

# WIRELESS HACKS™

**2nd  
Edition**

*Tips & Tools for Building, Extending,  
and Securing Your Network*



**O'REILLY®**

*Rob Flickenger & Roger Weeks*

HACK  
#20

## Share Your GPS

Caravanning with friends, but only one person has a GPS? Play nice and share.

Road trip caravans are more fun when the people in the two cars can communicate with each other. People have used CBs and Family Band Radio Service (FRS) radios for years just for this purpose. If you want to exchange data instead of voice over radio, you can set up a roving Wi-Fi network and then share a GPS between vehicles.

Connect a GPS to one computer and then set up a wireless network. Clients in other vehicles can connect to the GPS on the host machine. Normally, only one program at a time can access a serial port–connected GPS. This can quickly become a problem. You probably want to run `gpsmap` [Hack #22] to get maps, and then a program to log your position, and perhaps another program to let you create spatial annotations of your travels.

That is way too many connections for a single-user serial port! Fortunately, `GPSSd` is available as a daemon that connects directly to the GPS and then acts as a server for position information. Once you have `GPSSd` installed, all of your GPS aware applications can share a single GPS.

`GPSSd` is available at <http://gpsd.berlios.de/>. It runs under Linux, FreeBSD, Mac OSX, and any POSIX compliant Unix variant. You can build and install `GPSSd` in the standard way, as shown here, but do follow any current instructions from the web site (replace `x.xx` with the current version):

```
$ tar xvfz gpsd-x.xx.tar.gz
$ cd gpsd-x.xx
$ ./configure
$ make
$ make install
```

### Connecting the GPS

Most GPS units have a serial port and can be configured to output the current position using a standard protocol called NMEA. `GPSSD` expects your GPS to send NMEA formatted data, but can also accept data in different formats. Check the `GPSSd` site for how to enable other formats if you have a GPS that does not output standard NMEA sentences.

Once it is installed you can run it with this command:

```
$ gpsd -p /dev/ttyS0 -s 4800
```

If you have problems getting GPSd to start, consult the documentation. Once you've got it running you can connect to it with *telnet* to see what it is reporting:

```
$ telnet localhost 2947
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
```

Now you are connected to GPSD. The commands are single characters, and you can chain them together. So if you want your Position, Altitude, and GPS Status, issue this command:

**PADS**

You will get the full GPS position, with your latitude and longitude, altitude, current date and time, and GPS status. The status is the last section: S=0 or S=1, where 1 means the GPS is connected and sending data and 0 means there is no data coming from the GPS.

You can now run Kismet, GPSDrive, and any other GPS-aware applications at the same time, and your friends in the other car can connect to your GPS over the Wi-Fi network. The Perl code in [Example 1-1](#) will connect to GPSd and print the current position and time.

*Example 1-1. Connecting to GPSd with dump\_gps.pl*

```
#!/usr/bin/perl

# /usr/local/sbin/gpsd -p /dev/ttyS0
use 5.6.1;
use IO::Socket;
use Term::ReadKey;
use strict;

BEGIN { $|++ }; # autoflush STDOUT.

# Set parameters based on command line options, with defaults.
my $GPSD = shift( @ARGV ) || "localhost:2947";

# Connect to gpsd.
warn "connecting to gpsd \n";
my $gps = IO::Socket::INET->new($GPSD)
    or die "Can't connect to gpsd at $GPSD: $!\n";

while (1) {
    # Tell gpsd we want position, altitude, date, and status.
    $gps->print("pads\n");

    # Parse out the response. If the date is blank, gpsd needs
    # a second to catch up.
    my $location = <$gps>;
```

Example 1-1. Connecting to GPSd with *dump\_gps.pl* (continued)

```
my ($lat, $long, $alt, $date, $status) =
    ($location =~ /P=(.+) (.+),A=(.+),D=(.+),S=(+)/gos);

next unless $status;

# Print a line of data to both STDOUT and STDERR, and wait.
print $_ join("|", $lat, $long, $alt, $date || "" ), "\n"
    for ( \*STDOUT, \*STDERR );

sleep( 1 );
}
```

Save this into a file called *dump\_gps.pl* and run it with this command:

```
$ ./dump_gps.pl
```

You will get coordinates, the altitude, and current date at one-second intervals. We use this trick further in “Broadcast Your GPS Position” [Hack #21].

## See Also

- Casey West’s O’Reilly Network article on his cross-car road trip network: [http://www.oreillynet.com/pub/a/wireless/2002/08/01/highway\\_lan.html](http://www.oreillynet.com/pub/a/wireless/2002/08/01/highway_lan.html)

—Rich Gibson