

The GPSTABEL Manual

Version 0.0.2

Principal developer:

Robert Lipe

robertlipe@usa.net

Contents

1	What GPSTABEL is	3
2	General usage: the command line interface	3
3	Modes of operation: waypoints, routes, tracks	4
4	Supported input and output formats	5
4.1	Tabular overview on formats	6
4.2	CETUS	6
4.3	CoPilot Flight Planner for Palm OS	7
4.4	Delorme S&A	7
4.5	Delorme TopoUSA/XMap Conduit (XMap)	7
4.6	Delorme XMapHandHeld Street Atlas USA (XMapWpt)	7
4.7	GARMIN serial	8
4.8	GEO	8
4.9	Geocaching DB	8
4.10	GPSDRIVE	9
4.11	GPSMAN	9
4.12	GPSPILLOT	9
4.13	GPSUTIL	9
4.14	GPX	9
4.15	HOLUX	9
4.16	MAGELLAN (serial and SD card formats)	10

4.17	Magellan Navigator Companion (MAGNAV)	11
4.18	MAPSEND	11
4.19	MAPSOURCE	11
4.20	Maptech Exchange Format (MXF)	12
4.21	Navitrak DNA	12
4.22	OZI	12
4.23	PCX5	12
4.24	Pocket Streets Pushpin (PSP)	12
4.25	TIGER	14
4.26	TopoMapPro	14
4.27	National Geographic Topo! (TPG)	15
4.28	XCSV	15
5	Format-independent data filtering options	15
5.1	POSITION	15
5.2	RADIUS	16
5.3	DUPLICATE	16
6	GUI wrappers	17
7	Power usage	17
8	Building it from source	18
A	How to contribute	18
B	Magellan icon names	19
C	Acknowledgements and contact addresses	19
D	Gnu general public license	19

1 What GPSBABEL is

GPSBABEL is a conversion utility for a large and still increasing number of GPS related formats that represent waypoints, track points and route points.

There are simply too many and annoyingly different file formats to hold waypoint, route and track related information in various programs used by computers. Even though there are some promising attempts to create format standards, e.g. GPX as an XML variant, to carry all data types, there are too many programs that don't understand them yet and too much existing data that are in alternate formats.

GPSBABEL is based on an extensible foundation so that it is quite easy to add new formats. Most file formats implemented so far have taken less than 200 lines of reasonable ISO C code, so they can be stamped out pretty trivially. If you would like to contribute new features, see appendix A

GPSBABEL is designed for simple command line invocation for all supported platforms. For users not familiar with the command line (read: many Windows users) there is a mouse pitchforker's GUI interface to ease its use (see section 6). A full fathoming of GPSBABEL's features will need some command line familiarity, though.

GPSBABEL is released under the GNU general public license. For further details, see appendix D.

What GPSBABEL is not:

GPSBABEL is not intended to *create* any GPS related data, like waypoints or tracks. It does not support uploads or downloads of maps.

2 General usage: the command line interface

On every supported operating system platform GPSBABEL basically is a command line program. This kind of invocation, providing the utmost freedom of including runtime options, creates the best possible flexibility. Unfortunately, that can sometimes lead to unwieldy command lines.

To use this program, just tell it

- what kind of data you want to read,
- where to read it from,
- what format you want to be written,
- and what file or device to write it to.

For example¹

```
gpsbabel -i geo -f /tmp/infile.loc \  
        -o gpx -F outfile.gpx
```

tells it to read the file `infile.loc` in the `/tmp` directory² in the GEO-CACHING.COM format and create a new file named `outfile.gpx` in GPX format, to be placed in the present working directory.

The next command will read waypoint data from a Magellan unit attached to the first serial port and write them as a geocaching loc file. The first command is the Linux variant, the second command does the same for windows.

```
gpsbabel -i magellan -f /dev/ttyS0 -o geo -F mag.loc
```

```
gpsbabel -i magellan -f com1 -o geo -F mag.loc
```

`gpsbabel -?` will always show you the supported file types.

People less familiar with the command line will find a GUI interface helpful. A first one, for Windows, is described in section 6.

3 Modes of operation: waypoints, routes, tracks

There are three basic entities of GPS data that are of interest for a general conversion utility: waypoints, routes, and tracks.

Waypoints describe isolated and unrelated points in space each. Each is specified by its position (latitude, longitude, height above sea level), a "short" name (that is usually displayed to describe it on the receiver), a comment (or "long" name; an extended text associated to the waypoint), an icon (to display it on a receiver's map display page), and several less important features (like time of creation, display options, ...) that even vary between the different receivers.

Transferring waypoints is GPSBABEL's **default mode of operation**. It may as well be invoked by entering a `-w` as *first* command line argument.

Optionally, you may specify `-S` as a command line switch. This causes the program to ignore any "short" names that may be present in the source data format and synthesize one from the long name. This is particularly useful if you're writing to a target format that supports more elaborated short names but the source data was written in a very simple style. I use this for writing data from geocaching.com to my Magellan so my waypoints have "real" names instead of the 'GC1234' ones that are optimized for NMEA-only receivers.

¹In our examples we frequently split the respective command lines into several real lines to create a better readability. If a command line ends in a `"\"` (being the usual continuation character of an input line), this means for actual usage that the command itself extends into the next given line.

²Directory delimiters are always given in the Unix-ish variant, i.e.: `"/` is used where Windows users would enter a `"\"` or similar.

Routes are ordered sets of waypoints that describe a path to follow. Since most contemporary GPS receivers have very limited possible set sizes (typically 30 or 50), but support multiple routes to be administered in parallel, the usual way to handle them is to work on a set of routes at the same time. Read: a set of sets is handled.

Treatment of routes is only implemented for a part of GPSBABEL's formats. Details are given in the respective section on the various format types. To invoke route treatment, **enter - r as *first command line argument***.

Tracks are lists of waypoints with position and creation time information only. They are sampled in larger quantities when traveling. They are intended to describe the traveled path as exactly as possible (if so desired). Track logging memory of the receivers usually can gather 1000 to 10000 individual points.

Treatment of track data is only implemented for a part of GPSBABEL's formats. Details are given in the respective section on the various format types. To invoke track treatment, **enter - t as *first command line argument***.

4 Supported input and output formats

This section describes the supported file or transmission protocol formats in detail. Please remember that the capabilities of the various receiver types are quite different, so transfers from one receiver to another, or from a general file format to a receiver, might go wrong: you cannot deliver more waypoints, route or track points than the target is able to swallow. This kind of error cannot be handled by GPSBABEL.

4.1 Tabular overview on formats

File format	Waypoints		Tracks		Routes	
	Read	Write	Read	Write	Read	Write
Cetus	yes	yes				
CoPilot Flight Planner	yes	yes ³				
CSV	yes	yes				
Delorme S&A	yes	yes				
Delorme TopoUSA/XMap Conduit (XMap)	yes	yes				
Delorme HandHeld	yes	yes				
Street Atlas USA (XMapWpt)						
Garmin serial	yes	yes	yes			
GEO	yes	yes				
Geocaching DB	yes	yes			—	—
GPSDrive	yes	yes				
GPSman	yes	yes				
GPSPilot	yes	yes				
gpsutil	yes	yes				
GPX	yes	yes	yes	yes	yes	yes
Holux	yes	yes				
Magellan serial	yes	yes	yes	yes	yes	yes
Magellan SD card	yes	yes	yes	yes	yes	yes
Magellan Navigator Companion	yes	yes				
Mapsend	yes	yes	yes	yes	yes	yes
Mapsource	yes	yes				
Maptech Exchange Format (MXF)	yes	yes				
Navitrack DNA	yes	yes				
OziExplorer	yes	yes				
PCX5	yes	yes				
PocketStreets 2002 Pushpin	yes	yes				
Tiger	yes	yes				
TopoMapPro	yes	yes				
Topo by National Geographic	yes	yes				
XCSV	yes	yes				

In this table “—” means: not applicable since format will by definition not support a respective type of data.

4.2 CETUS

Cetus GPS (<http://www.cetusgps.dk/>) is a program for Palm/OS. Working with Ron Parker and Kjeld Jensen, we can now read and write files for that program. It hasn't been exhaustively tested, but has seemed fine on every input and output we've tried.

4.3 CoPilot Flight Planner for Palm OS

This code is mostly intended to convert CoPilot databases into other formats. I don't think you should use this to write CoPilot databases, although the code is there, mostly because GPSTabel doesn't convert magnetic declination values.

Questions, bug reports, etc, to ptomblin at xcski.com

<http://xcski.com/ptomblin/CoPilot/> <http://navaid.com/CoPilot/>

4.4 Delorme S&A

There are a billion variants of Comma Separated Value data. This is the one that makes Delorme S&A Deluxe 9 happy.

4.5 Delorme TopoUSA/XMap Conduit (XMap)

Delorme TopoUSA/XMap Conduit is a CSV variant. It's just like S&A with the addition of a completely pointless line at the beginning and end of the file. This is the format used to hot-sync to XMap from within TopoUSA. Done with help of Dan Edwards.

4.6 Delorme XMapHandHeld Street Atlas USA (XMapWpt)

Delorme XMapHandHeld Street Atlas USA is CSV variant. This is the format used by XmapHH SA USA on (at least) PocketPC O/S. This XMap is not to be confused with the XMap mentioned above. Contributed to gpsbabel by Alex Mottram.

Delorme XMap Handheld .WPT for PocketPC is a bit of a kludge. This document covers XMap Handheld Street Atlas USA edition.

XMap on the PocketPC stores it's waypoints in individual .wpt files. For example, waypoints generated by XMap on the PocketPC are stored by default in the "My Documents" folder using the sequential names "XMap1.wpt", "XMap2.wpt", ad nauseum. Needless to say, not very efficient.

As writing multiple waypoint files is outside of the scope of gpsbabel, gpsbabel chooses to write one big file, one waypoint per line. Extracting lines from this file is left as an exercise for the end user. A simple perl script to handle this conversion is included at the end of this README.

It should also be noted that READING multiple files is indeed possible, but if you have more than a few points, it can be a task. For example:

```
gpsbabel -i xmapwpt -f Xmap1.wpt -f Xmap2.wpt -o mapsend -F mapsend.wpt
```

will read the two Xmap .wpt files and write one mapsend file. This is fine for a small handful of points, but could be quite cumbersome for folks like me who have 100+ waypoints loaded into XMap. For *nix folks, something as simple as:

```
cat *.wpt & /tmp/foo.wpt gpsbabel -i xmapwpt -f foo.wpt -o mapsend -F mapsend.wpt
```

will do the trick just fine.

```
##### BEGIN SCRIPT
#!/full/path/to/perl
$INPUTFILE = @ARGV[0];
$TARGETDIR = @ARGV[1];
$FILENAME = @ARGV[2];

if (! $FILENAME) {
    print "Usage: xmap_split.pl INPUT_FILE OUTPUT_DIRECTORY FILENAME_BASE\n";
    print " (i.e. xmapl_split.pl points.wpt /tmp/points GPSB)\n";
    print " (created GPSB0001-GPSBXXXX in /tmp/points/ from points.wpt)\n";
    exit;
}

open (INFILE, $INPUTFILE) || die "Cannot open $INPUTFILE for read!\n";

while (<INFILE>) {
    $lc++;
    $filename = sprintf("%s/Gpsb%04d.wpt", $TARGETDIR, $lc);

    open (OUTFILE, ">$filename") || die "Cannot open $filename for write!\n";

    print OUTFILE $_;

    close(OUTFILE);
}

exit;

##### END SCRIPT
```

4.7 GARMIN serial

Waypoint serial upload and download works reliably under both POSIX and Windows. I tested it with a Vista graciously provided on loan by Joe Armstrong. The communications library used, jeeps, claims to support most models of Garmin hardware. Be sure the GPS is set for "Garmin mode" in setup and that nothing else (gpsd, getty, pppd, etc.) is using the serial port.

4.8 GEO

geocaching.com spits up geocaching.loc files that are XML-ish but not quite GPX. Because it's so close to GPX, this format is very well supported.

4.9 Geocaching DB

This is a PDA file format. It was tested against version 2 of GeocachingDB and a development snapshot of version 3. Information on the file format came from Dougs

Brat and Ron Parker. A particularly handy way to use GPSTranslate on these files is to use GPSTranslate to read a GPX file with Groundspeak (geocaching.com) extensions and let it write you a GeocachingDB file that contains the cache names, difficulty, terrain, and such.

<http://vip.hyperusa.com/dougs/geocachingdb/geocachingdb.htm>

4.10 GPSDRIVE

GpsDrive way.txt file format. A space separated format file. Tested against GpsDrive v 1.30 found @ <http://www.kraftvoll.at/software>. Contributed by Alan Curry.

4.11 GPSMAN

GPS Manager can read and write formats that this converter doesn't understand. The default formats (WGS84, DDD) work reliably.

4.12 GPSPILLOT

The file format for GPSPILLOT (<http://www.gpspilot.com>) was provided by Ron Parker. The output from this module has been tested with GPSPilot Tracker v5.05sx, but it is based on reverse-engineering so it may not work with all versions of all GPSPilot products. It had read-only support for Airport, Navaid, City and Landmark files but will read and write Point files.

4.13 GPSUTIL

GPSTool has a simple file format of this program that runs on POSIX-compliant OSes like UNIX and Linux. Reads and writes of this format are reliable. (I've also contributed to this program.) It's available at <http://www.cs.uakron.edu/hennings/gpstool/>.

4.14 GPX

This is the most capable and expressive of all the file formats supplied. It is described at <http://www.topografix.com/gpx.asp> and is supported by EasyGPS, ExpertGPS, and many other programs described at http://www.topografix.com/gpx_resources.asp

4.15 HOLUX

The Holuxgm-100 (e-fox) gps receiver uses standard compact flash cards. File formats were provided by Holux-Taiwan <http://www.holux.com.tw> to the author. The code was

tested against version 2.27E1; other versions and receivers may work but have not been explicitly tested. Anyone with information on other Holux receivers is encouraged to contact jochen@bauerbahn.net.

When copying the .wpo file to a flash card, the file must be named "tempwppt.wpo" as the receiver will ignore all other files.

Comparing the waypoints of a .wpo files against other formats like .gpx you may notice a small difference in the latitude and longitude values. The reason is the low resolution of the coordinates in the wpo file format. In a .wpo file the resolution is 1/10"; in gpx for example it is 1/100". As a practical matter, this loss is only about 1.7meters (5 feet).

The generated waypoint files can also be used by MapShow version 1.14. This program is free of charge from the Holux web site.

This format was contributed by Jochen Becker.

4.16 MAGELLAN (serial and SD card formats)

Description: This protocol supports serial uploads and downloads for the majority of the Magellan units. It has been tested to work reliably with the 315, 330, Meridian, and SportTrak family. In fact, it should work on any modern Magellan unit. Since the contents of the serial interface data transfer is the same that is written to or read from SD cards in the Meridian series, this kind of transfer is automatically supported as well.

GPSBABEL can also read and write the files that are written to or read from the SD flash memory cards in the Meridian models. Simply specify a file instead of a serial port.

As of this writing, there is still a lot of "scribbling" in the source for functionality that isn't hooked up to the rest of the program. Communication errors are handled.

Applicability: Waypoints, tracks, routes

Additional invocation options:

baud: Baud rate settings may be 1200, 2400, 4800, 9600, 19200 for serial line connections and must match receiver settings. Default value is 4800.

noack: Specifying "noack" turns handshake mode off. Without handshaking transfer of data is sped up significantly but *might* lead to overlooked transmission faults. Those have never been experienced during any testing, though. Default setting is "handshake on".

deficon: It is possible to specify a default icon to be displayed for every transmitted waypoint. Please refer to appendix B for a list of those.

Usage example: Read waypoint data from a Magellan receiver via the first serial line interface (Linux/Unix nomenclature given; for Windows replace "/dev/ttyS0" by "com1" etc.) with increased speed, but without handshaking, and write it to a Magellan formatted file.

```
gpsbabel -i magellan,noack,baud=9600 -f /dev/ttyS0 \
-o magellan -F outfile.mag
```

4.17 Magellan Navigator Companion (MAGNAV)

Magellan NAV Companion for Palm/OS is not really designed for this sort of use, but its file format is supported and with a little bit of patience you can both read and write NAV Companion waypoints.

This conversion is based on partially incomplete reverse-engineering of the record format, so it may not work with all versions of NAV Companion. It has been tested with version 2.10.

Translating NAV Companion waypoints to another format is as easy as with any other format. Just find the Companion_Waypoints database in your palm backup directory and use it as the input file.

When translating waypoints back to NAV Companion, though, you need to jump through some hoops:

First, you must merge any waypoints that already exist in the database in your Palm Backup directory with the ones you are adding; failure to do so will result in only the new points being available in NAV Companion, even if you give the new database a different name (it will overwrite the old database, even in your backup directory. That's a feature of PalmOS, not of NAV Companion.)

To merge the databases, use a command line like the following:

```
gpsbabel -i magnav -f Companion_Waypoints.PDB -i geo -f geocaching.loc -o magnav -F merged.pdb
```

Second, you must use the installer to install your new PDB file. Don't make the mistake of copying it over the existing Companion_Waypoints.PDB file; the one on the handheld will overwrite it rather than merging with it.

Finally, because NAV Companion is not designed to work with desktop applications, you must tell NAV Companion that its waypoints database has changed out from under it. One way to do this is to go to the waypoints screen and attempt to scroll; that will force it to reread the database and fix the record pointers that it keeps on the heap.

4.18 MAPSEND

Magellan was smart enough to document their file format to make creating software like this possible.

4.19 MAPSOURCE

Garmin Mapsource format appears compatible with the various members of that product family. Icon mapping is oriented toward versions above 4.07. Altitude, proximity, and depth are not supported.

Information on the Garmin Mapsource format was provided by Ian Cowley. The code was implemented by Robert Lipe.

4.20 Maptech Exchange Format (MXF)

Maptech Exchange Format - Another CSV format file. This format complies with (at least) Maptech Terrain Navigator, Terrain Professional, Take a Hike, and ExpertGPS import/export MFX. Contributed by Alex Mottram.

4.21 Navitrak DNA

Navitrak DNA marker format - Another CSV format file. This is the format that is compatible with the DNA Desktop import/export command. Reading the binary Markers.jwp format directly off the data card is not supported yet. Contributed by Tim Zickus.

4.22 OZI

OziExplorer Waypoint Format - Another CSV format file. Tested against OziExplorer v 3.90.3a / Shareware. Contributed by Alex Mottram.

4.23 PCX5

Garmin documents only PCX5, an older format limited to the lame NMEA six-character waypoint names that's treated as a second-class citizen in current versions of MapSource. Use file-`import` to read these files. Anyone with information on the *.mps file format that is now preferred is encouraged to contact me with the details or better yet, a module that implements it. I spent time trying to reverse-engineer a couple of *.mps files then I remembered that I don't own a Garmin and wasn't that inspired.

4.24 Pocket Streets Pushpin (PSP)

Microsoft's PocketStreets 2002 Pushpin (.PSP) format is not yet completely documented. THE .PSP MODULE DOES NOT WORK WITH MS STREETS & TRIPS 2002 .EST FILES. To create .PSP files from Streets & Trips 2002, you will need to have PocketStreets support installed. Please note that MS Streets & Trips only *EXPORTS* .PSP files. It does not import them. MS Streets & Trips 2002 only imports CSV files. To use .PSP files, simply copy them over to the same folder on the mobile device as the map (.MPS), and open PocketStreets. It should also be noted that in the case a pushpin is outside of the exported map area, the pin will be "grayed-out" and unused in PocketStreets. This is a good thing as it allows us to create one big .PSP

file that covers multiple .MPS files. Unfortunately, you need one .PSP file for every .MPS file. :(

Q: Why should I use gpsbabel/psp to make pushpins when Streets & Trips (S&T) already does that for me?

A: gpsbabel/psp has the advantage of being able to create pushpins WITHOUT creating the associated map file and the need to "import" the waypoint data into S&T. Through a series of scripts, you can create a dozen or so PSP files in a few seconds as opposed to a few weeks using the S&T interface. The maps are not going to change between sessions, only the pins will. Why waste all that time creating maps when all you really want are updated pins? As an aside, gpsbabel/psp creates points WITH THE PROPER coordinates where S&T does not in some areas of the U.S. (Nashville, TN for instance).

Q: I keep getting a blank (32 byte) PSP file?

A: There are either no points to write, or you have botched the command line for gpsbabel. gpsbabel is sensitive to UPPER and lower case on the command line. A simple command line to create PSP files looks like this:

```
gpsbabel -i geo -f geocaching.loc -o psp -F NewOrleans.psp
```

Note the use of "-f" for INPUT files and "-F" for OUTPUT files.

Q: I've created a PSP file, now what do I do with it?

A: To use pushpins in Pocketstreets, you need to have both a map and a pushpin file. These two files must exist in the same folder and have exactly the same base name as the map. For example, the pins that correspond to the map "NewOrleans.mps" should be named "NewOrleans.psp".

Q: I don't have a map? What do I do now?

A: Create one using the "Export map to Pocketstreets" option in S&T. You can also pick up some major city maps on the web from the MS Pocketstreets website if you are interested in seeing how it works.

Q: I have .EST files, not .PSP files. What's up with that?

A: In order to make PSP files, you need to use the "Export map to Pocketstreets" function in S&T. .EST files are for use in S&T, not Pocketstreets.

Q: The .PSP files differ when I use gpsbabel/psp versus Pocketstreets to create them. What's up?

A: Pocketstreets makes corrections to the S&T waypoint data upon initial loading. gpsbabel/psp writes PSP files with these corrections already made. Ask MS.

Q: Does gpsbabel/psp work with (Autoroute, Mappoint, etc..) .PSP files?

A: As of this writing, I haven't seen any so I can't be sure. If they follow the same layout as S&T 2002, I'd imagine so.

Q: Does gpsbabel/psp work with (S&T 2001, S&T 2002, etc...) files?

A: MS the file layout between S&T 2001 and S&T 2002. The gpsbabel psp module is known to work fine with S&T 2002 and 2003.

Q: Does gpsbabel/psp work with (insert your country/location here) maps?

A: If it doesn't, feel free to email the PSP files to me at: [geo_alexm at cox-internet.com](mailto:geo_alexm@cox-internet.com). I only had USA based data to work with while figuring out the file layout. Please include as much information about the points as possible (lat/long, name, etc..) and do it in English. I don't read or speak any other languages fluently.

Q: What do you mean S&T writes points with the wrong coordinates?

A: At some point in the "Export map to Pocketstreets" function in S&T, it goofs the lat/long data. Points in Nashville tended to shift 1.4 miles WEST of their original location. I'm not a geometry buff, but I'd imagine they have a reference point for generating coordinates that's wrong in (at least) that area.

Q: I have 800 waypoints that cover a dozen or so Pocketstreets maps. Do I need to split my points up into smaller chunks to match the area covered by the maps?

A: No. Pocketstreets will "ignore" points that are outside of the map area. Points that are not on the current map will be "grayed out" in pushpin explorer in Pocketsreets. This is the reason the PSP module was written for gpsbabel in the first place.

Q: Where can I find documentation for the layout of PSP files?

A: Just about everything I know about the PSP file format is documented in the source. To the best of my knowledge, there is no documentation (and for good reason, I've come to discover).

Q: I have some other problem, what do I do?

A: Email me at [geo_alexm at cox-internet.com](mailto:geo_alexm@cox-internet.com) and I'll see if I can work it out.

4.25 TIGER

The U.S. Census Bureau provides online mapping facilities. This format is described at: <http://tiger.census.gov/instruct.html>.

4.26 TopoMapPro

TopoMapPro Places File. Reads and writes places files for use in TopoMapPro (<http://www.topomappro.com>). As this file type can store links other than web links, anything that is not a http url will be discarded. Note that this does not do datum conversions, so if your input file does not have WGS84/NZGD2000 data, your output file won't either. Colour of waypoint icons defaults to red.

4.27 National Geographic Topo! (TPG)

National Geographic Topo! Waypoint Format. This filter reads and writes .TPG files created by various editions of NG Topo! This filter will **not** work with the newer combined .TPO files. Contributed by Alex Mottram.

4.28 XCSV

XCSV is an open-ended "Whatever Separated Values" parser / writer designed to work with user-supplied "style" files. It should handle at least a few thousand of the billion CSV variants available. By itself, it doesn't comply to any format, however **most** CSV variants can be described as a "style" and fine-tuned by the end user. For more information on it's use, please see README.style in the style/ sub-directory of gpsbabel. For an example of using the XCSV module within your C program, look at the ozi.c, mxf.c, and xmapwpt.c sources in the gpsbabel directory. This module was contributed to gpsbabel by Alex Mottram.

Additional Options: style - ****REQUIRED**** Path to XCSV style file.

snlen - Maximum length of synthesized shortnames. snwhite - Switch defining whether or not to allow whitespace in synthesized shortnames. (0 = NO WHITESPACE, 1 = WHITESPACE OK). snupper - Switch defining whether or not to force uppercase in shortnames. (0 = LEAVE AS IS, 1 = UPPERCASE ALL).

NOTE: sn* options require use of the '-s' command line option.

Example Usage:

```
gpsbabel -i gpsbabel -s -i gpx -f infile.gpx \
-o xcsv,style=my.style,snlen=8 -F outfile
```

```
gpsbabel -i xcsv,style=your.style -f infile \
-o xcsv,style=my.style -F outfile
```

5 Format-independent data filtering options

GPBABEL supports data filtering. Data filters are invoked from the command line via the -X option. They are invoked in the order they appear on the command line and can be used intermittently between several variations of input and output functions. It should also be noted that filtering data from different input types can sometimes produce undesirable results due to differences in the native data formats.

5.1 POSITION

The position filter is designed to remove points based on their proximity to each other. Distances can be passed on the command line by passing the distance=XXX option

to the filter. Distance options may be expressed in feet (distance=3f) or meters (distance=1m). The default is zero feet, essentially a duplicate position.

For example:

```
gpsbabel -i geo -f 1.loc -f 2.loc \  
-x position,distance=1f \  
-o mapsend -F 3.wpt
```

would load two input files of the GEO type, remove multiple points that are within 1 foot of each other, leaving just one, and finally would save a MAPSEND type output file.

5.2 RADIUS

The radius filter is designed to include points based on their proximity to a central point. Distances and the central point are declared on the command line by passing the distance=X.XX, lat=X.XX, and lon=X.XX options to the filter. Distance options may be expressed in miles (distance=3M) or kilometers (distance=3K). The default is zero miles.

For example:

```
gpsbabel -i geo -f 1.loc \  
-x radius,distance=1.5M,lat=30.0,lon=-90.0 \  
-o mapsend -F 2.wpt
```

would load a GEO input file, but include only points that lie within 1.5 miles of N 30.000 W 90.000 into the MAPSEND output file.

5.3 DUPLICATE

The duplicate filter is designed to remove duplicate points based on their shortname (traditionally a waypoint's name on the GPS receiver), and/or their location (to a precision of 6 decimals). This filter supports two options, "shortname" and "location". Generally, at least one of these options is REQUIRED.

For example:

```
gpsbabel -i gpx -f 1.gpx -f 2.gpx \  
-x duplicate,location,shortname \  
-o gpx -F merged_with_no_dupes.gpx
```

would remove points that have duplicate shortnames *AND* duplicate locations. The result would be a GPX file that more than likely contains only unique points and point data.

6 GUI wrappers

gpsbabelfront(.exe) is a Windows front-end for GPSTabel. It was contributed and is maintained by Josh McKee. It is written in Delphi. Figure 1 shows a screenshot of it.

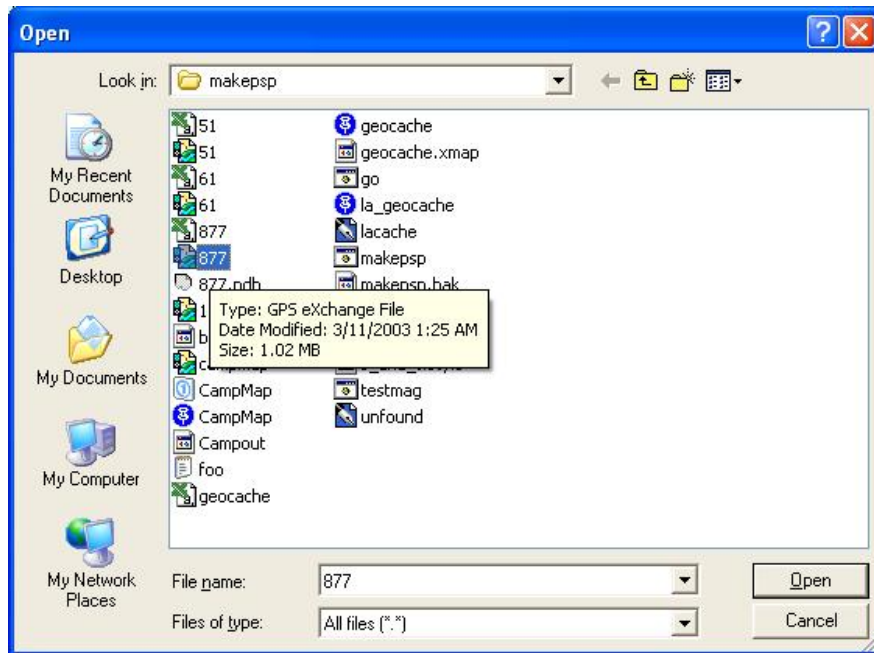


Figure 1: Screenshot of the Babel frontend.

7 Power usage

Argument are processed in the order they appear on the command line. **Input is cumulative.** The input file type remains unchanged until a new **-i** argument is seen. Files are read in the order they appear. So you could merge three input files into one output file with:

```
gpsbabel -i geo -f in-1.loc -f in-2.loc -f in-3.loc \
-o geo -F big-out.loc
```

You can as well **merge files of different types**:

```
gpsbabel -i geo -f 1.loc \
-i gpx -f 2.gpx \
-i pcx -f 3.pcx \
-o gpsutil -F big-out.gps
```

You can **write** the same data **in different output formats**:

```
gpsbabel -i geo -f 1.loc \
-o gpx -F out-1.gpx \
-o pcx -F out-2.wpt
```

8 Building it from source

The source code should be compilable on any system with ISO C89 compilers. It has been tested on UnixWare, OpenServer, OS/X, Linux, Solaris, and a variety of further processors and compilers.

LIBEXPAT is required. If you get errors about EXPAT.H being missing, you must either edit the MAKEFILE to tell the compiler where it is or install it in a sensible place. Expat can be downloaded from <http://expat.sourceforge.net>. As part of Apache it is very portable.

A How to contribute

If you are interested in contributing to this program, here are some guidelines:

- The frontier development state can always be obtained from a publicly accessible CVS on Sourceforge: CVS.GPSBABEL.SOURCEFORGE.NET. Check this CVS first before beginning your own enhancement efforts.
- Please ensure that you are building and testing against the latest code from the top of the CVS tree and that any code you modify is the latest version from the CVS. Please remember: Code changes sometimes occur frequently!
- Standards are good. ISO C and POSIX are great preferred.
- Reuse is OK, if doing so is not onerous. For example, using the EXPAT libraries vastly simplifies the XML parsers while increasing their robustness plus those libraries are ubiquitous. So I consider it OK to require EXPAT.
- Mail patches to **robertlipe@usa.net** for consideration and integration.
- If you are creating a new target you should submit patches (use "cvs diff -uN" to create patches) to the following files:
 - Yourcode.c and/or Yourcode.h - this is the code required to do your conversions and any support files that your code requires.
 - vecs.c - an updated vecs.c file implementing your conversion code into GPSBabel.
 - Makefile - an updated Makefile telling the compiler how to build and link your conversion into GPSBabel
 - README - an excerpt for the README about your conversion and any idiosyncrasies it may have.
 - testo - an updated script that tests your conversion (this should produce no output if all is good, see the current testo script for examples)
 - YourOutput - a sample file of code produced by your function (used in testo and lives in a directory called "reference").

- Compilers complain for a reason. Code shouldn't emit warnings.
- The entire world doesn't run just *<enter your favorite OS here>*. I've tested the present code on five different OSes. If you find yourself wanting to insert compiler or OS specific magic, please resist. We do not want to separate but to integrate.

B Magellan icon names

The table shows the list of icon names available on a variety of Magellan receivers. The names may be used to define a default icon that will be displayed after uploading sets of waypoints to a respective receiver (see section 4.16).

...

C Acknowledgements and contact addresses

D Gnu general public license

...