philadelphia Documentation

Release 0.9

Andy Mastbaum

February 26, 2014

Contents

Philadelphia is a shift report system designed for the SNO+ neutrino experiment. It is inspired by Manhattan, the shift report system used for the SNO (Sudbury Neutrino Observatory) experiment, but makes several improvements.

The Philadelphia software itself contains nothing SNO+-specific, and so is easily portable to other sites.

Requirements: Philadelphia requires JavaScript to be enabled in your browser. Internet Explorer is not supported.

Source code: Philadelphia is free software. Source is available on github.

Bug reports: Please report any bugs through the Issue Tracker.

CHAPTER 1

Motivation

The major strength and weakness of Manhattan was the flexibility of its input. The shift report was built by pasting templates into a large text box and filling in or modifying as needed. This is good, because it allows the shift operator to enter any relevant data in any format. It is bad, because this data is extremely difficult to query, plot, etc.

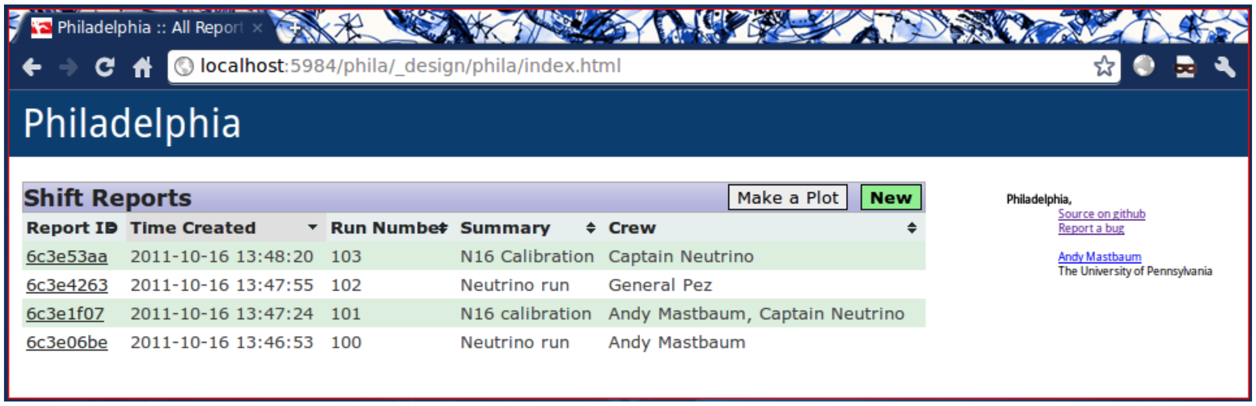
Philadelphia's interface aims to preserve the flexibility of Manhattan while introducing just enough structure to make queries possible. As in Manhattan, the report is composed by instantiating templates. Philadelphia's templates consist of form fields, but they can be removed and new ones can be added as the user sees fit. Additionally, free text-entry template gives the user a canvas to enter large amounts of uncategorized data, as was possible in Manhattan.

All fields in Philadelphia are queryable and plottable, even those added by users.

CHAPTER 2

User's Guide

2.1 The Index Page


Philadelphia :: All Report
localhost:5984/phila/_design/phila/index.html
Philadelphia
Make a Plot
New
Philadelphia,
Source on github
Report a bug
Andy Mastbaum
The University of Pennsylvania

Shift Reports

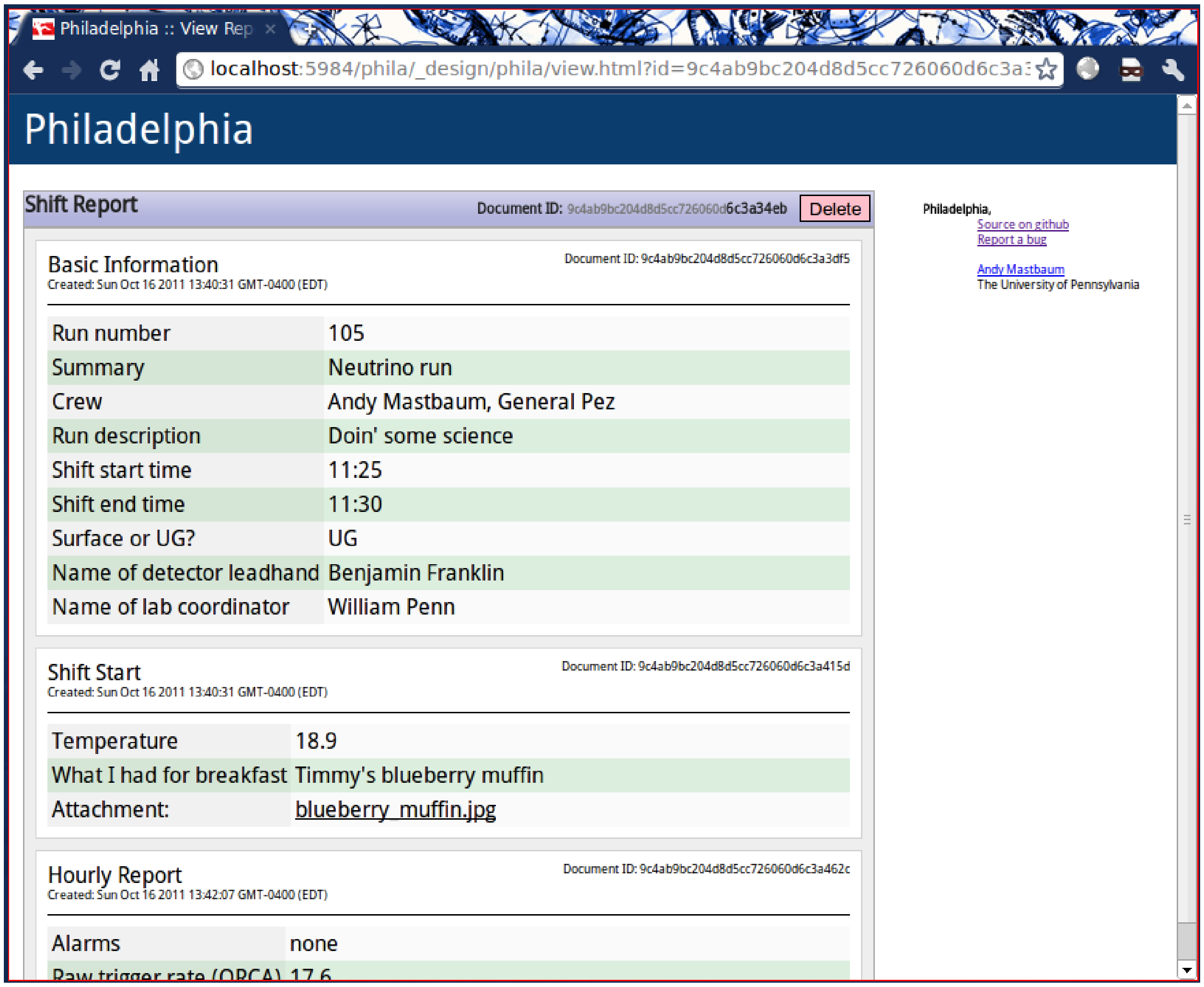
Report ID	Time Created	Run Number	Summary	Crew
6c3e53aa	2011-10-16 13:48:20	103	N16 Calibration	Captain Neutrino
6c3e4263	2011-10-16 13:47:55	102	Neutrino run	General Pez
6c3e1f07	2011-10-16 13:47:24	101	N16 calibration	Andy Mastbaum, Captain Neutrino
6c3e06be	2011-10-16 13:46:53	100	Neutrino run	Andy Mastbaum

The index page provides a list of all past shift reports. By default, it is sorted by date (newest first), but clicking on the column headers allows sorting by anything.

Clicking on a report ID will take you to that shift report.

There are two buttons in the upper right of the report list: “Make a Plot” and “New.” Clicking “New” takes you to the report composition page.

2.2 The Report View Page


Philadelphia :: View Rep
localhost:5984/phila/_design/phila/view.html?id=9c4ab9bc204d8d5cc726060d6c3a
Delete

Philadelphia

Shift Report — Document ID: 9c4ab9bc204d8d5cc726060d6c3a34eb

Philadelphia,
Source on github
Report a bug

Andy Mastbaum
The University of Pennsylvania

Basic Information
Created: Sun Oct 16 2011 13:40:31 GMT-0400 (EDT)
Document ID: 9c4ab9bc204d8d5cc726060d6c3a3df5

Run number	105
Summary	Neutrino run
Crew	Andy Mastbaum, General Pez
Run description	Doin' some science
Shift start time	11:25
Shift end time	11:30
Surface or UG?	UG
Name of detector leadhand	Benjamin Franklin
Name of lab coordinator	William Penn

Shift Start
Created: Sun Oct 16 2011 13:40:31 GMT-0400 (EDT)
Document ID: 9c4ab9bc204d8d5cc726060d6c3a415d

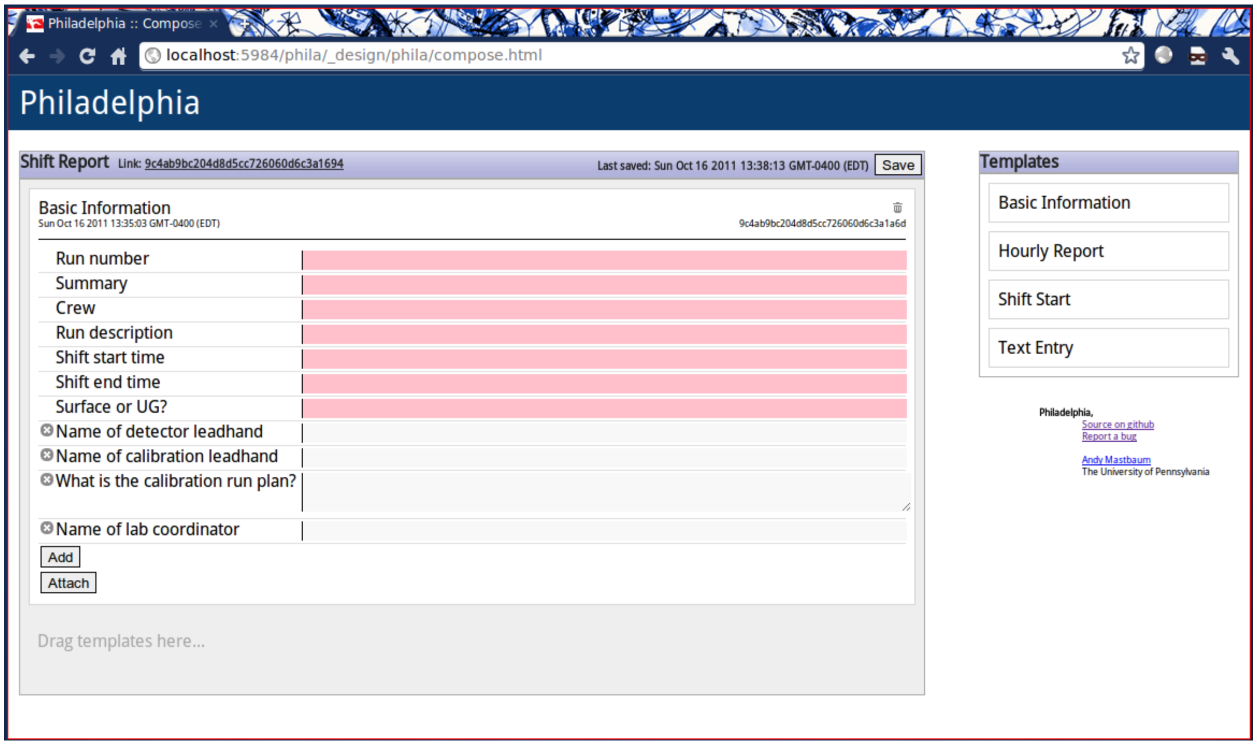
Temperature	18.9
What I had for breakfast	Timmy's blueberry muffin
Attachment:	blueberry_muffin.jpg

Hourly Report
Created: Sun Oct 16 2011 13:42:07 GMT-0400 (EDT)
Document ID: 9c4ab9bc204d8d5cc726060d6c3a462c

Alarms	none
Raw trigger rate (ORCA)	17.6

The report view page shows the contents of a shift report.

2.3 The Report Composition Page



Philadelphia :: Compose

localhost:5984/phila/_design/phila/compose.html

Philadelphia

Shift Report Link: 9c4ab9bc204d8d5cc726060d6c3a1694 Last saved: Sun Oct 16 2011 13:38:13 GMT-0400 (EDT) Save

Basic Information
Sun Oct 16 2011 13:35:03 GMT-0400 (EDT) 9c4ab9bc204d8d5cc726060d6c3a1a6d

Run number
Summary
Crew
Run description
Shift start time
Shift end time
Surface or UG?
Name of detector leadhand
Name of calibration leadhand
What is the calibration run plan?
Name of lab coordinator

Add
Attach

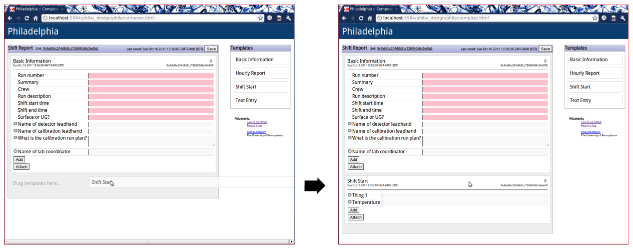
Drag templates here...

Templates
Basic Information
Hourly Report
Shift Start
Text Entry

Philadelphia,
Source on github
Report a bug
Andy Mastbaum
The University of Pennsylvania

2.3.1 Composing a Report

To compose a report, click and drag templates from the right into the report area ("Drag templates here..."). Once dropped, the templates can be re-arranged by dragging and dropping.



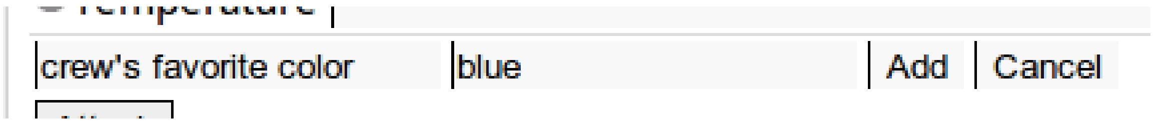
Fields marked as required by the shift report administrator appear with a red background when empty and cannot be removed.

2.3.2 Removing Fields

To remove a field or attachment, click the X to the left of the field name.

2.3.3 Adding New Fields

To add a new field, click "Add" at the bottom of the list of fields. Enter a field name and a value, then click "Add" or press Enter.



2.3.4 Adding Attachments

To attach a file to your shift report, click “Attach” at the bottom of the list of fields. Choose a file, then click “Upload.”

2.3.5 Saving

The report is automatically saved after every change and when you leave the page. You may also save at any time by clicking the “Save” button in the upper right corner of the report area.

2.3.6 Collaborative Editing

Shift reports can be edited by multiple users simultaneously. However, due to network latency editing the same block from multiple locations can lead to “choppy” behavior. Currently, newly-added blocks will not appear in others’ browsers until the page is reloaded.

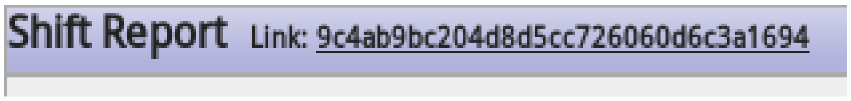
The report view page also updates in real time if the report is being edited.

2.3.7 Removing a Block

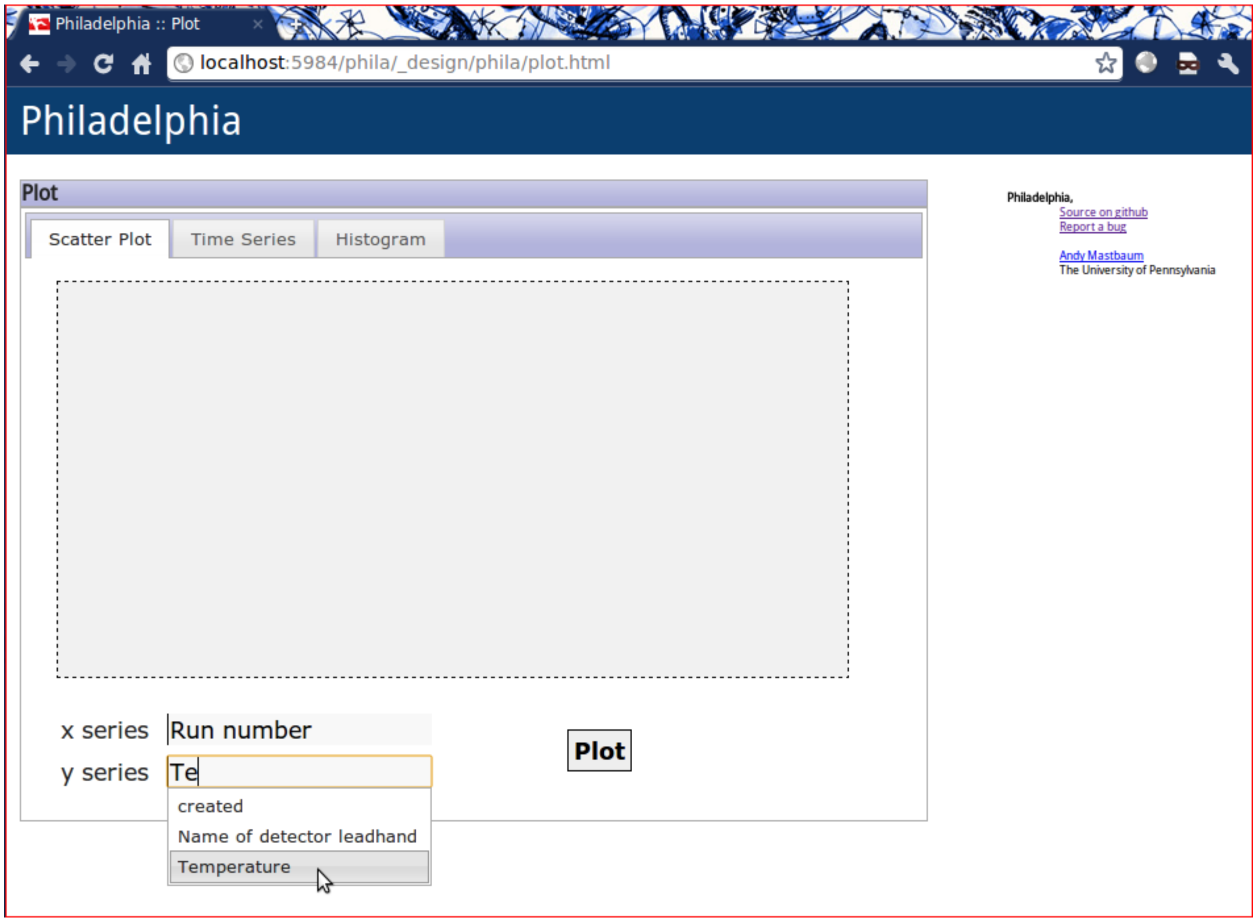
To remove a block from your report, click the trash can icon in the upper right corner of the block. Note that this operation is irreversible.

2.3.8 Remote Assistance

If you need help with a shift report, ask an expert. Send them the link at the top of the page, and they can see your report in real time as you modify it.



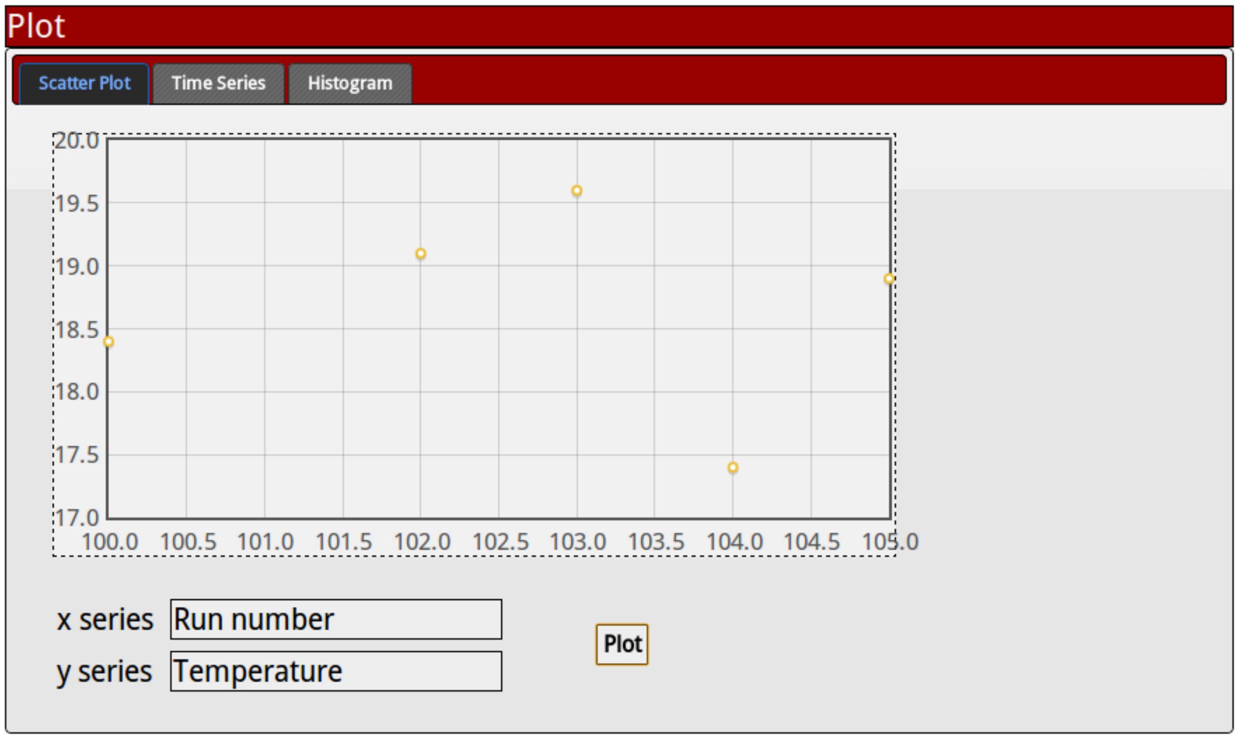
2.4 The Plotting Page



Philadelphia includes a built-in interface for plotting the data stored in shift reports. Currently, it supports scatter plots, time series, and histograms of numerical fields. All fields are plottable, including any added by users. Since field names are arbitary, all fields for entry of a series name autocomplete with valid field names.

2.4.1 Scatter Plots

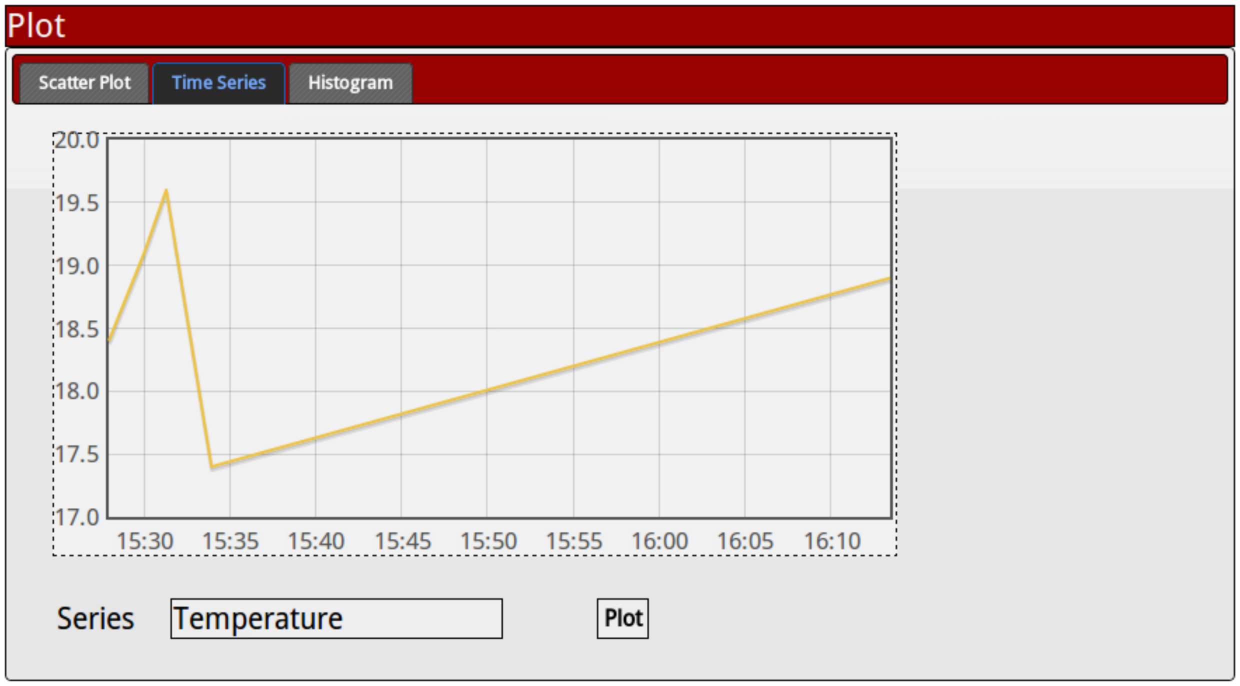
Enter an X series (e.g. "Run number") and a Y series (e.g. "Temperature") to create a scatter plot.



x series: Run number
y series: Temperature

2.4.2 Time Series

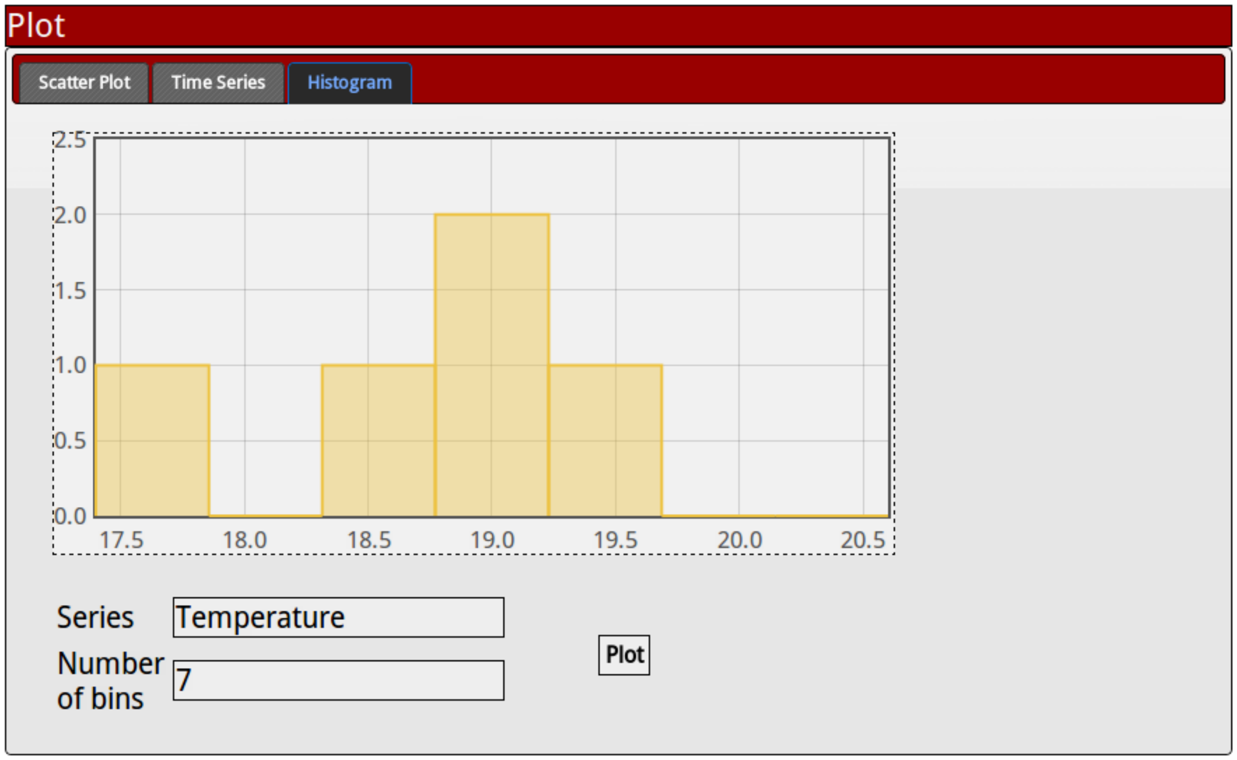
Enter a series name (e.g. "Temperature") to plot the history of that value over time.



Series: Temperature

2.4.3 Histograms

Enter a series name (e.g. “Temperature”) and the desired number of bins to create a histogram.



Series: Temperature
Number of bins: 7

CHAPTER 3

Administrator's Guide

3.1 Installation

Philadelphia is a lightweight CouchDB application developed in the egret framework. egret is a pure Python couch app development tool, and is included with Philadelphia for convenience.

3.1.1 Requirements

- Python >= 2.6
- CouchDB 1.1.0
- couchdb-lucene 0.9

3.1.2 Installing

Clone the Philadelphia repository from github:

```
$ git clone git://github.com/mastbaum/philadelphia.git
```

If you wish, customize application settings in *settings.json* and templates in *templates.json*.

Then, from within the *philadelphia* directory, run:

```
$ ./egret push http://your-server:port/database_name
$ ./egret pushdata http://your-server:port/database_name templates.json
```

Your Philadelphia installation should now be live at *http://your-server:port/database_name/_design/designname/index.html*, where *designname* is that in *settings.json* (default: *phila*).

3.1.3 Search Functionality

The search feature of Philadelphia relies on couchdb-lucene, an external Java indexer.

To set up lucene:

1. Follow the instructions in the couchdb-lucene readme
    (a) Download or git clone the couchdb-lucene source
    (b) Build with `mvn`. This will create a tarball in `target/`

   (c) Untar the tarball where the lucene server binary should live

2. If you are using CouchDB's HTTP authentication (i.e. you set `require_valid_user=true` in your CouchDB configuration), change the last lines of `config/couchdb-lucene.ini` to something like:

   ```
   [local]
   url = http://USERNAME:PASSWORD@localhost:5984
   ```

3. Start the lucene server with `/PATH/TO/LUCENE/bin/run`. For production, this should be run as a daemon.
4. Add the following to the `[httpd_global_handlers]` section in CouchDB's `local.ini` and restart the CouchDB server:

   ```
   _fti = {couch_httpd_proxy, handle_proxy_req, <<"http://127.0.0.1:5985">>}
   ```

5. Push the lucene indexing design document to the Philadelphia database:

   ```
   $ egret pushdata http://localhost:5984/phila /PATH/TO/PHILA/lucene/lucene.json
   ```

3.1.4 Nice URLs

URLs can easily be rewritten to be more user-friendly. For example, in Apache:

```
ProxyRequests Off
<Proxy *>
       Order Allow,Deny
       Allow from all
</Proxy>

 RewriteEngine On
 RewriteOptions Inherit

 RewriteRule ^/_fti/(.+)$ http://127.0.0.1:5984/_fti/$1 [QSA,P]
 RewriteRule ^/_uuids(.*) http://127.0.0.1:5984/_uuids [QSA,P]
 RewriteRule ^/_utils/script/(.*) http://127.0.0.1:5984/_utils/script/$1 [QSA,P]

 RewriteRule ^/philadelphia$ /philadelphia/ [R]
 RewriteRule ^/philadelphia/$ http://[INSERT YOUR FQDN]/philadelphia/index.html?user=%{LA-U:REMOTE_US

 RewriteRule ^/philadelphia/(.+)$ http://127.0.0.1:5984/phila/_design/phila/$1?user=%{LA-U:REMOTE_USE

 RewriteRule ^/phila/(.+)$ http://127.0.0.1:5984/phila/$1?user=%{LA-U:REMOTE_USER} [QSA,P]
 RewriteRule ^/phila/_changes(.+)$ http://127.0.0.1:5984/phila/_changes/$1?user=%{LA-U:REMOTE_USER}
 RewriteRule ^/phila/(.*) http://127.0.0.1:5984/phila?user=%{LA-U:REMOTE_USER} [QSA,P]
```

This will put everything at `http://server/philadelphia`.

There are only a few parameters that get passed around and it all happens via the query string, so even prettier URLs with no `.html` are possible too. Specifically, if `edit.html` receives an `id` it will open the report for editing (without an `id` it starts a new report), and `results.html` expects a search query `q=key:value`.

3.2 Report Templates

Although users may add and remove fields as they wish, all "standard" fields should be defined in a template. These are the fields the user sees when they first drop a template into their report.

Templates are defined in JSON in *templates.json*, in the root directory of your Philadelphia installation, with the following structure:

```
[
    {
        "_id": "template_name",
        "type": "template",
        "default": boolean,
        "fields": [
            {
                "name": "field 1 name",
                "type": "field 1 type",
                "attrib": "field 1 attributes",
                "required": boolean
            },
            {
                "name": "field 2 name",
                "type": "field 2 type",
                "attrib": "field 2 attributes",
                "required": boolean
            }
        ]
    }
]
```

After modifying *templates.json*, you must push the changes to the server with *./egret pushdata* as shown in the installation guide.

3.2.1 Field Types

The value of *"type"* must be an HTML form input type. Currently, Philadelphia supports *text*, *textarea*, and *checkbox*.

3.2.2 Field Attributes

Attributes are optional, and any provided will be passed along directly to the HTML input element. This is useful, for example, in fixing the size of a *textarea*:

```
{
    "name": "terminal contents",
    "type": "textarea",
    "attrib": "rows=80 cols=25"
}
```

3.2.3 Default Templates

If a template has the optional key *"default"* set to *true*, an instance of the template is automatically placed at the top of new shift reports.

3.2.4 Required Fields

Fields with the optional key *"required"* set to *true* cannot be deleted from the template. They also appear red if null.

CHAPTER 4

Indices and tables

- *genindex*
- *modindex*
- *search*