.
4. The test result is the latest Minneapolis weather feed from NOAA. The result is a nicely formatted string that is easily viewable by people, but it makes it difficult to use the values within any type of calculations or for graphing. The next step will demonstrate how to filter out only the numerical temperature from the full result.



1. Change **Extract from feed** to Title (since the temperature is located within the title field) and click on Advanced.
2. We need to create two Regex operations to trim the text before and after the actual temperature value so that only the value is returned. Regex is a powerful tool for searching and replacing elements of text. It is commonly used for RSS parsing. Google Regex to learn more about Regex usage.
 - a. Click the **Add** button **twice** so that two regex rows appear.
 - b. In the first row, choose **Title** item and enter the following for the **Replace Value**: " F .+\$" (only the value between the quotes. Include the space before the F). Leave the **With Value** blank. This regex expression will remove everything after the temperature value.
 - c. In the second row, choose **Title** item and enter the following for the **Replace Value**: " .* " (only the value between the quotes). Leave the **With Value** blank. This regex expression will remove everything before the temperature value.
3. Click **Test** and verify that only the temperature is returned.

- a. The **Time** will be used for the sample time. Time is extracted from the feed items published dates. If those dates don't exist then it will use the feed's published date. If that date doesn't exist then it will use the current system time.
- b. The **Value** will be used for the sample value.

4. Click **Submit** to save your new component.

GroveStreams will import RSS data into your streams when you save your component and then once every hour.

There are many popular RSS feed websites that can:

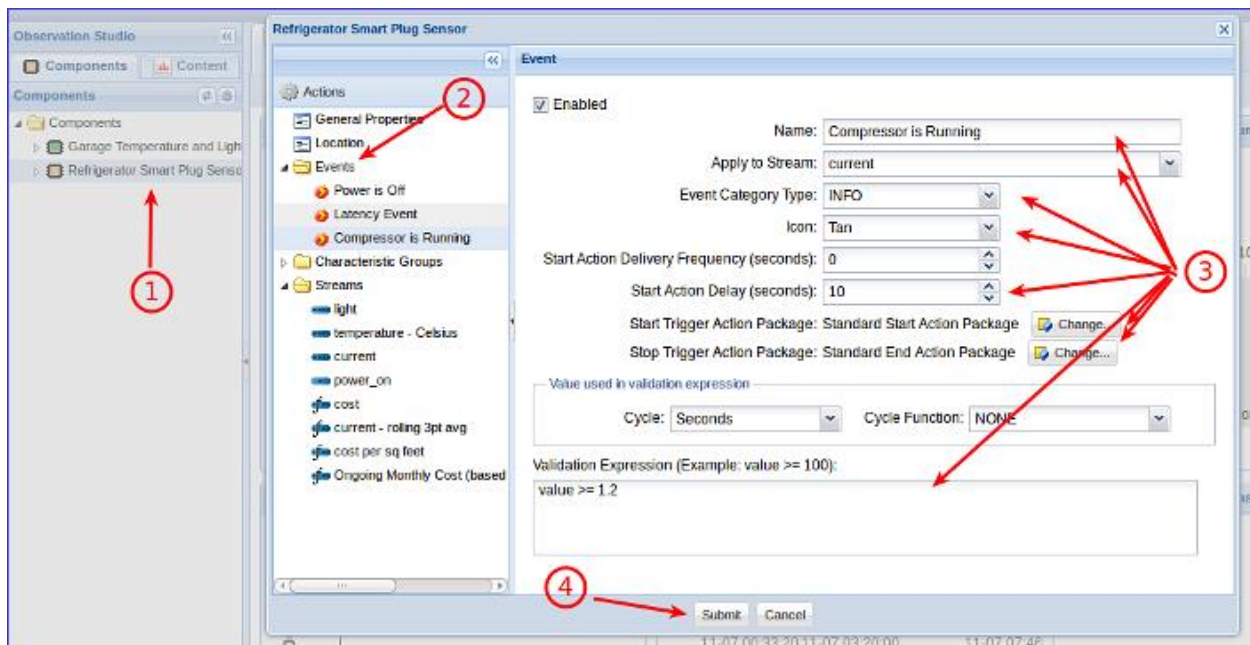
- Aggregate and Mashup Feeds into One Feed
- Convert web pages and XML files into RSS feeds
- And much more...

Search the GroveStreams Forum for more more RSS Feed RegEx examples.

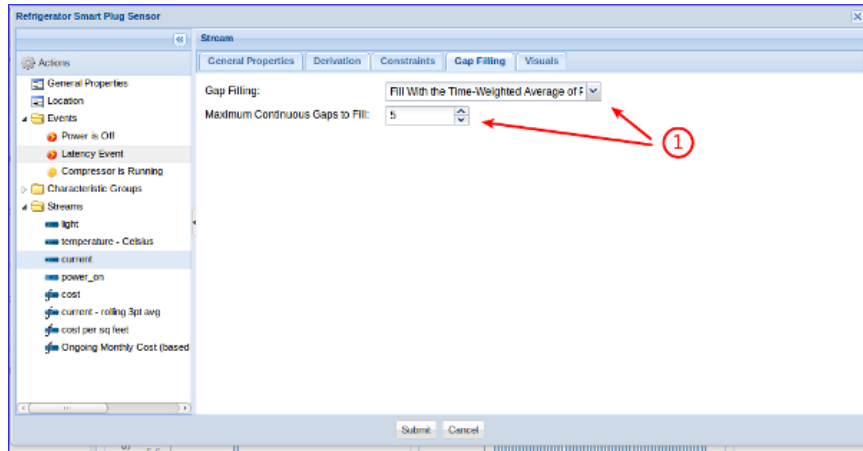
Congratulations! You have successfully created a stream derived from an RSS feed. You can use the stream throughout GroveStreams for any stream operation.

Creating an Event with Notifications

This section describes how to create an event, with notifications, on a component. It describes how to add an event to detect when the refrigerator compressor runs which occurs every few minutes.



1. Right click on the Refrigerator component and choose **Edit Component**.
 - a. The refrigerator current stream contains many gaps. A gap can currently cause the event engine to think an event has ended and then retrigger the event when the next non-gap value is uploaded. To prevent this scenario from filling your inbox with notifications, let's have GroveStreams automatically perform gap filling when feed data is uploaded but prior to event processing.



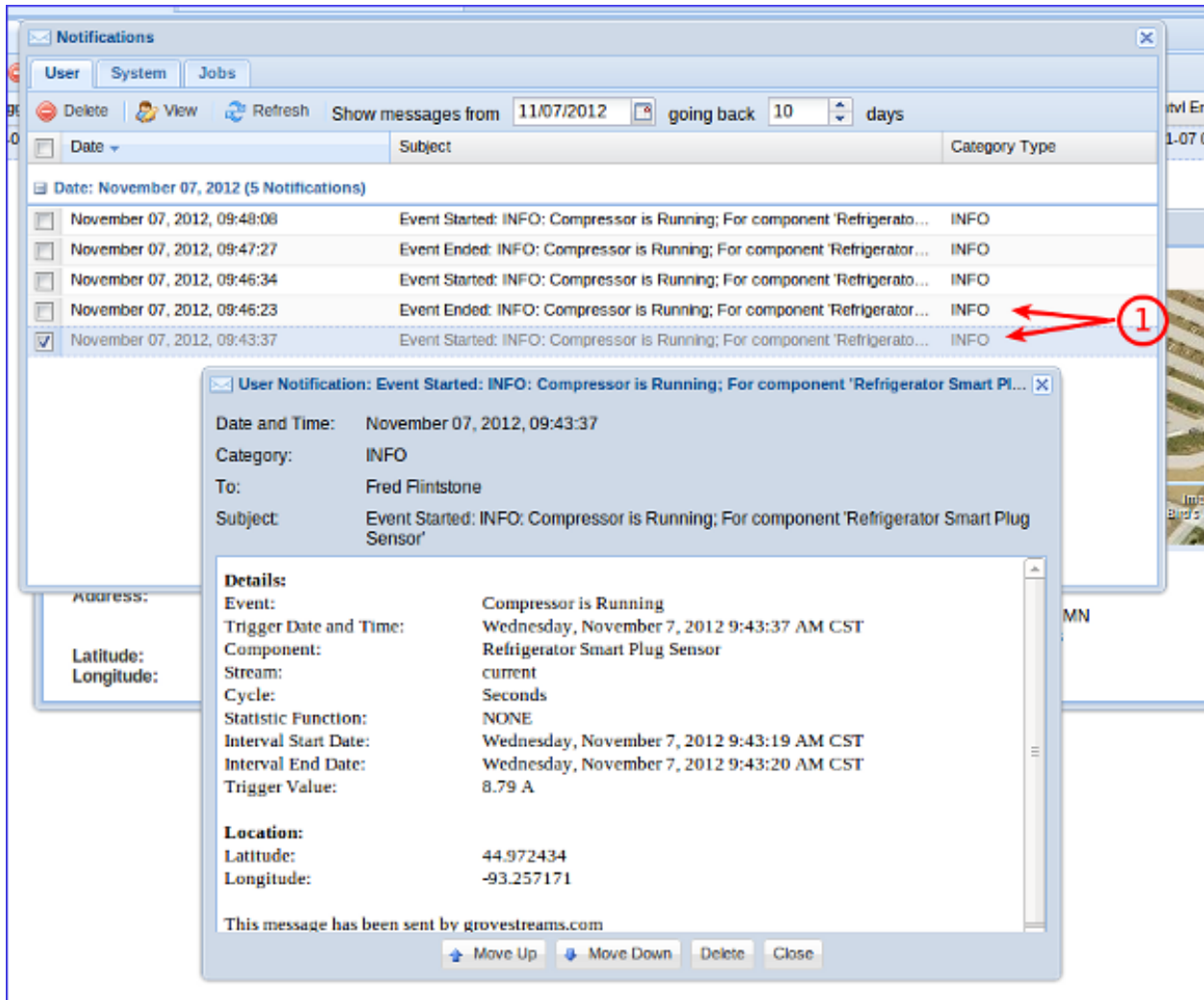
- b.
 - c. 1. Select the “current” stream on the left, select the **Gap Filling** tab, select **Fill With Time-Weighted Average ...** and set max gaps to fill to **five**.
2. Select the **Events** folder, right click it and choose **Add Event – Trigger Event**.
3. Set the following properties as seen in the image above:
 - a. Select an **Icon**. The icon is used in the tree view and in maps.
 - b. The **Start Action Delay** amount will prevent the event from starting until the below expression is true for the amount of seconds entered here. Setting a delay will prevent anomalies from triggering the event such as ampere spikes that may occur for only one second or maybe when the refrigerator door is opened for less than ten seconds – the light bulb amperes will increase ampere usage. If the door is opened for more than 10 seconds then we may get an invalid compressor event.
 - c. The Trigger action packages were automatically created when the sandbox organization was created. Each of these packages will send GroveStreams user notifications and emails to you, the person that created the organization. You can click the **Change** button, select a package and choose **Edit** to view the package definition. Packages can also be designed to send SMS text messages and make HTTP calls. Action packages are designed to be shared across components within an organization.
4. Click OK to save your changes to the component. Your event logic will now be active. Trigger events are processed as feed uploads occur and during stream derivation.

Value expressions can be based on virtual streams (rollup calendar cycle functions) to capture events such as:

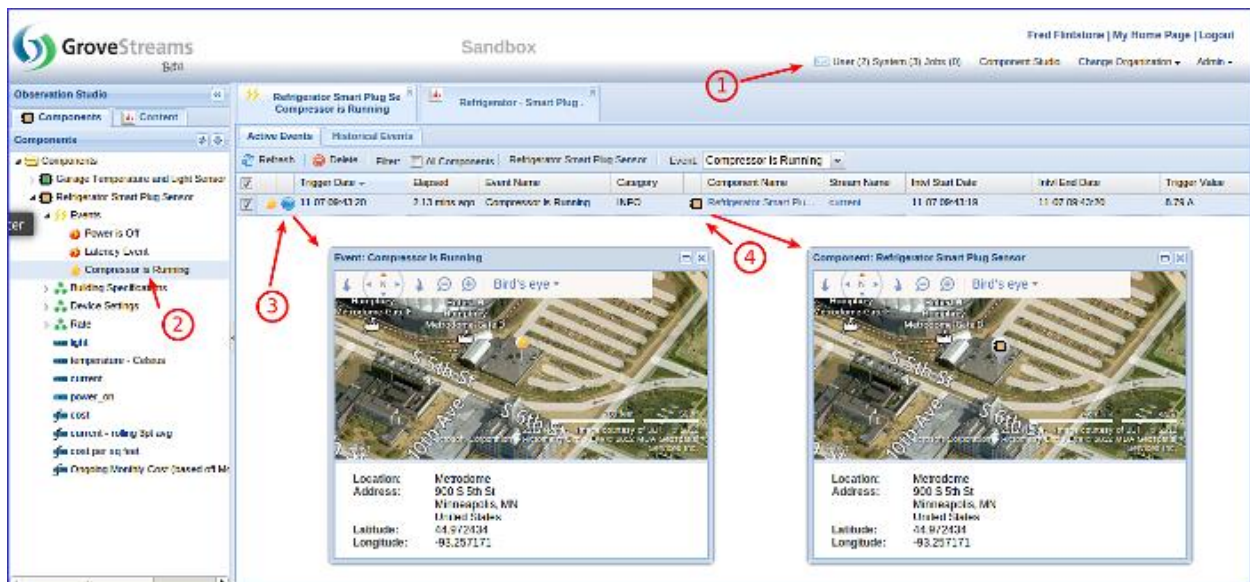
- Start an event if the daily maximum is greater than a certain amount.
- Start an event if there are too many gaps in a 5 minute interval.

- Start an event if my energy billing metric has exceeded a certain amount.

A trigger expression allows for complex expressions, but sometimes a more complex trigger needs to be defined. Create a derived stream to model more complex scenarios that involve other streams and previous intervals and set an event on the derived stream.



1. After a few minutes, you should receive a start event user notification. Click on the notifications button located at the top of the studio. Double click the notification to view its details. Check the email address you used when you registered with GroveStreams to see if you received an email describing the event. Since we associated a Start and Stop Action Package, you will receive a notification when the event starts and when it ends.



1. Notification button. Notice a number appears in the button for each type of notification that has not yet been read.
2. You can view active events (events that have started but have not yet ended) by double clicking the event under the component in the left tree panel. Although, there is a **Historical Events** tab, GroveStreams does not currently store historical events. That functionality is on our roadmap.
3. Click on the map icon to view where the event took place.
4. Click on the component icon to view the component's current location. This is useful if your component is a mobile component.

Active events also appear in Map views. Create a dashboard map and watch for event icons to appear as events start. You may have to filter out component icons to see the event icons. Since sandbox components have a fixed location, event icons will appear exactly where the component icons are located and may be obscured by the component icon.

To disable the event so that you no longer receive emails and notifications, simply edit the component, select the event and uncheck the enabled field. Then save the component.