

**Getting Started with Kepler  
Sensor View 0.9  
and the Kepler Sensor Platform**

Jun 2012

# Getting Started with the Sensor View Suite and the Kepler Sensor Platform

The Kepler Sensor Platform enables management of field-deployed sensors.

## Table of Contents

<b>1. Introduction</b> .....	<b>2</b>
<b>2. Downloading and Installing the Sensor View Suite</b> .....	<b>3</b>
<b>3. An Overview of the Kepler Sensor Platform</b> .....	<b>3</b>
<b>3.1 Field Components</b> .....	<b>3</b>
3.1.1 SPAN.....	4
3.1.2 DataTurbine.....	5
3.1.3 Sensor View Utilities, Sensor Simulator, and Startup Scripts .....	5
3.1.4 Sensor Manager API .....	5
<b>3.2 Desktop Components</b> .....	<b>5</b>
3.2.1 Inspection, Analysis, and Control .....	5
3.2.2 Data Archival and Retrieval .....	11
<b>3.3 Server Components</b> .....	<b>12</b>
3.3.1 Metacat.....	12
3.3.2 Datafeeder (SpanToDT).....	12
3.3.3 DataTurbine.....	12
3.3.4 Workflow Run Engine and Workflow Scheduler .....	13

## 1. Introduction

This guide introduces the beta release of the Kepler Sensor Platform, which enables monitoring and controlling sensors; sensor data analysis, modeling, visualization, documentation, archival, and archive retrieval; and scheduling workflows (e.g. for QA/QC, alerting, processing, and archival) for periodic, remote execution. The Sensor View suite is an add-on Kepler module-suite, enabling access to the broader Kepler Sensor Platform. Conceptually we can divide the Platform into three areas: Desktop Components, Field Components, and Server Components. An overview is shown in Figure 1, and these three component areas are discussed in section 3.

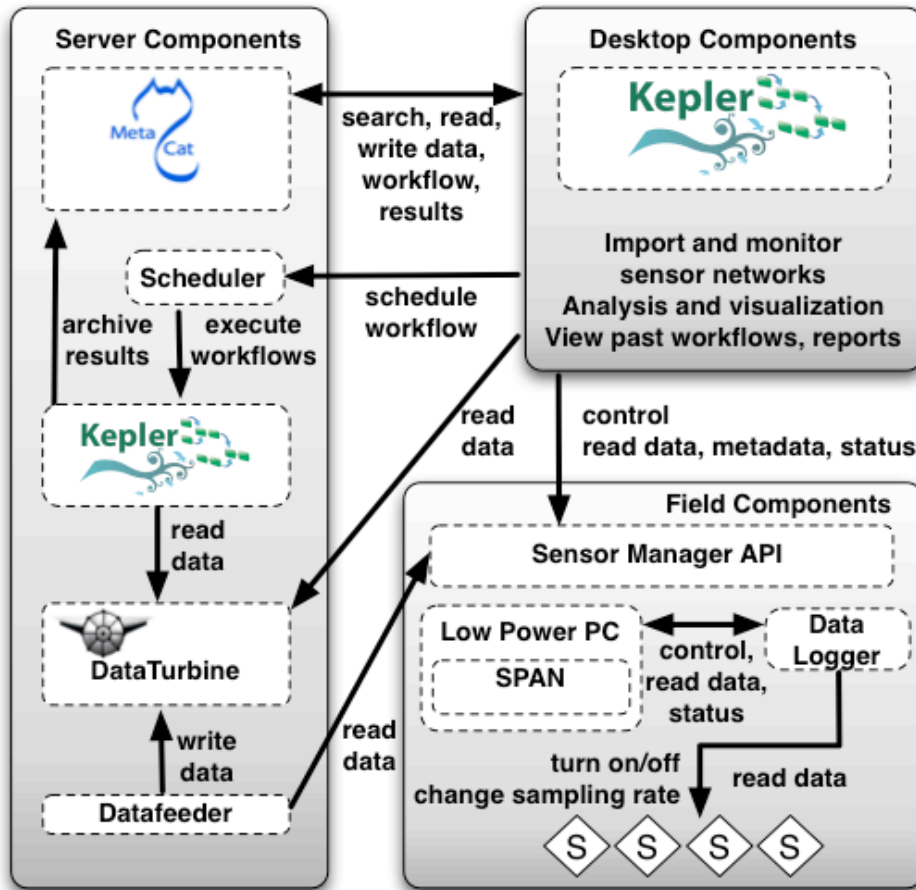


Figure 1: Kepler Sensor Platform.

## 2. Downloading and Installing the Sensor View Suite

To install the Sensor View suite, use the Module Manager accessible from the main Kepler menu. Select Tools => Module Manager. From the Module Manager dialog, select the Available Modules tab. Select 'sensor-view-0.9' from the list of Available Suites, then click the right arrow button to move it to the list of Selected Modules. Click the 'Apply and Restart' button to download and restart into the suite.

## 3. An Overview of the Kepler Sensor Platform

Here we discuss key pieces and capabilities of the three main component areas of the Kepler Sensor Platform. It's important to keep in mind much of the architecture is flexible, and need not necessarily be utilized, or divided across server, desktop, and field deployment exactly as discussed. For example, you may not wish to deploy a low power PC to the field, instead choosing to run SPAN (discussed below) and other software on a server.

### 3.1 Field Components

For the sake of this overview, we will consider a sensor site that consists of a low power PC (e.g. Gumstix), connected to a datalogger (e.g. Campbell Scientific CR800), connected to a number of sensors.

### 3.1.1 SPAN

The Kepler Sensor Platform utilizes a software developed by University Southern California/Information Sciences Institute called Sensor Processing and Acquisition Network (SPAN). We have fixed bugs and added features to SPAN, and host our fork at: <https://code.ecoinformatics.org/code/span/trunk/>. Instructions for configuring and compiling SPAN, as well as a detailed API document (including overview and architecture diagrams and descriptions) are available there. SPAN is written in C, licensed under the LGPL, and provides drivers for controlling, and receiving, buffering, and making available data from a number of dataloggers, among other things. Included in our fork of SPAN and the Sensor View suite are scripts and programs for receiving data from and controlling SPAN.

An overview of the relevant software for one possible scenario, in which a low power PC is deployed to the field, is shown in Figures 2 and 3 below. This scenario differs from Figure 1 in that here we've chosen to run SpanToDT (a.k.a. Datafeeder) and DataTurbine on the deployed PC.

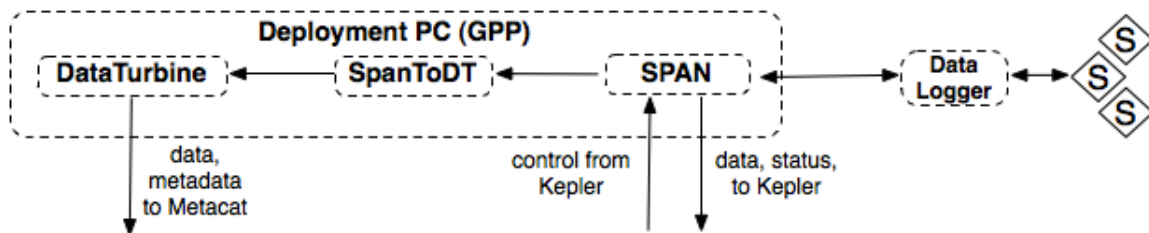


Figure 2: Field Components

Files	Description
dcd_mgr, cr1k_d, cr10x_d, crio_d, sdi12_d, vaisala_wxt_d, webcam_d	SPAN binaries
cr_parser.py	Script that converts native datalogger program into SPAN configuration file
/etc/init.d/span.sh	Script to start/stop SPAN server (dcd_mgr and cr1k_d)
rbnb-3.2b5.jar	DataTurbine
/etc/init.d/dataturbine.sh	Script to start/stop DataTurbine
sensorview-utils-1.0.jar	JAR containing SpanToDT, MirrorDT, and Sensor Simulator
/etc/init.d/spanTodt.sh	Script to start/stop SpanToDT

Figure 3: Field Component Software

### 3.1.2 DataTurbine

DataTurbine is an open-source real-time streaming data middleware that can be used to buffer and access data. Details about the project may be found at <http://www.dataturbine.org/>. Kepler provides access to data in DataTurbine servers through a DataTurbine actor, and the Kepler Sensor Platform includes functionality for pushing sensor data and associated metadata into DataTurbine, and for retrieving and automating the archival of this data as EML-described data packages into a Metacat server. In the Kepler Sensor Platform, a DataTurbine can optionally be used on a low power field-deployed PC to provide a layer of buffering, or to leverage its mirroring capability. Note that SPAN also provides a simple file-based buffering mechanism, writing data to files in a “pending” folder.

### 3.1.3 Sensor View Utilities, Sensor Simulator, and Startup Scripts

The Sensor View Kepler suite includes useful programs that may be run independently of Kepler, on systems like the field deployed PC or server. These programs are bundled into `sensorview-utils-1.0.jar`, found in the Kepler sensor-view module directory: `KeplerData/kepler.modules/sensor-view-0.9.0/lib/jar/`. Included in the jar are `SpanToDT`, a program that transfers sensor data and metadata changes from SPAN to a DataTurbine server; `MirrorDT`, a utility to mirror data between DataTurbine servers; and the `Sensor Simulator`, which can be used to simulate a site, and is useful for experiments within Kepler. To view help on using these programs, invoke them with the `help (-h)` argument:

```
java -cp sensorview-utils-1.0.jar util.SpanToDT -h
java -cp sensorview-utils-1.0.jar util.MirrorDT -h
java -cp sensorview-utils-1.0.jar org.kepler.sensor.span.SpanSim -h
```

As shown in the table in Figure 3, we also include with the SPAN source code three scripts for starting, stopping, and restarting SPAN, DataTurbine, and `SpanToDT`. You can configure these scripts and place them in your field deployed PC's “init.d” folder so that the associated programs automatically start up should the PC lose and regain power, or be rebooted.

### 3.1.4 Sensor Manager API

The Sensor Manager API is an optional server that may be run to provide an abstraction layer between the software that interacts with the logger/sensors, and Server and Desktop components. The Sensor Manager will likely be reworked in future releases of sensor-view.

## 3.2 Desktop Components

The Kepler Sensor View suite includes a `Sensor Simulator`; `Plotting`, `Workflow Scheduler`, and `Reporting` interfaces; and functionality for representing, inspecting, controlling, analyzing, and archiving your sensor site and sensor site data from within Kepler.

### 3.2.1 Inspection, Analysis, and Control

The Sensor View suite allows you to connect to and inspect, visualize, and control your deployed or simulated sensors. To connect to your site, use the `Import Sensor Site` menu,

as shown in Figure 4 below. If you are running the Sensor Simulator (3.1.3), you can connect to “localhost”.

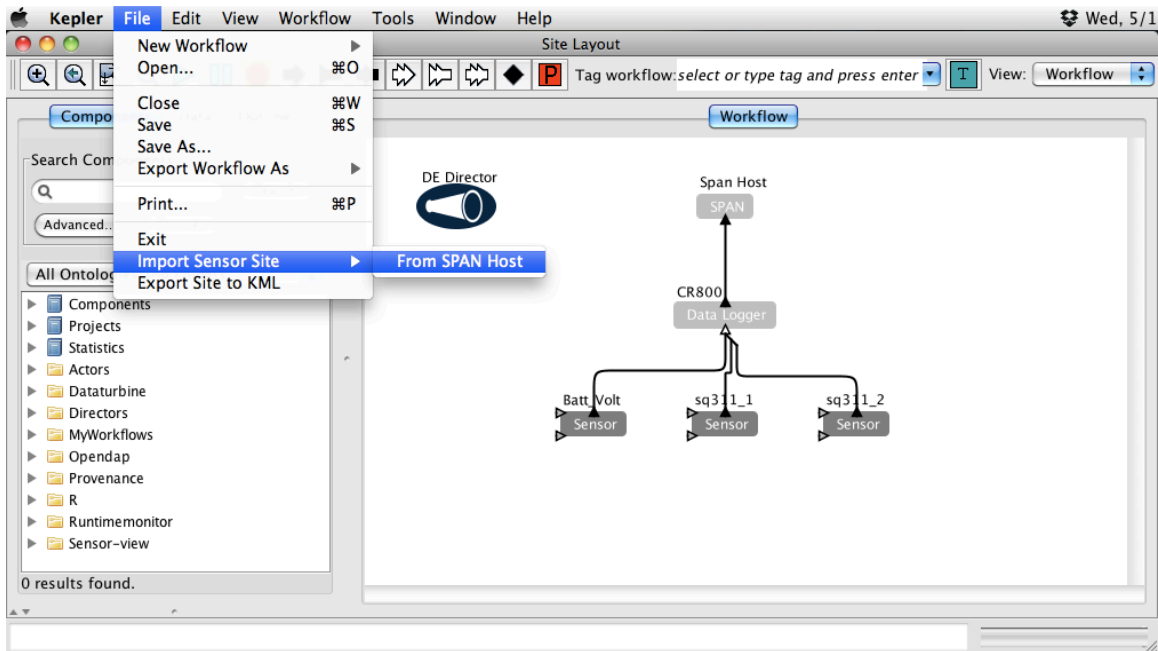


Figure 4: Importing a sensor site.

The site layout workflow is automatically generated. After import, you may execute the site layout workflow to monitor sensor data in realtime. Sensors change color based on status, and their most recent data points are displayed (Figure 5).

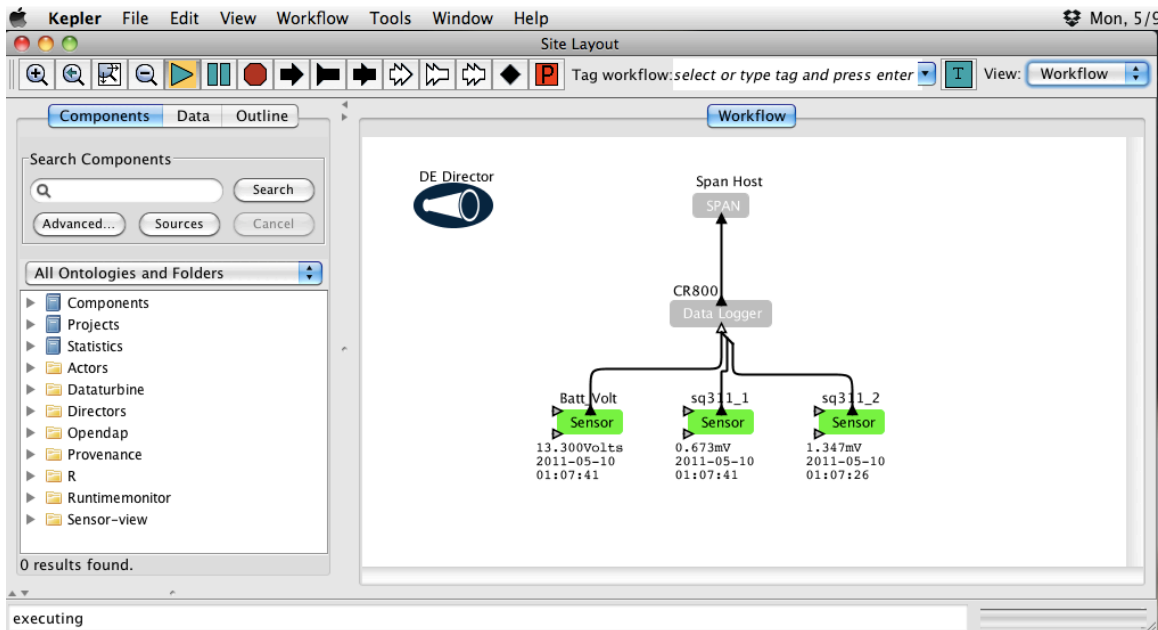


Figure 5: Realtime Monitoring.

It's worth noting that Kepler offers annotation and some primitive drawing capabilities that can be used to add contextual information to your site (Figure 6).

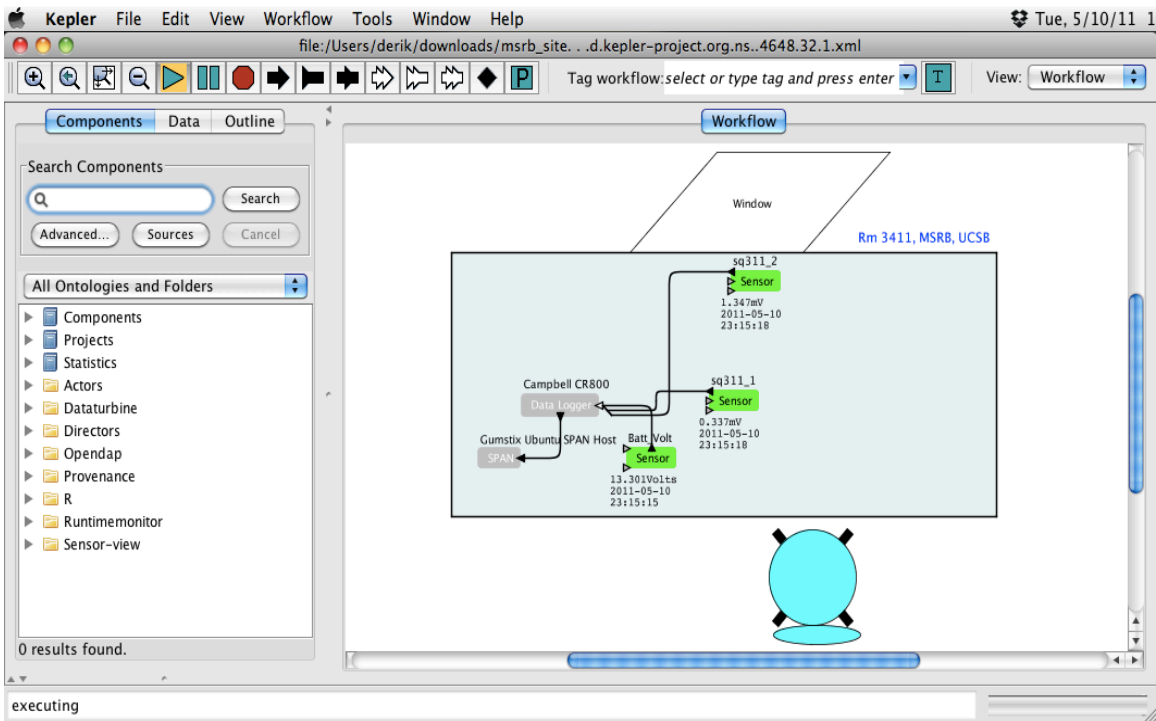


Figure 6: Site layout with contextual informational added.

You may inspect and manually control a sensor just like any other actor in Kepler, by right-clicking on it and selecting Configure Actor (Figures 7 and 8).

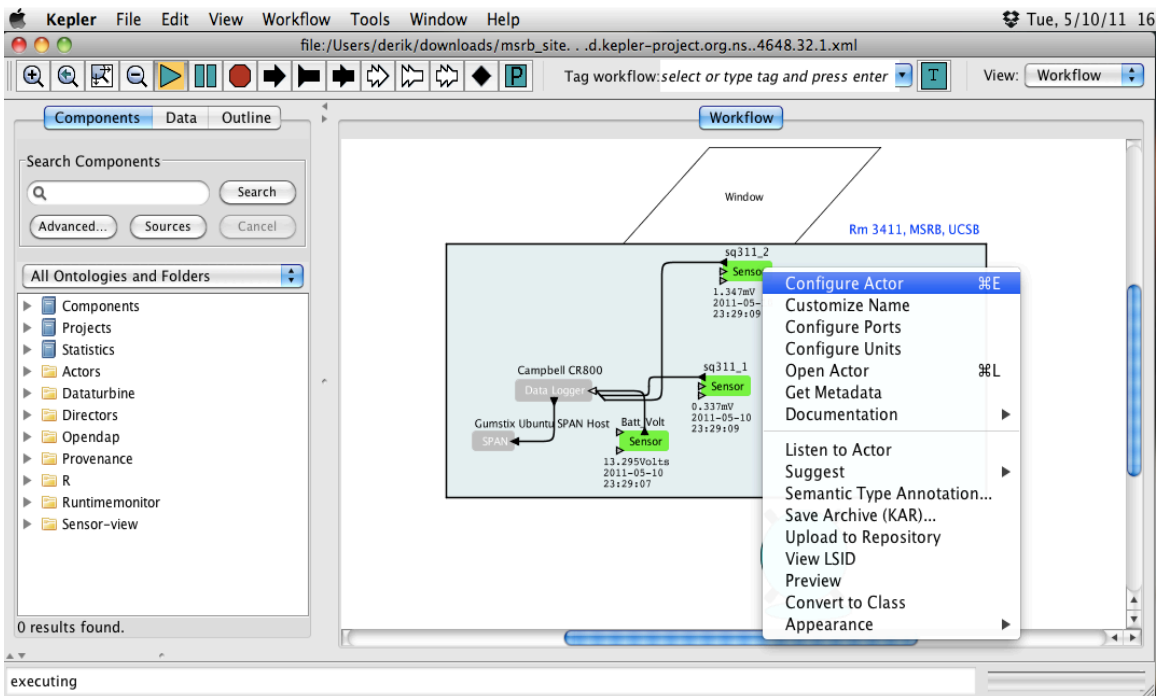
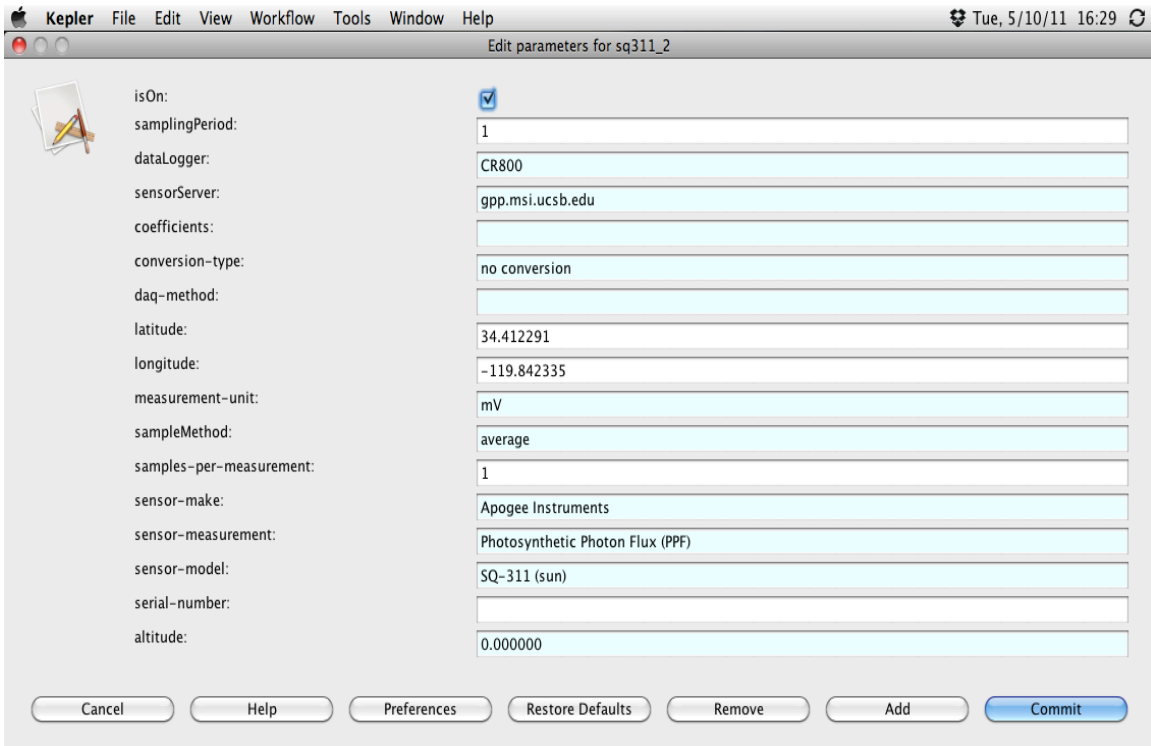


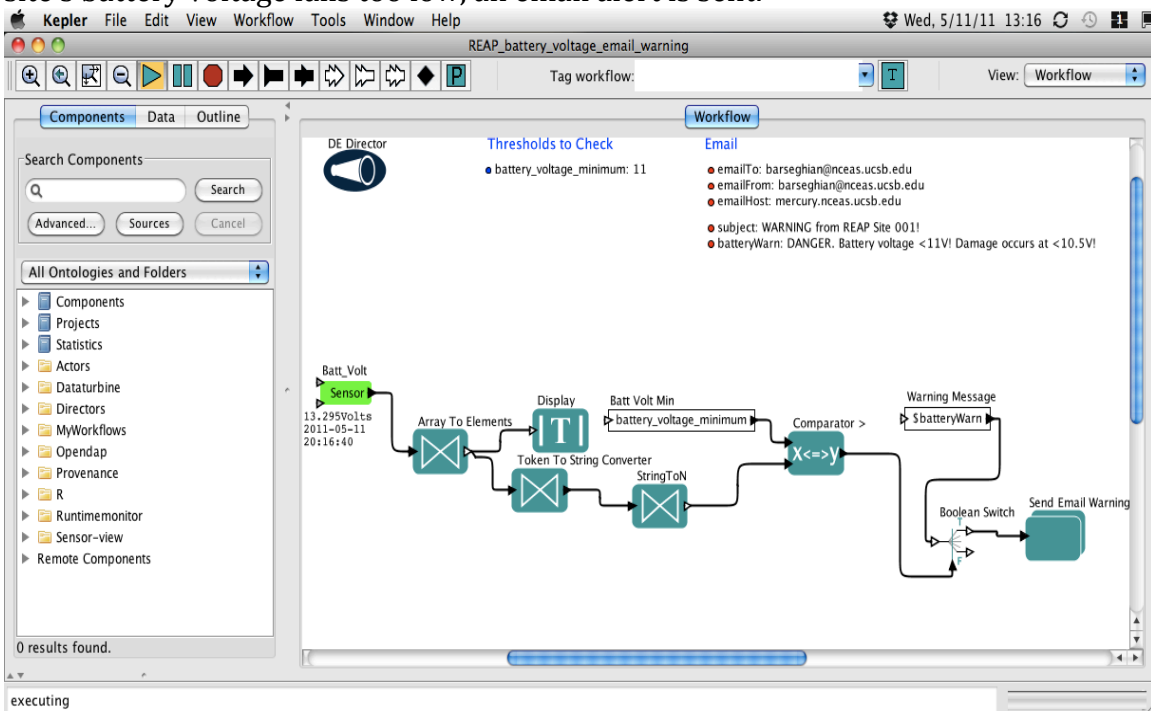
Figure 7: Configuring a sensor.



**Figure 8: Configuring a Sensor.**

As the sensor configuration dialog in Figure 8 shows, you may turn this sensor on or off, change its sampling rate, and adjust its metadata.

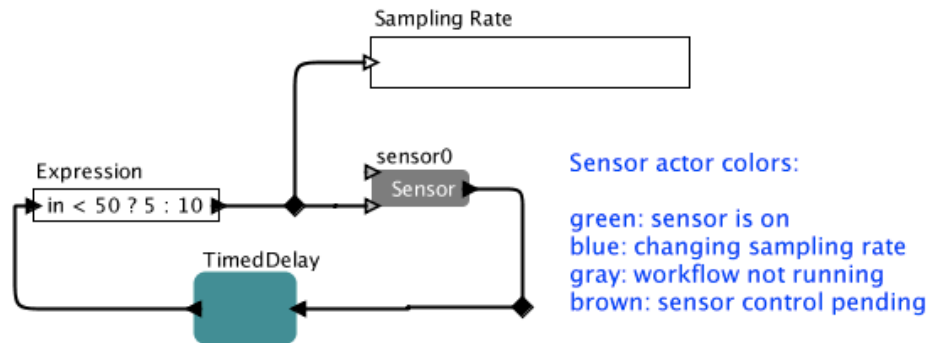
Sensor actors may be put to use in a number of ways, such as workflows that send alerts, or do continuous QA/QC. In the example workflow shown in Figure 9, a battery-voltage sensor actor serves as a live datasource, continuously outputting samples. The data points are run through a simple threshold check. If the threshold is reached, i.e. if the site's battery voltage falls too low, an email alert is sent.



**Figure 9: An example email alert workflow.**



Using sensor actors, workflows may also be designed that automate site control. Each sensor actor has two input ports through which you may toggle the sensor on/off, or change its sampling rate. The simple feedback workflow shown in Figure 10 demonstrates how a workflow can change a sensor's sampling rate (in this case, based on its own data values).



This workflow changes the sampling rate of the sensor based on the data values.

NOTE: to run this workflow, the sensor simulator must be running. Instructions are here: <https://kepler-project.org/developers/incubation/kepler-engineering-view-for-reap/running-sensor-simulator>

Daniel Crawl, SDSC/UCSD, July 2010

Included in the Sensor View suite is a Plotting view. While running a site layout, you may change to the Plotting view to quickly create and view plots of your data streams in realtime (Figures 10-12).

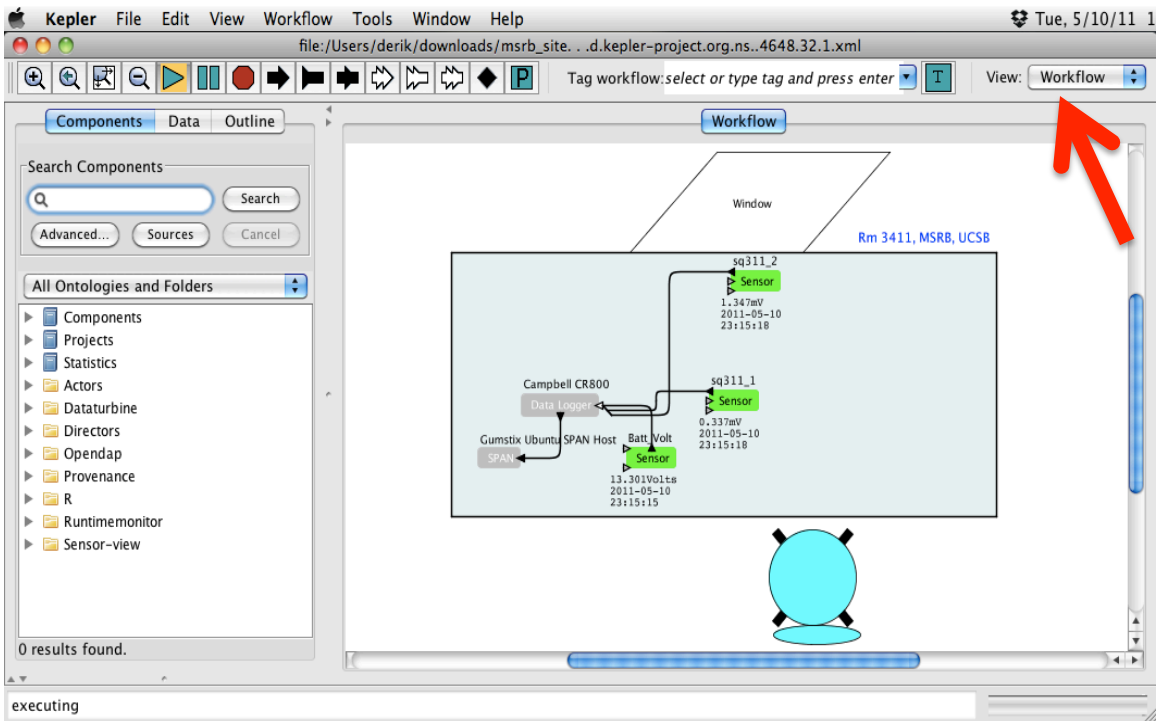


Figure 10: To use plotting, change from Workflow to the Plotting View.

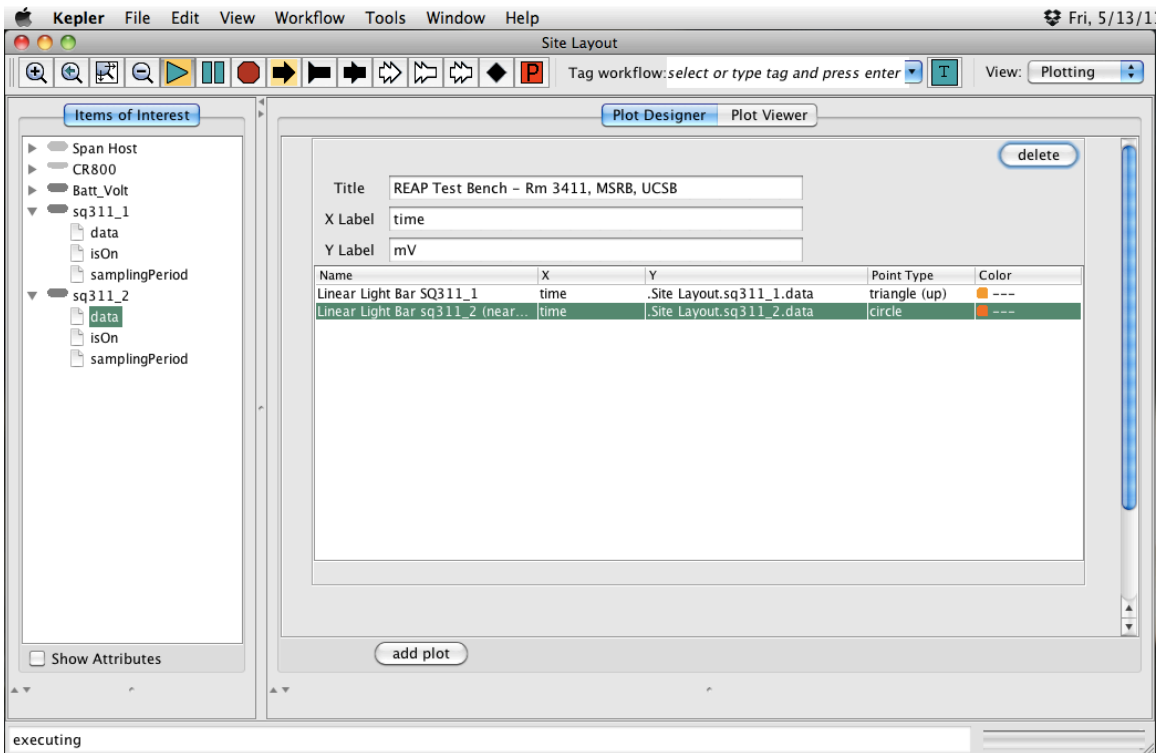


Figure 11: Design a plot by dragging sensor data ports from Items of Interest into the Plot Designer.

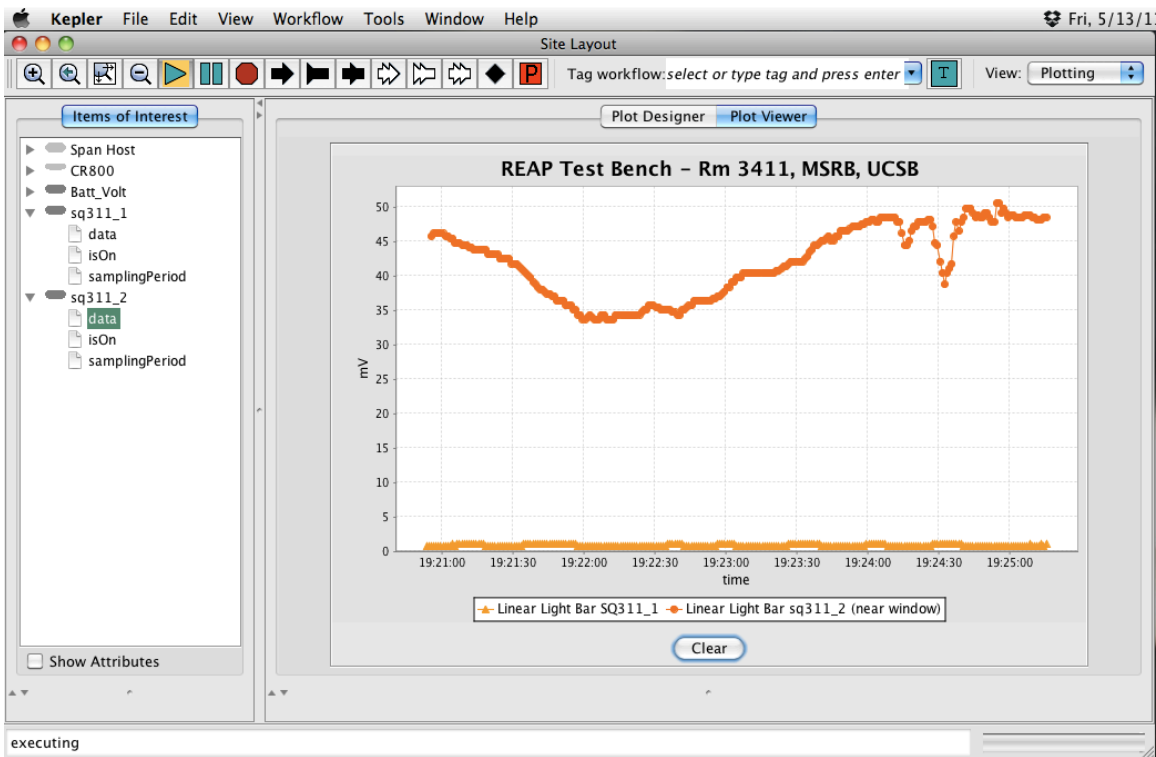


Figure 12: The Plot Viewer tab displays your designed plots in realtime.

### 3.2.2 Data Archival and Retrieval

The Sensor View suite includes a workflow (~ /KeplerData/workflows/module/sensor-view/demos/archiveDataturbineDataToMetacat.kar) that archives sensor data from a DataTurbine to a Metacat (Figure 13). This workflow can be run manually, but was designed more with automation in mind. Using the Workflow Scheduler, accessible from the Kepler Tools menu, you can schedule this workflow to be run periodically on a remote server. To learn more about the Workflow Scheduler and the Kepler Run Engine it invokes, please see the Workflow Scheduler GUI documentation.)

**This workflow archives streaming data from a DataTurbine server into a Metacat.**

For each sensor, a datapackage is created. The workflow keeps track of what data have already been archived. DataTurbine channels must follow a naming convention, i.e. this workflow expects to be run against a DataTurbine server that is receiving data from SpanTodt (details here: TODO). It's intended one person schedule this workflow to be run periodically. To schedule, use the Workflow Scheduler, from the Tools menu.

Configure these parameters:

- Source DataTurbine:
  - DataTurbineServerAddress: "gpp.msi.ucsb.edu:3333"
  - OnlyArchiveSpecificSensorIDs: true
  - SensorName: ("CR800\_sq311\_2")
  - DataLoggerName: ("CR800")
  - SiteName: ("gpp")
- Destination Metacat:
  - EcoGridPutServiceURL: "http://kepler-dev.nceas.ucsb.edu/kepler/services/PutService"
  - EcoGridAuthServiceURL: "http://kepler-dev.nceas.ucsb.edu/kepler/services/AuthenticationService"

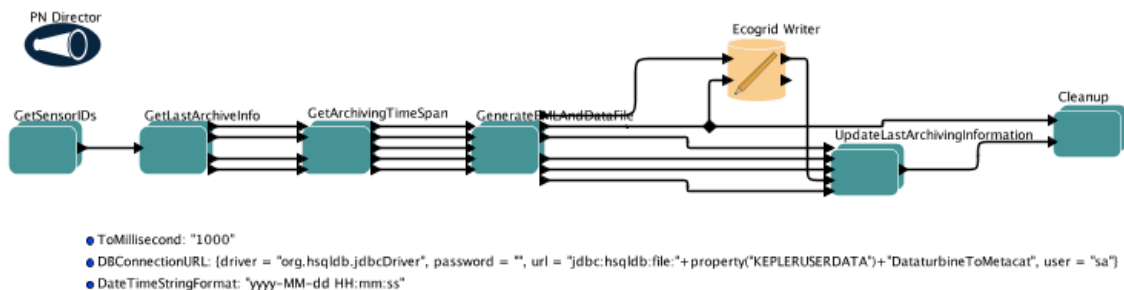


Figure 13: DataTurbine to Metacat sensor data archival workflow.

When run, the workflow gathers any new data that may be available to be archived from a DataTurbine server, chunks the data based on any metadata changes (e.g. sampling rate changes), generates EML metadata (with embedded SensorML) describing the data, and submits this metadata and data datapackage to a Metacat server. These archived datasets can then be immediately available to Kepler and other Metacat clients. From within Kepler you can find and use such archived data from the Data tab (Figure 14).

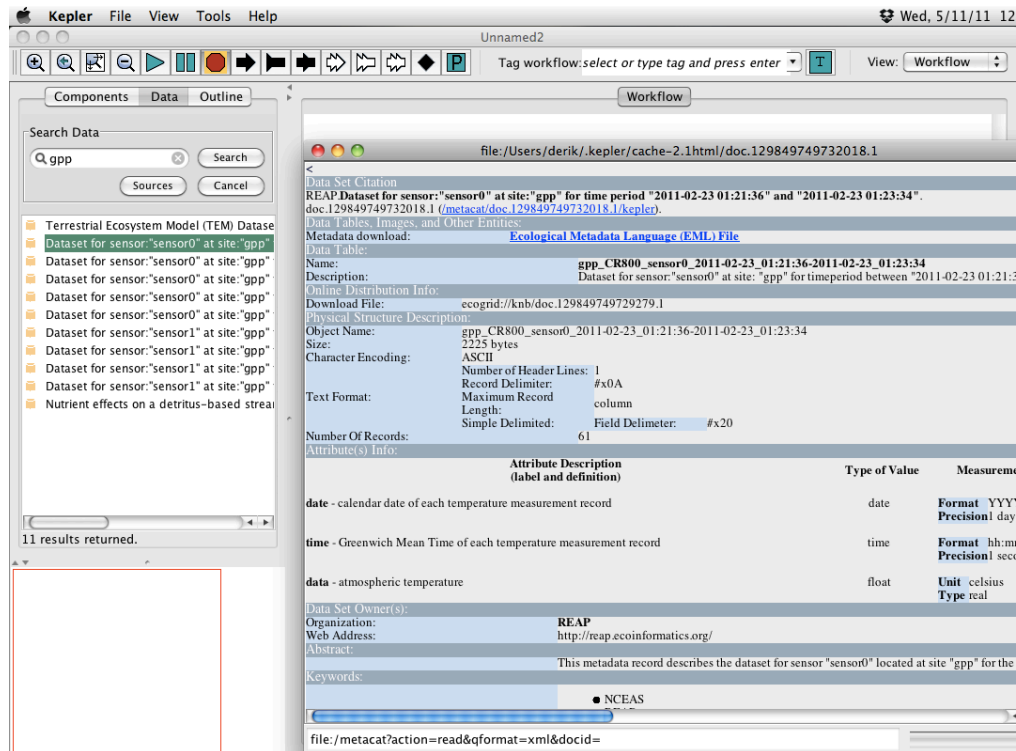


Figure 14: Finding and inspecting archived datasets within Kepler.

### 3.3 Server Components

#### 3.3.1 Metacat

The Kepler Sensor Platform uses Metacat for EML-described sensor data archival. Metacat is a repository for data and metadata that helps scientists find, understand, and effectively use data sets they manage or that have been created by others. Please see the Metacat documentation for more details, and installation information.

<http://knb.ecoinformatics.org/knb/docs/>

#### 3.3.2 Datafeeder (SpanToDT)

The Datafeeder (SpanToDT) component, described in section 3.1.3, streams data from SPAN to DataTurbine. It is designed to be run continuously, and seamlessly resume should SPAN and/or DataTurbine be restarted.

#### 3.3.3 DataTurbine

DataTurbine, described in section 3.1.2, is a real-time streaming data middleware that can be used to buffer and access data. The Kepler Sensor Platform can use it as a buffering mechanism, and as a way to quickly provide clients like Kepler with access to

data streams. Using the DataTurbine actor, you may request portions or streams of sensor data in a variety of ways.

#### **3.3.4 Workflow Run Engine and Workflow Scheduler**

Kepler may also run on a server as a webservice called the Workflow Run Engine. The Workflow Run Engine supports SOAP-based and RESTful invocation. The Workflow Scheduler webservice invokes the Workflow Run Engine, and handles requests to schedule, delete, and serve up workflows. For more details on these applications, please see the Workflow Scheduler GUI documentation, accessible from the Kepler Help=>Modules Documentation menu item. One of the main ways in which the Kepler Sensor Platform uses these services is in the automated archival of sensor data, using the DataTurbine to Metacat workflow discussed in section 3.2.1.