Database Deployment

1. Create database and table

Create a table on a specific database, the table schema can be found in **database.sql** file in **gpscomm.tar.gz** tarball. Follow this command to create a database named **gpscomm** and related tables with user **postgres**

```
$ psql -h localhost -U postgres -c “CREATE DATABASE gpscomm”
$ psql -h localhost -U postgres gpscomm < database.sql
```

*Please consider to create another user and grant any required permissions for each table, the user **postgres** is intended for administration and example only.*

2. Table schema

Table **gpsclientcfg** - Client configuration

```
 client_name | unicast_port | multicast_port | multicast_group | broadcast_port | packet_validation | location_writeival |  server_host  | server_ctlport | server_retryival
-------------+--------------+----------------+-----------------+----------------+-------------------+--------------------+---------------+----------------+------------------
 demo1       |      6000 |       6001 | 224.0.0.1     |            6002 | 1                 |            5000 | 127.0.0.1   |       5000 |              5000
 demo2       |      6003 |       6004 | 224.0.0.1     |            6005 | 1                 |            5000 | 192.168.0.2 |       5000 |              5000
 demo3       |      6007 |       6008 | 224.0.0.1     |            6009 | 1                 |            5000 | 192.168.2.2 |       5000 |              5000
(3 rows)
```

Column description:

Column Name	Type	Description
client_name	string	Client name
unicast_port	integer	Client unicast port
multicast_port	integer	Client Multicast port
multicast_group	string	Client multicast group address
broadcast_port	integer	Client broadcast port
packet_validation	integer	Whether to enable packet validation sent by server, value: 1 or 0
location_writeival	integer	Interval to write client location in mili-seconds
server_ip	string	Server hostname or IP address to send CTL message
server_ctlport	integer	Server port to send CTL message

server_retryival	integer	Timeout in mili-second to wait for TGR (trigger) message from server

Table **gpsdata** - GPS and event data

```
 uid  | client_name |  client_ip  |  sender_ip  | client_timestamp | client_lat | client_long | event_type
------+-------------+-------------+-------------+------------------+------------+-------------+------------
 4280 | demo2       | 192.168.0.2 | 192.168.0.2 |     1352037773 | 55.671385  | 12.518613   | 1
 4279 | demo2       |             |             |     1352037762 | 55.671367  | 12.518827   | 0
 4278 | demo3       | 192.168.2.2 | 192.168.2.2 |     1352037773 | 55.671385  | 12.518613   | 1
 4277 | demo3       |             |             |     1352037762 | 55.671367  | 12.518827   | 0
 4276 | demo3       | 192.168.2.2 | 192.168.2.2 |     1352037772 | 55.671377  | 12.518622   | 1
 4275 | demo3       |             |             |     1352036986 | 55.671190  | 12.521440   | 0
 4274 | demo2       | 192.168.0.2 |             |     1352037773 | 55.671385  | 12.518613   | 4
 4273 | demo3       | 192.168.2.2 |             |     1352037773 | 55.671385  | 12.518613   | 4
 4272 | demo2       | 192.168.0.2 | 192.168.0.2 |     1352037772 | 55.671377  | 12.518622   | 1
 4271 | demo2       |             |             |     1352037010 | 55.671105  | 12.521337   | 0
(10 rows)
```

Column description:

Column Name	Type	Description
uid	integer	Unique ID for each row
client_name	string	Name of the client
client_ip	string	Client IP address
sender_ip	string	Sender IP address (trigger)
client_timestamp	integer	Client GPS fix timestamp
client_lat	string	Client GPS fix latitude
client_long	string	Client GPS fix longitude
event_type	integer	Type of event, as follow: 1 = EVENT_UNICAST 2 = EVENT_BROADCAST 3 = EVENT_MULTICAST 4 = EVENT_ACK 7 = EVENT_ONLINE / READY 8 = EVENT_OFFLINE 9 = EVENT_TIMEOUT