GPS Server Manual

1. Installing Required Softwares

This manual is using **Ubuntu 12.04 LTS** Linux distribution and assuming build tools (GCC C compiler, make, etc.) has been previously installed.

1.1. PostgreSQL

Issue these following commands to install PostgreSQL related packages:

```
$ sudo apt-get install postgresql postgresql-client libpq-dev
$ sudo /etc/init.d/postgresql start
```

Refer to **Database Deployment** document to deploy related database tables.

2. Build, Configure and Run

The supplied gpsserver tarball already contain gpsserver binary file in order to just running, goto section **2.3 Running** to start gpsserver without building.

2.1 Build

Assuming build tools have been installed, issue these following command to build:

```
$ tar xzf gpscomm.tar.gz
$ cd gpscomm/gpsserver
$ make
```

If success gpsserver binary file name **gpsserver** will be created.

2.2. Configuration

Configuration can be changed by editing **gpsclient.conf** file, the following table describes each of configuration key.

Key name	Default value	Description
control-port	5000	Control port
unicast-enable	yes	Whether to enable unicast packet
unicast-port	6000	Unicast port
broadcast-enable	yes	Whether to enable broadcast packet
broadcast-port	6001	Broadcast port
multicast-enable	6002	Whether to enable multicast packet

multicast-group-addr	224.0.0.1	Multicast group address
clientport-enable	no	Whether to use client ports (unicast, multicast, broadcast) that sent by client, this is useful to run many clients on a single host (each client is able to run on different port for each packet)
packet-interval	3000	Interval of time (in milliseconds) to send message (trigger) for each enabled packet type
prune-interval	5000	Timeout of waiting for ACK reply after sending trigger message
db-host	127.0.0.1	Database host address, can be a hostname or IP address
db-port	5432	Database host port
db-name	db-name	Database name
db-user	db-user	Database username
db-passwd	db-passwd	Database username's password
db-table	gpsclientdata	Table to write for event
logfile-path	/tmp/gpsserver.log	Log file to write (only if "daemonize-enable" option is set to "yes")
daemonize-enable	no	Whether to daemonize process and run in background, all output will be written in "logfile-path" file

2.3. Running

To run gpsclient use the following command template:

```
$ ./gpsserver <configuration-file>
```

For example:

```
$ ./gpsserver gpsserver.conf
```

Or to run in background (without "daemonize-enable" option enabled) and redirect output to a file gpsclient.log:

```
$ ./gpsserver gpsserver.conf > gpsserver.log 2>&1&
```

Example output

```
$ ./gpsserver gpsserver.conf
[INFO] 140525.230957.54 control-port=5000 [config.c:44]
[INFO] 140525.230957.54 unicast-enable=yes unicast-port=6000 [config.c:46]
[INFO] 140525.230957.54 multicast-enable=no multicast-port=6001 [config.c:48]
[INFO] 140525.230957.54 broadcast-enable=no broadcast-port=6002 [config.c:50]
[INFO] 140525.230957.54 clientport-enable=yes [config.c:51]
[INFO] 140525.230957.54 packet-interval=3000ms [config.c:52]
[INFO] 140525.230957.54 prune-interval=5000ms [config.c:53]
[INFO] 140525.230957.54 db-host=127.0.0.1 db-port=5432 db-name=test db-user=postgres db-passwd=postgres
db-table=gpsdata [config.c:56]
[INFO] 140525.230957.54 logfile-path=/tmp/gpsserver.log [config.c:57]
[INFO] 140525.230957.54 pidfile-path=/tmp/gpsserver.pid [config.c:58]
[INFO] 140525.230957.54 daemonize-enable=no [config.c:59]
[INFO] 140525.230957.57 connected to database 127.0.0.1:5432 [server.c:497]
[INFO] 140525.230957.57 control socket was created on 0.0.0.0:5000 [server.c:503]
[INFO] 140525.230957.89 connection from addr=127.0.0.1 fd=6 [server.c:373]
[INFO] 140525.230957.89 recvd CTL msg code=CLIENT_ONLINE client='demo1' addr=127.0.0.1 fd=6
[server.c:184]
[INFO] 140525.230958.90 connection from addr=192.168.2.2 fd=6 [server.c:373]
[INFO] 140525.230958.90 recvd CTL msg code=CLIENT_ONLINE client='demo3' addr=192.168.2.2 fd=6
[server.c:184]
[INFO] 140525.230959.02 connection from addr=192.168.0.2 fd=6 [server.c:373]
[INFO] 140525.230959.02 recvd CTL msg code=CLIENT_ONLINE client='demo2' addr=192.168.0.2 fd=6
[server.c:184]
[INFO] 140525.231001.03 sent TGR msg type=ucast client='demo1' addr=127.0.0.1 fd=7 [server.c:276]
[INFO] 140525.231001.10 recvd ACK msg client='demo1' lat=55.670062 lon=12.520798 tsp=1352037212
addr=127.0.0.1 fd=7 [server.c:241]
[INFO] 140525.231002.11 sent TGR msg type=ucast client='demo3' addr=192.168.2.2 fd=8 [server.c:276]
[INFO] 140525.231002.11 sent TGR msg type=ucast client='demo2' addr=192.168.0.2 fd=9 [server.c:276]
[INFO] 140525.231002.20 recvd ACK msg client='demo2' lat=55.671377 lon=12.518622 tsp=1352037772
addr=192.168.0.2 fd=9 [server.c:241]
[INFO] 140525.231002.21 recvd ACK msg client='demo3' lat=55.671377 lon=12.518622 tsp=1352037772
addr=192.168.2.2 fd=8 [server.c:241]
[INFO] 140525.231003.21 connection from addr=127.0.0.1 fd=6 [server.c:373]
[INFO] 140525.231003.21 recvd CTL msg code=CLIENT_OFFLINE client='demo1' addr=127.0.0.1 fd=6
[server.c:184]
```

4. Problem and Solving

4.1 Examine Error and Warning Message

Some known error messages

Error message	Resolution
[ERROR] unable to open logfile	Can not access log file to write, check file's path and permission
[ERROR] unable to read config file	Configuration file was not found or could not be read, check if configuration file exists and has permission to read

[ERROR] unable to connect to database	Check whether PostgreSQL server is running and make sure database address, port, name, username and password has been configured correctly in configuration file. Examine gpsserver log (output) to see further error messages
[WARNING] invalid CTL msg hdr=XX addr=X.X.X.X	Client send an invalid CTL message
[WARNING] invalid ACK msg hdr=XX addr=X.X.X.X	Client send an invalid ACK message

4.2. Examining Database Record

Example command to view database record:

```
$ psql -h localhost -U postgres gpsdata -c "select * from gpsclientdata order by uid desc
limit 10"
```

```
uid  | client_name |  client_ip  |  sender_ip  | client_timestamp | client_lat | client_long | event_type
-----+-------------+-------------+-------------+------------------+------------+-------------+----------
 3636 | demo3       | 192.168.2.2 |             |      1352037681 | 55.670882  | 12.519515   | 4
 3635 | demo2       | 192.168.0.2 |             |      1352037681 | 55.670882  | 12.519515   | 4
 3634 | demo2       |             |             |      1352036901 | 55.671652  | 12.521612   | 0
 3633 | demo2       | 192.168.0.2 | 192.168.0.2 |      1352037667 | 55.670773  | 12.519322   | 1
 3632 | demo2       | 192.168.0.2 | 192.168.0.2 |      1352037186 | 55.670345  | 12.520807   | 1
 3631 | demo2       |             |             |      1352036983 | 55.671192  | 12.521445   | 0
 3630 | demo2       | 192.168.0.2 | 192.168.0.2 |      1352037200 | 55.670185  | 12.520733   | 1
 3629 | demo1       | 127.0.0.1   |             |      1352037119 | 55.670750  | 12.521905   | 4
 3628 | demo3       | 192.168.2.2 |             |      1352037667 | 55.670773  | 12.519322   | 4
 3627 | demo2       | 192.168.0.2 |             |      1352037667 | 55.670773  | 12.519322   | 4
(10 rows)
```

Database table column's description:

Column Name	Description
uid	Unique ID for each row
client_name	Name of the client
client_ip	Client IP address
client_timestamp	Client GPS fix timestamp
client_lat	Client GPS fix latitude
client_long	Client GPS fix longitude
event_type	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