Corona Users Manual

Tim Dawborn Raymes Khoury

Contents

Chapter 1

What is Corona?

Corona is a Wireless Sensor Network (see http://en.wikipedia.org/wiki/Wireless_Sensor_Networks) query processor. This means that a user can enter an SQL-like query on a desktop computer connected to a basestation to obtain sensor values from nodes in the network. For example, a user might enter a query like:

```
SELECT temp, light
WHERE temp > 25
```

to find out the temperature and light readings of all sensor nodes whose temperature readings are greater than 25 degrees Celsius. A table of results with two columns (one for temperature and one for light) will be returned. Each row of the column represents a reading from one node in the network.

1.1 Features of Corona

It has:

- A resource-aware tree-based networking protocol - sensor nodes form a tree network topology in a way that conserves battery life
- In-network data aggregation - query results are processed and aggregated in-network to reduce transmission size and conserve battery
- Time triggered query execution - sensor nodes are time-synchronised. A query that is to be executed multiple times will only be sent into the network once and subsequent executions will be time-scheduled
- Data compression - we compress data using zlib compression to reduce transmission size
- An RMI interface (API) for application programmers - we provide an API so that Java applications can easily utilise Corona, even over a network.
- Easily extendable (custom sensors, custom data types, etc) - Corona is designed with simplicity in mind

- User management - there are 2 user levels to control who has access to various features of Corona
- a GUI - we provide a default GUI to easily interact with the query engine, which includes a query builder

Chapter 2

Installing and running Corona

2.1 Setting Up The SunSPOT SDK

The first step to setting up Corona is to ensure the SunSPOT SDK is properly installed and configured. Instructions for doing this can be found in the documentation included with the SunSPOT SDK or at `http://www.sunspotworld.com/GettingStarted/`.

Importantly, the current version of Corona requires the **BLUE** version of the SunSPOT SDK. We suggest that users are comfortable with deploying and running the demonstration applications included with the SunSPOT SDK before attempting to use Corona.

2.2 Getting the Source Code

The source code can be obtained from the Corona website at `http://www.it.usyd.edu.au/~wsn/corona/`.

Extract the source code to a directory using your favourite software ensuring the directory structure is preserved. We will call this directory `corona`.

2.3 The Three Components of Corona

There are 3 components to the Corona application which must all be running to use the system. These are as follows:

- The `spot` application - This application runs on the SunSPOT nodes. It handles coordination of the nodes networking, operation of the sensors and everything else that is performed on the SunSPOT nodes.
- The `server` application - This application runs on a desktop computer which is attached to a SunSPOT basestation via USB (refer to SunSPOT documentation regarding the basestation). It handles communication with the SunSPOT node network, dissemination

of queries into the network and collection and persistence of results. It also allows multiple client applications to log in to execute queries and view results.

- The `gui` application - This application runs on a desktop computer which is connected to the `server` via a TCP network. It provides a visual method of executing queries and viewing results.

2.4 Running the `spot` application

To deploy and run the `spot` application, we first need to open up a terminal window. Next, change into the directory of the `spot` module of Corona.

```
$ cd corona/spot
```

Now we want to clear up any old build files to ensure that we are compiling from scratch.

```
$ ant clean
```

Ensure the SunSPOTs you wish to deploy to are connected via USB. Compile, deploy and run the application on each SunSPOT with the following commands (you may need to choose which SPOT to deploy to if you have multiple connected at the same time):

```
$ ant deploy run
```

2.5 Running the `server` application

Before running the `server` component of the system, you *must* have deployed the application to at least one spot using the previous steps (or alternately run the `jar-app` command in the `spot` directory). Relocate into the `server` sub directory.

```
$ cd corona/server
```

Clean up any old build files

```
$ ant clean
```

Ensure that you have a basestation node connected to the desktop computer via USB. Compile and run the `server` application with the following command:

```
$ ant host-run
```

Note: Due to the dependencies between the `server` and the `spot` components, a local code repository exists (in the users home directory) for storing the most recent compiled version of each component.

2.6 Running the `gui` application

After the previous components are running, we can now run the GUI. Relocate into the `gui` sub directory.

```
$ cd corona/gui
```

Run the `gui`:

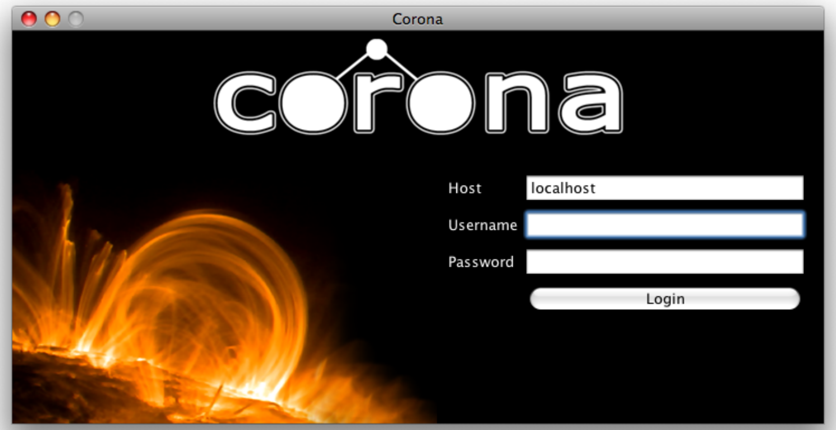
```
$ ant clean run
```

Now all components of the system should be up and running and ready for use

Chapter 3

Logging into Corona

When you have Corona up and running, the `gui` application will present you with a login screen as shown.



Host: localhost

In the **Host** field enter the name or IP address of the computer where the `server` program is running. If it is running on the same computer, the default value of `localhost` will work.

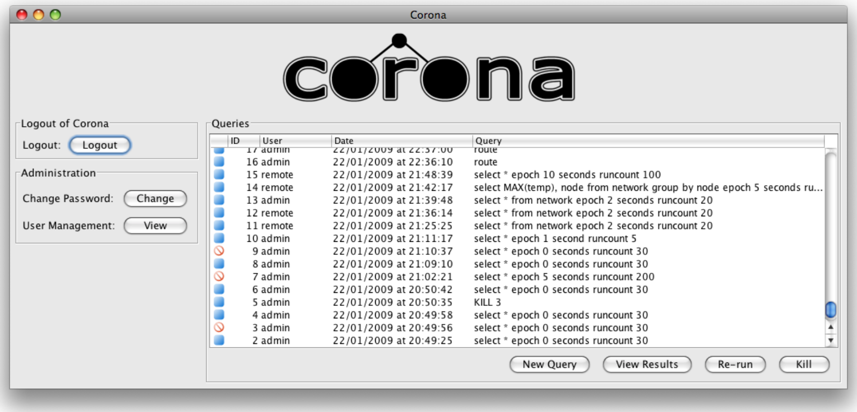
In the **Username** and **Password** fields, enter your username and password. The default values for a new install are:

- **Username:** admin
- **Password:** password

Chapter 4

Running your first query

Once you are logged into the system, you will be presented with the main screen.



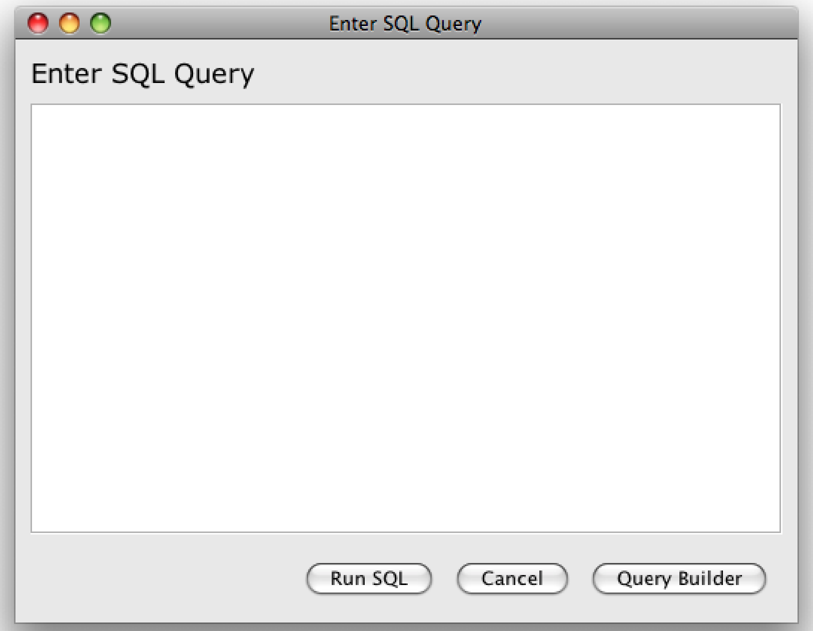
	ID	User	Date	Query
🟦	17	admin	22/01/2009 at 22:37:00	route
🟦	16	admin	22/01/2009 at 22:36:10	route
🟦	15	remote	22/01/2009 at 21:48:39	select * epoch 10 seconds runcount 100
🟦	14	remote	22/01/2009 at 21:42:17	select MAX(temp), node from network group by node epoch 5 seconds ru...
🟦	13	admin	22/01/2009 at 21:39:48	select * from network epoch 2 seconds runcount 20
🟦	12	remote	22/01/2009 at 21:36:14	select * from network epoch 2 seconds runcount 20
🟦	11	remote	22/01/2009 at 21:25:25	select * from network epoch 2 seconds runcount 20
🟦	10	admin	22/01/2009 at 21:11:17	select * epoch 1 second runcount 5
🚫	9	admin	22/01/2009 at 21:10:37	select * epoch 0 seconds runcount 30
🟦	8	admin	22/01/2009 at 21:09:10	select * epoch 0 seconds runcount 30
🚫	7	admin	22/01/2009 at 21:02:21	select * epoch 5 seconds runcount 200
🟦	6	admin	22/01/2009 at 20:50:42	select * epoch 0 seconds runcount 30
🟦	5	admin	22/01/2009 at 20:50:35	KILL 3
🟦	4	admin	22/01/2009 at 20:49:58	select * epoch 0 seconds runcount 30
🚫	3	admin	22/01/2009 at 20:49:56	select * epoch 0 seconds runcount 30
🟦	2	admin	22/01/2009 at 20:49:25	select * epoch 0 seconds runcount 30

This is the main window by which you interact with Corona. The table in the middle of the window shows a list of the queries that have been executed in the past. The *ID* column shows the unique ID of the query, the *User* column lists the username of the user who executed the query, the *Date* column lists the timestamp of when the query was first submitted to be executed, and the *Query* column shows the SQL query string. The first column with the icons shows the "status" of the query. A green shape means that the query is still executing, and blue shape means the query has completed executing, and a red cross shape indicates that the query has been killed.

The buttons on the left hand side of the main window allow you to perform more administrative functionality such as logging out, and changing your password.

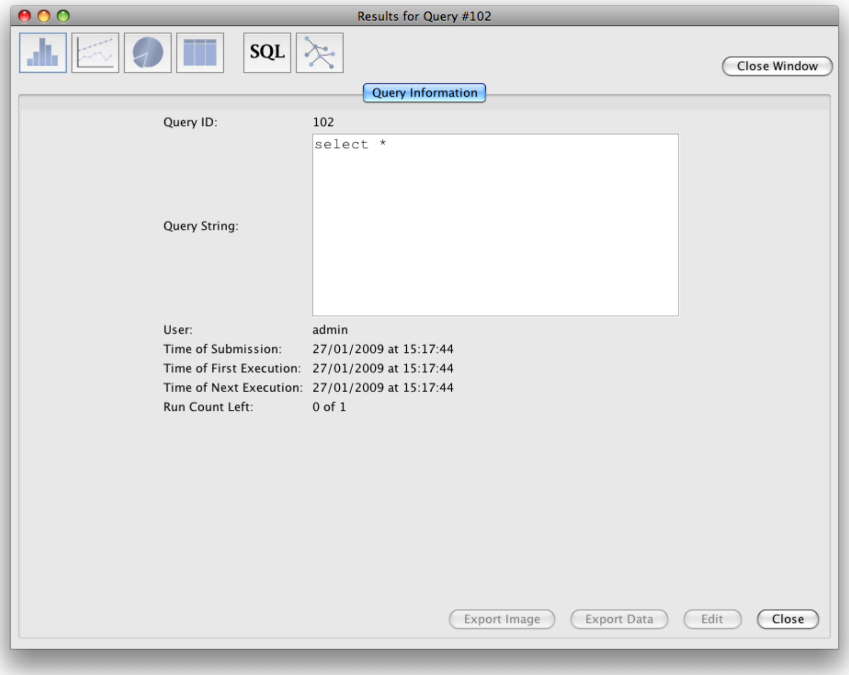
The buttons below the main table allow you to interact with the queries listed in the table. From here you can kill a query, re-run a query, view the results of a query, or alternatively create a new query. Lets create a new query. To do this, click on the **New Query** button underneath the main table and upon doing so, a new dialogue box will appear.

Using this "Enter SQL Query" window, you either have the option to enter the query using a simple SQL query language or using the query builder. The query builder is a drag-and-drop environment which allows you to formulate your queries visually without having to know the exact syntax of the SQL query language. The syntax of the SQL query language is discussed



in the section Available Query Types. For now, we will enter a simple query which will query every node in the network for all their sensor values. In the window enter the following query, and then click on the **Run SQL** button.

```
SELECT *
```



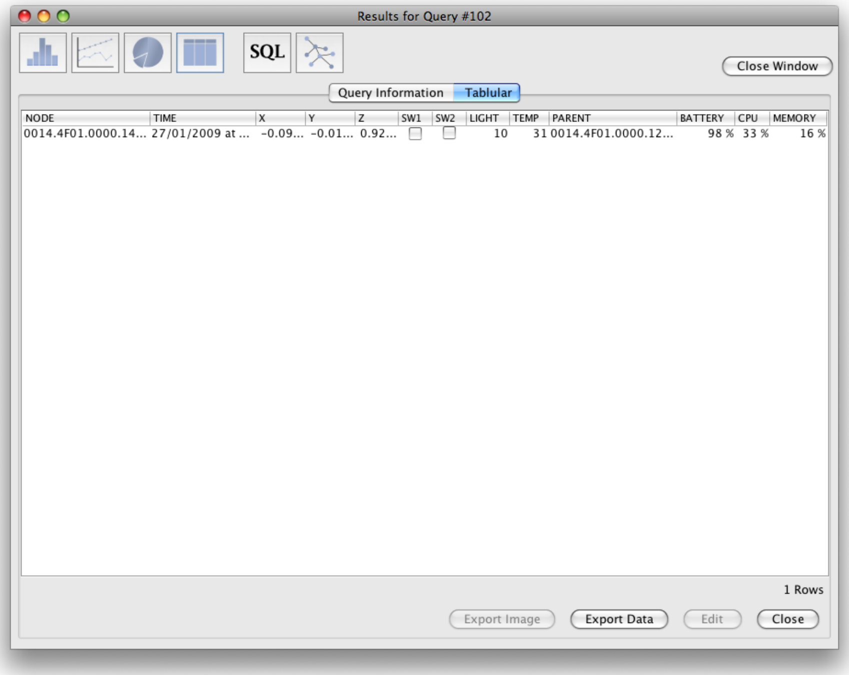
Query ID: 102

Query String:
```
select *
```

User: admin
Time of Submission: 27/01/2009 at 15:17:44
Time of First Execution: 27/01/2009 at 15:17:44
Time of Next Execution: 27/01/2009 at 15:17:44
Run Count Left: 0 of 1

The dialogue box will disappear and you will see a new row appear at the top of the Query table with the query you just executed. The icon in the left-most column should initially be green (which means that the query is in progress) but should quickly turn blue (meaning the query is complete). Double-click the row of the table that corresponds to this query to view the results of the query.

You will initially be presented with a new window containing a tab with the details of the query you just executed. Click the table icon at the top of the window and you will be presented with the results of the query in tabular form.



NODE	TIME	X	Y	Z	SW1	SW2	LIGHT	TEMP	PARENT	BATTERY	CPU	MEMORY
0014.4F01.0000.14...	27/01/2009 at ...	-0.09...	-0.01...	0.92...	☐	☐	10	31	0014.4F01.0000.12...	98 %	33 %	16 %

1 Rows

Chapter 5

Available query types in Corona

This page describes the types of queries that can be executed in Corona and the syntax of the queries to execute them.

Corona uses an SQL-like query language similar to that used in other WSN's.

Note that this is not intended to provide the full grammar of the system but only to aid in understanding the query syntax.

- SELECT queries
- KILL queries
- SYNC queries
- ROUTE queries
- SET queries

5.1 SELECT queries

5.1.1 Syntax

```
query ::=
"SELECT" <attributes>
["FROM" "SENSORS"]
["WHERE" <condition>]
["GROUP BY" <attributes>]
["HAVING" <condition>]
["START" ("IN" <relative_time> | "AT" <absolute_time> ["ON DATE" <date>])]
["EPOCH" <relative_time>]
["RUNCOUNT" (<integer> | "FOREVER")]

attributes  ::= "*" | <attributes2>
attributes2 ::= <attribute> ["," <attributes2>]

condition  ::= <condition2> <comparator> <condition2> [("AND" | "OR") <condition>]
condition2 ::= (<attribute> | <value>) [<operator> <condition2>]
comparator ::= "<" | "<=" | "!=" | "==" | ">=" | ">"
operator   ::= "+" | "-" | "*" | "/"
```

```
relative_time ::= [<integer> "hour"["s"]] [<integer> "minute"["s"]]  [<integer> "second"["s"]]

absolute_time ::= <integer> ":" <integer> ":" <integer>

date ::= <integer> "." <integer> "." <integer>

value   ::= <integer> | <float> | <node_id>
integer ::= ("0".."9") [<integer>]
float   ::= <integer> "." <integer>
node_id ::= <hex4> "." <hex4> "." <hex4> "." <hex4>
hex4    ::= <hex> <hex> <hex> <hex>
hex     ::= ("0".."9") | ("A".."F")
```

5.1.2 Description

This query type used to gather sensor information from nodes. An understanding of SQL would be helpful in producing queries for the system.

The components of the query are as follows:

Clause	**Description**	**Default value**
`SELECT`	Specifies which attributes/aggregates are required	(Required)
`WHERE`	The conditions to be applied to data prior to aggregation	(No Conditions)
`GROUP BY`	Specifies how aggregate data should be grouped	(All data grouped together)
`HAVING`	The conditions to be applied to data after aggregation	(No Conditions)
`START IN`	The relative delay before the query is executed	0 seconds
`START AT`	The time at which the query is to be executed. Format `hours:minutes:seconds`	now
`ON DATE`	The date at which the query is to be executed. Format `day.month.year`	today
`EPOCH`	The time between subsequent executions of the query	5 seconds
`RUNCOUNT`	The number of times to execute the query	1

The items you can specify in the `SELECT` clause are either "`*`", which implies all, *OR* any number of the following attributes separated by commas:

Attribute	**Description**
`node`	The IEEE address of the node
`time`	The timestamp that the sensor readings were taken at
`x`	The value of the x-axis accelerometer
`y`	The value of the y-axis accelerometer
`z`	The value of the z-axis accelerometer
`sw1`	Whether or not the first switch button is pressed down or not
`sw2`	Whether or not the second switch button is pressed down or not
`light`	The reading of the light sensor
`temp`	The temperature in degrees Celsius
`parent`	The IEEE address of the nodes networking parent in the network
`battery`	How much battery power is left as a percentage
`cpu`	The estimated usage of the CPU as a percentage
`memory`	How much memory (RAM) is in use currently as a percentage

Each of these attributes can have an aggregation modifier also. The aggregates are often used in conjunction with the `HAVING` and/or `GROUP BY` clauses as is the case in most SQL semantics. The available aggregates are:

Aggregate	**Description**
`MIN`	The minimum value
`MAX`	The maximum value
`SUM`	The sum of all the values
`AVG`	The average of all the values
`COUNT`	The number of values

5.1.3 Examples

To select every attribute from the sensors:

```
SELECT *
```

To select every attribute from the sensors every 10 seconds, 30 times:

```
SELECT *
EPOCH 10 seconds
RUNCOUNT 30
```

To select the node ID, the time, and the light from all the nodes whose temperature is greater than 15 degrees Celsius, once every second for a minute, starting in 5 minutes time:

```
SELECT node , time , light
WHERE temp > 15
START IN 5 minutes
EPOCH 1 second
```

```
RUNCOUNT 60
```

To select the maximum and minimum light readings per temperate group in the environment, once every 5 minutes for an hour:

```
SELECT MIN(light), MAX(light)
GROUP BY temp
EPOCH 5 minutes
RUNCOUNT 12
```

To see how many nodes have a critical battery level (less than 10%):

```
SELECT COUNT(node)
WHERE battery < 10
```

To see which nodes have 5 or more networking children:

```
SELECT parent
GROUP BY parent
HAVING COUNT(*) >= 5
```

5.2 KILL queries

5.2.1 Syntax

```
query ::= "KILL" <query id>
```

5.2.2 Description

The `KILL` query allows you to terminate a currently executing query that exists in the SunSPOT network. Executing a `KILL` query will cause all nodes to forget any future executions of the task specified. Killing a query which is not currently executing, or does not exist, causes nothing to happen.

5.2.3 Examples

To kill the query whose query ID number was 42, you would execute the query

```
KILL 42
```

5.3 SYNC queries

5.3.1 Syntax

```
query ::= "SYNC"
```

5.3.2 Description

The SYNC query type requests that the entire network resynchronise its time with the server (basestation). This resynchronising is done periodically (see the SYNC_EPOCH variable for the SET query), but the use of this SYNC query forces a resync to happen right now.

5.3.3 Examples

To force a resync of the network:

```
SYNC
```

5.4 ROUTE queries

5.4.1 Syntax

```
query ::= "ROUTE"
```

5.4.2 Description

The ROUTE query type requests that the entire network routing tree be reformed. This process takes sometime to complete, up to around 10 seconds on a medium to large sized network. While the network is re-routing, results which are being generated from currently executing queries will be lost.

5.4.3 Examples

To force a reroute of the network:

```
ROUTE
```

5.5 SET queries

5.5.1 Syntax

```
query ::= "SET" <property> "=" <integer>
```

5.5.2 Description

These types of queries are used to update constants in the Corona network. The properties that are changeable are listed in the table below:

Property	**Unit**	**Default Value**	**Description**
`SYNC_EPOCH`	Milliseconds	900000 (15 minutes)	The time period between when the network automatically resynchronises the time on all the nodes
`MONITORING_EPOCH`	Milliseconds	300000 (5 minutes)	The time period between when heartbeat messages are sent out from each node. The heartbeat messages are used to automatically detect if nodes have dropped out of the network
`REROUTE_EPOCH`	Milliseconds	2000000 (30 minutes aprox)	The time period between when the network will automatically reroute itself
`INTERCLUSTER_POWER`	Decibels	2	The power level the radio uses when sending data to nodes in other clusters. This should generally be 3 times greater than the intra-cluster equivalent
`INTRACLUSTER_POWER`	Decibels	-22	The power level the radio uses when sending data to nodes inside the same cluster

Note that queries of this type do not return any values.

5.5.3 Examples

To update the `SYNC_EPOCH` variable to be once every hour, the query would be

```
SET SYNC_EPOCH = 3600000
```

as 3600000 = 60 (minutes) * 60 (seconds) * 1000 (milliseconds).

Chapter 6

Result windows

The Corona GUI provides a number of different ways of viewing the data returned by an executed query. After you have fired off a query into the network, you can open up the results viewer window by any of the following:

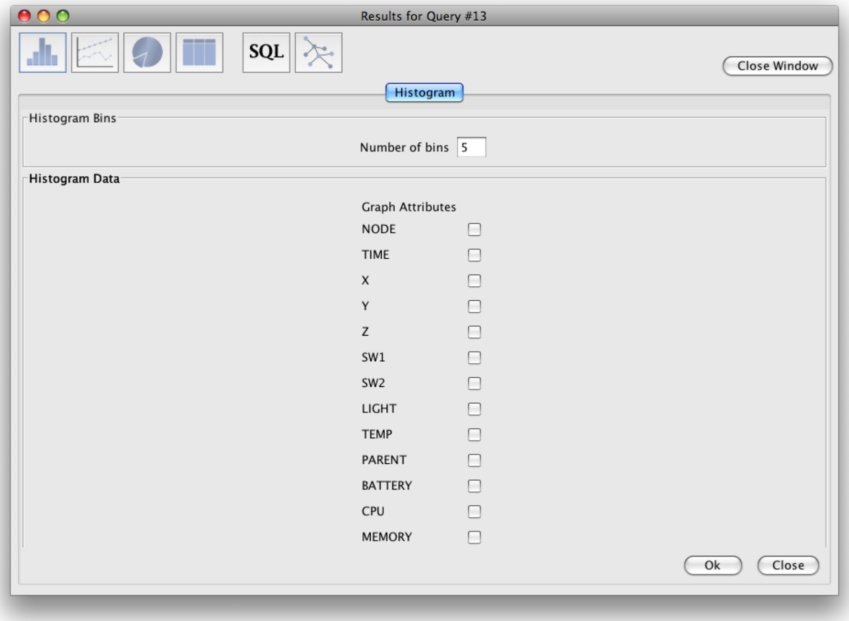
- double clicking on the query in the table of queries
- selecting the query in the table of queries, and pressing the "View Results" button at the bottom of the window
- selecting the query in the table of queries, right clicking it, and selecting the "View Results" option that drops down

Upon doing this, the results viewing window will open and display to you some information about the query. At the top of the results viewing window there a number of icons; each one of these icons represents a different way that the results of the query can be shown to you. These are:

- Histogram View
- Line Graph View
- Pie Chart View
- Table View
- Query Information View
- Network Tree View

6.1 Histogram

To open an instance of the histogram mode, click on the histogram icon from the top menu (the first icon from the left).



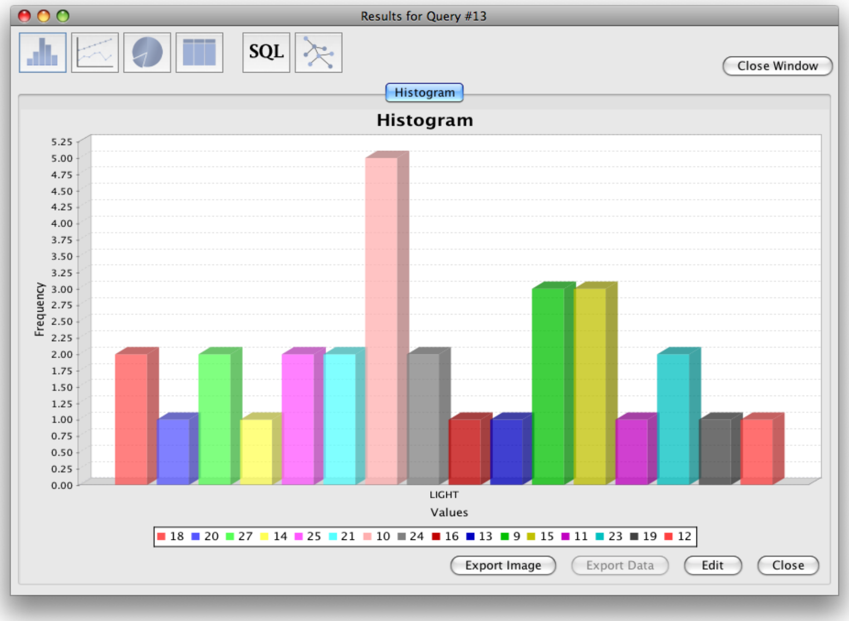
Histogram Bins

Number of bins 5

Histogram Data

Graph Attributes	
NODE	☐
TIME	☐
X	☐
Y	☐
Z	☐
SW1	☐
SW2	☐
LIGHT	☐
TEMP	☐
PARENT	☐
BATTERY	☐
CPU	☐
MEMORY	☐

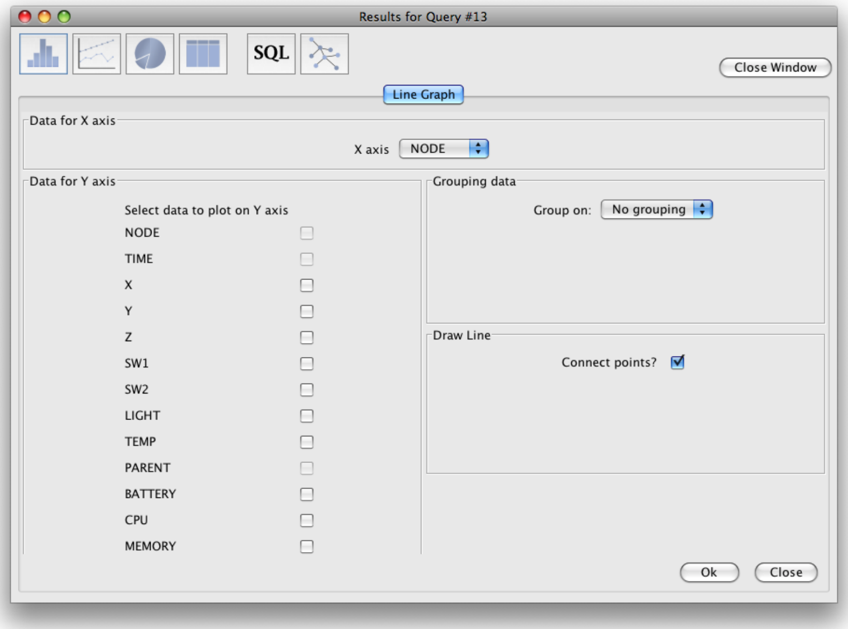
The histogram mode allows you to visualise the results of one or more attributes in the query, and see the relative frequencies of each value that attribute took in the set of results. When you first open up the histogram mode, you will be presented with a screen similar to the screenshot shown below. There will be a list of all the attributes that you requested in the query, as well as a place to enter the number of "bins" you want to have. A bin is a group of data that is considered to be the same in a histogram. If you have 10 distinct values in your results but you only request 5 bins to be visualised, this means that the first two distinct values will be combined into one, and the relative percentage for this combined value is calculated.



One you have selected your attribute to graph, as well as entered your desired number of bins, click on the "Ok" button at the bottom of the tab and your histogram will be presented to you. On the "x-axis", there will be a number of coloured dots, one for each bin. The legend at the base of the tab shows what each of these colours means in terms of the values they represent. The “y-axis” shows the relative frequency. So if a bar reads 45 on the y-axis, this means that 45% of the values that were returned from this query had the value co responding to the data in that bin.

6.2 Line Graph

To open an instance of the line graph mode, click on the line graph icon from the top menu (the second icon from the left).



Data for X axis

X axis: NODE

Data for Y axis

Select data to plot on Y axis

NODE	☐
TIME	☐
X	☐
Y	☐
Z	☐
SW1	☐
SW2	☐
LIGHT	☐
TEMP	☐
PARENT	☐
BATTERY	☐
CPU	☐
MEMORY	☐

Grouping data

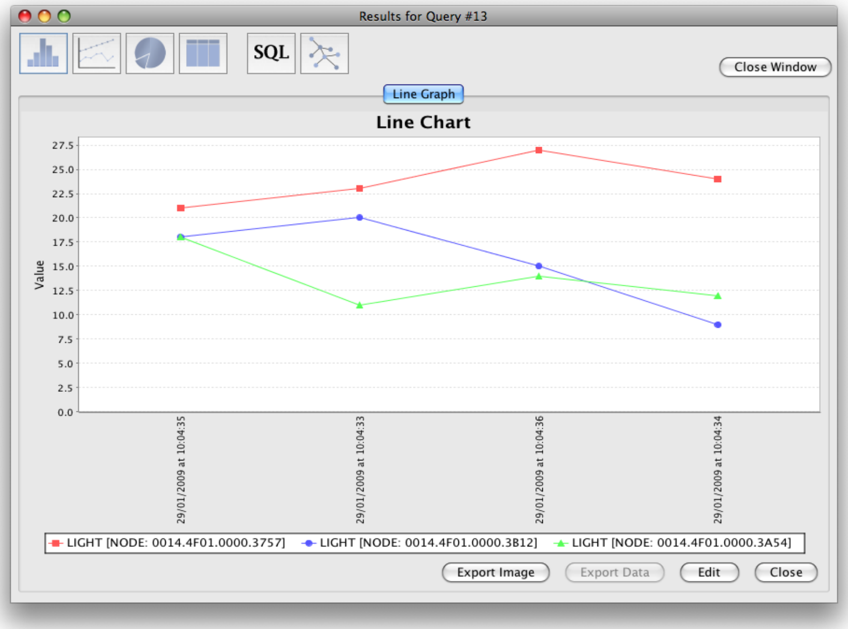
Group on: No grouping

Draw Line

Connect points? ✓

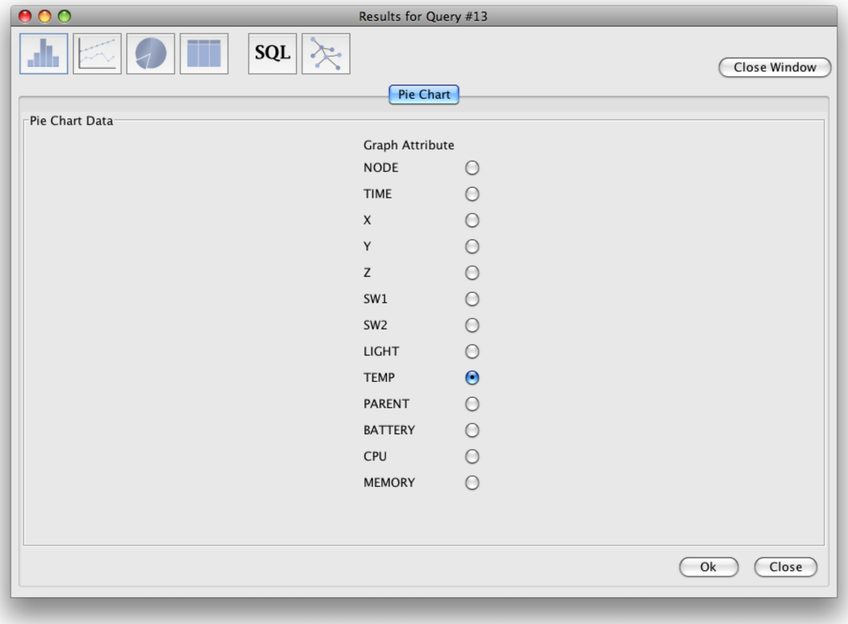
The line graph mode is arguably the most complex visualisation mode the Corona GUI provides. It allows you to plot one attribute of data against one or more other attributes of data, with the option of grouping by another attribute of data. When you click on the line graph icon, you are presented with a screen something similar to what is shown below, with all of the attributes you requested in the query listed. The example query was a “`SELECT *`” query, so we have all of the attributes available to us.

Lets explore this visualisation by example. Say we want to view how the light changed over time in our network while the query was active, and we want to group the data by node, so that we can see how the light changes on each node over time, rather than overall at each point in time. To do this, from the “x-axis” dropdown menu we select the “time” attribute as that is what we want to plot against. From the “y-axis” menu, we tick the “light” attribute, as we want to plot light against time. Lastly, seeing as we would like to see the change in light per node, we select “node” from the “group by” dropdown menu on the right hand side of the tab. Once this is all done, press the “Ok” button at the bottom of the tab to generate the graph for the options you have chosen. For our example query, the line graph generated is shown below.



6.3 Pie Chart

To open an instance of the pie chart mode, click on the pie chart icon from the top menu (the third icon from the left).

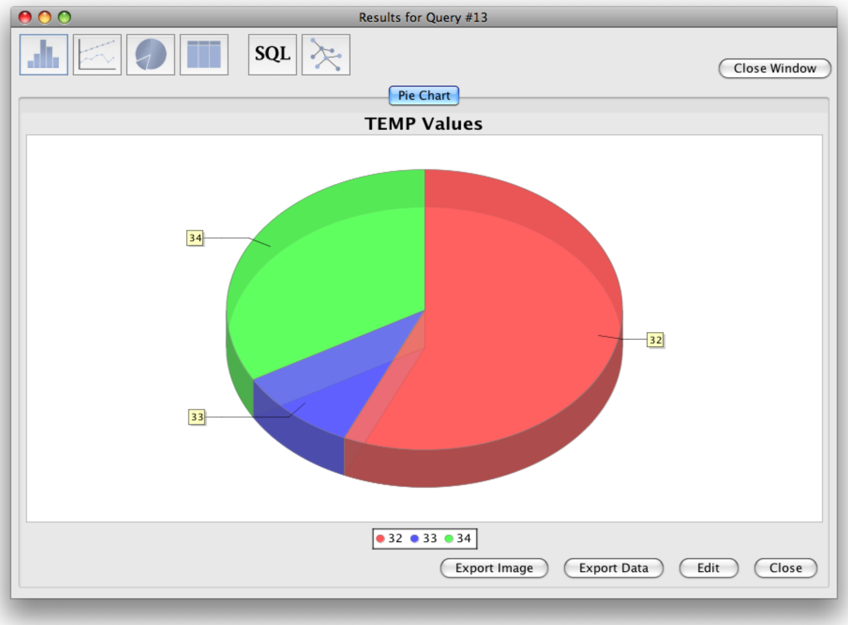


Pie Chart Data

Graph Attribute

NODE	○
TIME	○
X	○
Y	○
Z	○
SW1	○
SW2	○
LIGHT	○
TEMP	●
PARENT	○
BATTERY	○
CPU	○
MEMORY	○

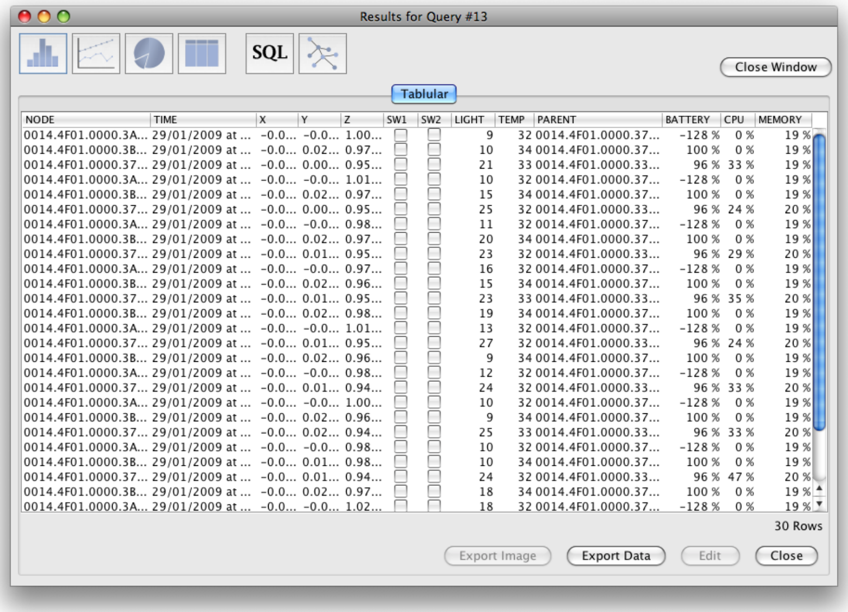
The pie chart visualisation is very similar to the histogram visualisation in the sense it lets you visualise the relative frequencies of data. The pie chart view restricts you to one attribute however, rather than one or more, and it presents these frequencies as portions of a single circle, rather than as a series of columns on a xy axis. When you open up a new instance of the pie chart view, you will be presented with something similar to what is shown below, with only the attributes you requested in your query being available to you to visualise. Select which attribute you would like to graph, and press the "Ok" button at the base of the tab.



Lets view the temperature in the network this time rather than the light. Check the "temp" option in the list and upon pressing the "Ok" button, we are presented with something similar to what is shown below. The distinct result values are coloured as portions of the pie.

6.4 Table

To open an instance of the table mode, click on the table icon from the top menu (the fourth icon from the left).



NODE	TIME	X	Y	Z	SW1	SW2	LIGHT	TEMP	PARENT	BATTERY	CPU	MEMORY
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	1.00...	☐	☐	9	32	0014.4F01.0000.37...	-128 %	0 %	19 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.02...	0.97...	☐	☐	10	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.37...	29/01/2009 at ...	-0.0...	0.00...	0.95...	☐	☐	21	33	0014.4F01.0000.33...	96 %	33 %	19 %
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	1.01...	☐	☐	10	32	0014.4F01.0000.37...	-128 %	0 %	19 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.02...	0.97...	☐	☐	15	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.37...	29/01/2009 at ...	-0.0...	0.00...	0.95...	☐	☐	25	32	0014.4F01.0000.33...	96 %	24 %	20 %
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	0.98...	☐	☐	11	32	0014.4F01.0000.37...	-128 %	0 %	19 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.02...	0.97...	☐	☐	20	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.37...	29/01/2009 at ...	-0.0...	0.01...	0.95...	☐	☐	23	32	0014.4F01.0000.33...	96 %	29 %	20 %
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	0.97...	☐	☐	16	32	0014.4F01.0000.37...	-128 %	0 %	19 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.02...	0.96...	☐	☐	15	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.37...	29/01/2009 at ...	-0.0...	0.01...	0.95...	☐	☐	23	33	0014.4F01.0000.33...	96 %	35 %	20 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.02...	0.98...	☐	☐	19	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	1.01...	☐	☐	13	32	0014.4F01.0000.37...	-128 %	0 %	19 %
0014.4F01.0000.37...	29/01/2009 at ...	-0.0...	0.01...	0.95...	☐	☐	27	32	0014.4F01.0000.33...	96 %	24 %	20 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.02...	0.96...	☐	☐	9	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	0.98...	☐	☐	12	32	0014.4F01.0000.37...	-128 %	0 %	19 %
0014.4F01.0000.37...	29/01/2009 at ...	-0.0...	0.01...	0.94...	☐	☐	24	32	0014.4F01.0000.33...	96 %	33 %	20 %
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	1.00...	☐	☐	10	32	0014.4F01.0000.37...	-128 %	0 %	19 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.02...	0.96...	☐	☐	9	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.37...	29/01/2009 at ...	-0.0...	0.02...	0.94...	☐	☐	25	33	0014.4F01.0000.33...	96 %	33 %	20 %
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	0.98...	☐	☐	10	32	0014.4F01.0000.37...	-128 %	0 %	19 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.01...	0.98...	☐	☐	10	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.37...	29/01/2009 at ...	-0.0...	0.01...	0.94...	☐	☐	24	32	0014.4F01.0000.33...	96 %	47 %	20 %
0014.4F01.0000.3B...	29/01/2009 at ...	-0.0...	0.02...	0.97...	☐	☐	18	34	0014.4F01.0000.37...	100 %	0 %	19 %
0014.4F01.0000.3A...	29/01/2009 at ...	-0.0...	-0.0...	1.02...	☐	☐	18	32	0014.4F01.0000.37...	-128 %	0 %	19 %

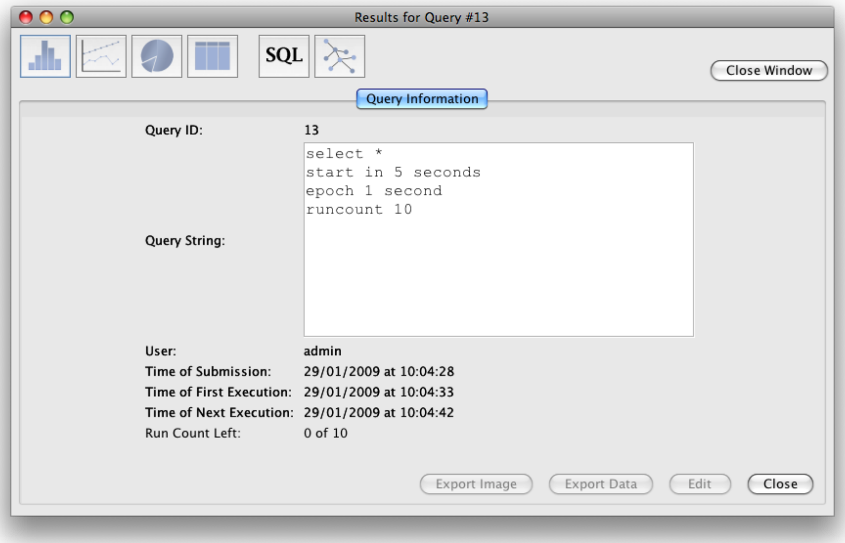
30 Rows

This mode is the mode most people would associate the viewing of database data with; all of the data displayed in a table format. Here, as the data comes back from the network, it is displayed in the table. The columns of this table are the attributes you specified in the query (in the case of "`SELECT *`" you will get all the attributes returned), and each row represents one epoch's worth of data from one SunSPOT (thus if you have n SunSPOTs and a m epoch query, the maximum number of rows you will get back is mn).

Clicking on any of the column names allows you to sort the data in the table by this column. Clicking on the column name once will sort it in ascending order, and then another click will toggle it to become sorted in descending order. Underneath and to the right the table there is some text displaying the total number of rows in this table of results.

6.5 Query Information

To open an instance of the query information mode, click on the "SQL" icon from the top menu (the second last icon). This is the mode that is first activated when you open a window to view the results of a query.



Query ID: 13

Query String:

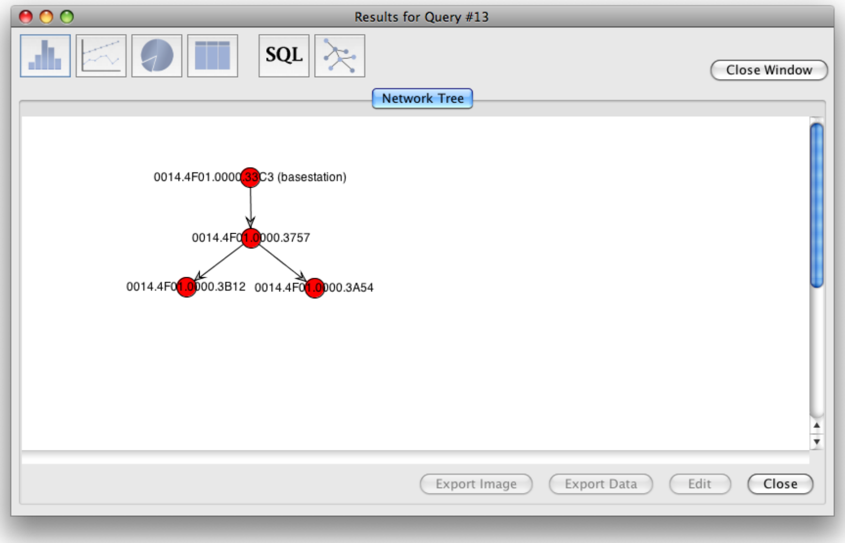
```
select *
start in 5 seconds
epoch 1 second
runcount 10
```

User: admin
Time of Submission: 29/01/2009 at 10:04:28
Time of First Execution: 29/01/2009 at 10:04:33
Time of Next Execution: 29/01/2009 at 10:04:42
Run Count Left: 0 of 10

The query information view allows you to view miscellaneous information about the queries execution. The **Query ID** is a unique identification number given to this query which can be used for tasks such as killing the query. The **Query String** field shows the SQL query that this set of results is generated for. The **User** field shows the username of the user who executed this query. **Time of Submission** displays the time (on the Corona server) at which the query was submitted to be executed. The **Time of First Execution** field shows the time that the first epoch of this query was executed, and the **Time of Next Execution** field shows the time that we expect the next epoch to be run. If this time is in the past then this implies that the query has finished executing all of its epochs. The **Run Count Left** field shows the number of epochs that this query has left to execute.

6.6 Network Tree

To open an instance of the network tree mode, click on the tree icon from the top menu (the last icon).



This mode provides an interactive way to view the network tree that the SunSPOTs have formed. Arrows point from parent nodes to their children. When Corona does its in-network aggregation, the data flow goes back up from children to their parents. You can use the mouse to drag and drop the nodes around on the graph, as well as use a scroll-mouse to zoom in and out. The use of the scroll-mouse when combined with modifier keys (shift, control, etc) provides additional navigational functionality such as panning.

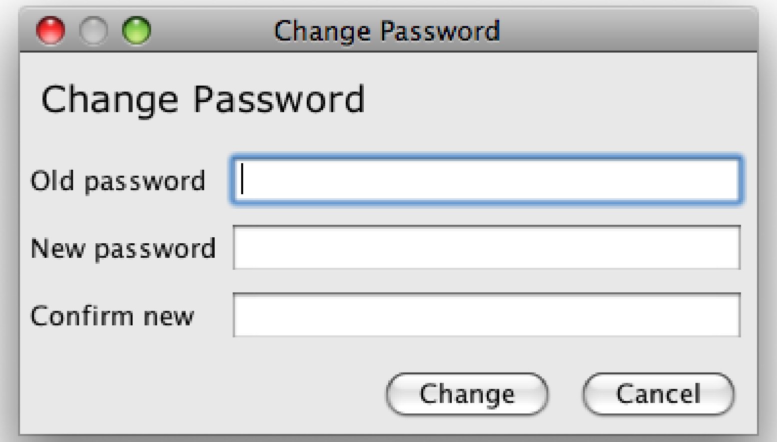
Chapter 7

User management in Corona

User management takes two main forms in Corona. All users are able to change their login password. Users with administrative privileges are able to modify properties of all the users in the system.

7.1 Changing Your Password

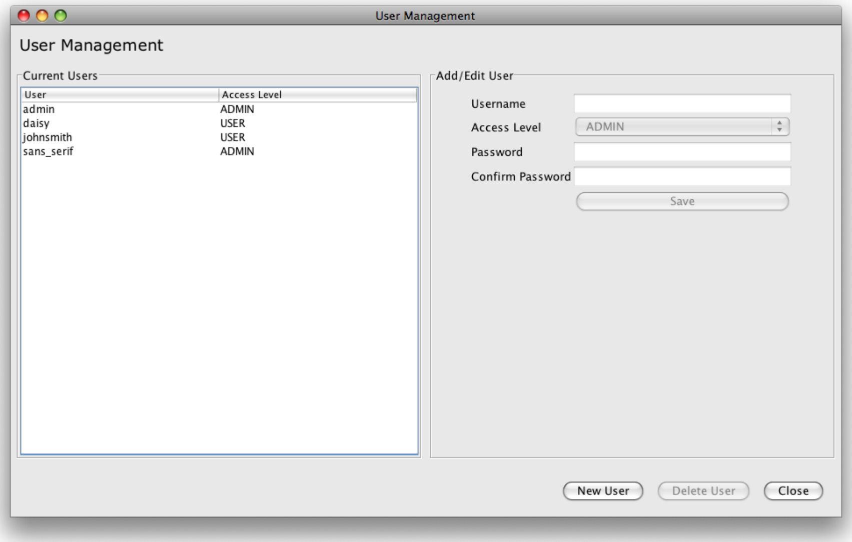
To change your Corona login password, click on the "Change Password" button on the left of the main window. This will bring up a new window which asks you for your old password, followed by your new password, and then a confirmation of your new password.



Once you have filled in the required details, clicking the "Ok" button will change your password. If you have somehow managed to forget your password, you need to get in contact with one of the Corona administrators for the server you are connecting to, and ask them to reset your password.

7.2 Administrative Features

Users with administrative privileges are able to create, modify, and delete users. To open up the user management window in Corona, click on the "User Management" button on the left hand side of the main window.



User Management

Current Users

User	Access Level
admin	ADMIN
daisy	USER
johnsmith	USER
sans_serif	ADMIN

Add/Edit User

Username	
Access Level	ADMIN
Password	
Confirm Password	

This window is split into two sections. The left hand side of the window lists all of the users in the system. Clicking on one of these users actives the section on the right hand side. The right hand side lets you edit the attributes of the currently selected user, or create a new user, which ever the case may be.

7.2.1 Creating a New User

To add a new user to the Corona server, click on the "New User" button at the bottom of the user management window. This then actives the right hand side of the window, allowing you to set the attributes of this new user. Once you have entered the users username, their privilege level, and their password, clicking on the "Save" button will add the user to the system. Upon a successful addition, the user should then appear in the list of users on the list hand side of the window.

7.2.2 Deleting an Existing User

To delete an existing user in the Corona server, select that user from the list of users on the left hand side of the window. This will active the editable fields on the right hand side of the window for this user. Click on the "Delete User" button underneath the editable fields. Once you confirm that you do want to delete the users account, the user will be removed from the server and they will disappear from the users list immediately.

7.2.3 Editing an Existing User

To edit an existing user in the Corona server, select that user from the list of users on the left hand side of the window. This will active the editable fields on the right hand side of the window for this user. After you have made your desired changes to the user in the editable fields, click on the "Save" button. Doing this will update the user in the server, and if successful, the changes will be reflected in the users list on the left hand side of the window immediately.