

# **GPSTBabel Documentation**

---

## GPSTBabel Documentation

---

---

---

---

# Table of Contents

Introduction .....	xvi
The Problem: Too many incompatible GPS file formats .....	xvi
The Solution .....	xvi
1. Getting it and Building it .....	1
2. Usage .....	2
Invocation .....	2
Suboptions .....	3
Advanced Usage .....	3
Route and Track Modes .....	4
Working with predefined options .....	5
Realtime tracking .....	6
Batch mode (command files) .....	6
3. The Formats .....	7
? Character Separated Values (xcsv) .....	7
style option .....	7
snlen option .....	7
snwhite option .....	7
snupper option .....	7
snunique option .....	8
urlbase option .....	8
prefer_shortnames option .....	8
datum option .....	8
Alan Map500 tracklogs (.trl) (alantrl) .....	8
Alan Map500 waypoints and routes (.wpr) (alanwpr) .....	9
All database fields on one tab-separated line (tabsep) .....	9
Brauniger IQ Series Barograph Download (baroiq) .....	9
Cambridge/Winpilot glider software (cambridge) .....	10
CarteSurTable data file (cst) .....	10
Cetus for Palm/OS (cetus) .....	10
dbname option .....	10
appendicon option .....	10
CoastalExplorer XML (coastexp) .....	11
Comma separated values (csv) .....	11
CompeGPS data files (.wpt/.trk/.rte) (compegps) .....	11
deficon option .....	12
index option .....	12
radius option .....	12
snlen option .....	12
CoPilot Flight Planner for Palm/OS (copilot) .....	12
cotoGPS for Palm/OS (coto) .....	13
zerocat option .....	13
Custom "Everything" Style (custom) .....	13
Dell Axim Navigation System (.gpb) file format (axim_gpb) .....	13
DeLorme .anl (drawing) file (anl) .....	14
type option .....	14
road option .....	14
nogc option .....	15
nourl option .....	15
deficon option .....	15
color option .....	16
zoom option .....	16

wpt_type option .....	16
radius option .....	16
DeLorme GPL (gpl) .....	16
DeLorme Street Atlas Plus (saplus) .....	16
DeLorme Street Atlas Route (saroute) .....	17
turns_important option .....	17
turns_only option .....	17
split option .....	17
controls option .....	17
times option .....	18
DeLorme XMap HH Native .WPT (xmap) .....	18
DeLorme XMap/SAHH 2006 Native .TXT (xmap2006) .....	18
DeLorme XMat HH Street Atlas USA .WPT (PPC) (xmapwpt) .....	18
Destinator Itineraries (.dat) (destinator_itn) .....	20
Destinator Points of Interest (.dat) (destinator_poi) .....	20
Destinator TrackLogs (.dat) (destinator_trl) .....	20
EasyGPS binary format (easygps) .....	21
Embedded Exif-GPS data (.jpg) (exif) .....	21
filename option .....	21
FAI/IGC Flight Recorder Data Format (igc) .....	21
IGC Data Format Notes .....	22
Converting to IGC format .....	22
Converting from IGC format .....	23
Merging into IGC format .....	23
timeadj option .....	24
Franson GPSGate Simulation (gpssim) .....	24
wayptspd option .....	24
split option .....	24
Fugawi (fugawi) .....	25
G7ToWin data files (.g7t) (g7towin) .....	25
Garmin 301 Custom position and heartrate (garmin301) .....	26
Garmin Logbook XML (glogbook) .....	26
Garmin MapSource - gdb (gdb) .....	26
cat option .....	26
bitscategory option .....	26
ver option .....	27
via option .....	27
roadbook option .....	27
Garmin MapSource - mps (mapsource) .....	27
snlen option .....	28
snwhite option .....	28
mpsverout option .....	28
mpsmergeout option .....	28
mpsusedepth option .....	28
mpsuseprox option .....	28
Garmin MapSource - txt (tab delimited) (garmin_txt) .....	29
date option .....	29
datum option .....	29
dist option .....	29
grid option .....	30
prec option .....	30
temp option .....	30
time option .....	30
utc option .....	30

Garmin PCX5 (pcx) .....	30
deficon option .....	31
cartoexploreur option .....	31
Garmin POI database (garmin_poi) .....	31
Garmin Points of Interest (.gpi) (garmin_gpi) .....	31
alerts option .....	32
bitmap option .....	32
category option .....	32
hide option .....	33
descr option .....	33
notes option .....	33
position option .....	33
proximity option .....	33
sleep option .....	34
speed option .....	34
unique option .....	35
units option .....	35
Garmin serial/USB protocol (garmin) .....	35
snlen option .....	38
snwhite option .....	38
deficon option .....	38
get_posn option .....	38
power_off option .....	38
resettime option .....	39
category option .....	39
bitscategory option .....	39
Garmin Training Centerxml (gtrnctr) .....	39
Geocaching.com .loc (geo) .....	40
deficon option .....	40
nuke_placer option .....	40
GeocachingDB for Palm/OS (gcdb) .....	40
Geogrid Viewer tracklogs (.log) (ggv_log) .....	40
GEOnet Names Server (GNS) (geonet) .....	41
GeoNiche .pdb (geoniche) .....	41
dbname option .....	41
category option .....	41
GlobalSat DG-100/BT-335 Download (dg-100) .....	41
erase option .....	42
Google Earth (Keyhole) Markup Language (kml) .....	42
deficon option .....	43
lines option .....	43
points option .....	43
line_width option .....	43
line_color option .....	43
floating option .....	43
extrude option .....	43
trackdata option .....	43
trackdirection option .....	44
units option .....	44
labels option .....	44
max_position_points option .....	44
Google Maps XML (google) .....	44
GpilotS (gpilots) .....	45
dbname option .....	45

GPS TrackMaker (gtm) .....	45
GPSBabel arc filter file (arc) .....	45
GpsDrive Format (gpsdrive) .....	46
GpsDrive Format for Tracks (gpsdrivetrack) .....	46
GPSman (gpsman) .....	46
GPSPilot Tracker for Palm/OS (gpspilot) .....	46
dbname option .....	46
gpsutil (gpsutil) .....	47
GPX XML (gpx) .....	47
snlen option .....	47
suppresswhite option .....	47
logpoint option .....	47
urlbase option .....	48
gpxver option .....	48
HikeTech (hiketech) .....	48
Holux (gm-100) .wpo Format (holux) .....	48
HSA Endeavour Navigator export File (hsandv) .....	49
HTML Output (html) .....	49
stylesheet option .....	49
encrypt option .....	49
logs option .....	49
degformat option .....	50
altunits option .....	50
IGN Rando track files (ignrando) .....	50
index option .....	50
Kartex 5 Track File (ktf2) .....	50
Kartex 5 Waypoint File (kwf2) .....	50
Kompass (DAV) Track (.tk) (kompass_tk) .....	51
Kompass (DAV) Waypoints (.wp) (kompass_wp) .....	51
KuDaTa PsiTrex text (psitrex) .....	51
Lowrance USR (lowranceusr) .....	51
ignoreicons option .....	52
writeasicons option .....	52
merge option .....	52
break option .....	52
Magellan Explorist Geocaching (maggeo) .....	52
Magellan Mapsend (mapsend) .....	52
trkver option .....	53
Magellan NAV Companion for Palm/OS (magnav) .....	53
Magellan SD files (as for eXplorist) (magellanx) .....	53
deficon option .....	54
maxcmts option .....	54
Magellan SD files (as for Meridian) (magellan) .....	54
deficon option .....	55
maxcmts option .....	55
Magellan serial protocol (magellan) .....	55
deficon option .....	56
maxcmts option .....	56
baud option .....	57
noack option .....	57
nukewpt option .....	57
MagicMaps IK3D project file (.ikt) (ik3d) .....	57
Map&Guide 'TourExchangeFormat' XML (tef) .....	57
routevia option .....	58

Map&Guide to Palm/OS exported files (.pdb) (mag_pdb) .....	58
Mapopolis.com Mapconverter CSV (mapconverter) .....	58
Notes .....	59
MapTech Exchange Format (mxf) .....	59
Microsoft AutoRoute 2002 (pin/route reader) (msroute) .....	59
Microsoft Streets and Trips (pin/route reader) (msroute) .....	60
Microsoft Streets and Trips 2002-2007 (s_and_t) .....	60
Motorrad Routenplaner (Map&Guide) .bcr files (bcr) .....	60
index option .....	61
name option .....	61
radius option .....	61
prefer_shortnames option .....	61
MS PocketStreets 2002 Pushpin (psp) .....	61
Frequently Asked Questions .....	62
MTK Logger (iBlue 747,...) Binary File Format (mtk-bin) .....	63
csv option .....	64
MTK Logger (iBlue 747,Qstarz BT-1000,...) download (mtk) .....	64
erase option .....	64
csv option .....	65
National Geographic Topo .tpg (waypoints) (tpg) .....	65
datum option .....	65
National Geographic Topo 2.x .tpo (tpo2) .....	65
National Geographic Topo 3.x/4.x .tpo (tpo3) .....	65
Navicache.com XML (navicache) .....	66
noretired option .....	66
Navigon Mobile Navigator .rte files (nmn4) .....	66
index option .....	66
NaviGPS GT-11/BGT-11 Download (navilink) .....	66
nuketrk option .....	67
nukerte option .....	67
nukewpt option .....	67
power_off option .....	67
Navitrak DNA marker format (dna) .....	68
NetStumbler Summary File (text) (netstumbler) .....	68
nseicon option .....	68
nsneicon option .....	68
seicon option .....	68
sneicon option .....	68
snmac option .....	69
NIMA/GNIS Geographic Names File (nima) .....	69
NMEA 0183 sentences (nmea) .....	69
snlen option .....	70
gprmc option .....	70
gpgga option .....	70
gpvtg option .....	70
gpgsa option .....	70
date option .....	70
get_posn option .....	71
pause option .....	71
append_positioning option .....	71
baud option .....	71
gisteq option .....	71
Nokia Landmark Exchange (lmx) .....	72
OpenStreetMap data files (osm) .....	72



tag option .....	72
tagnd option .....	72
OziExplorer (ozi) .....	73
pack option .....	73
snlen option .....	73
snwhite option .....	73
snupper option .....	73
snunique option .....	73
wptfgcolor option .....	74
wptbgcolor option .....	74
proximity option .....	74
PalmDoc Output (palmdoc) .....	74
nosep option .....	74
dbname option .....	74
encrypt option .....	75
logs option .....	75
bookmarks_short option .....	75
PathAway Database for Palm/OS (pathaway) .....	75
date option .....	75
dbname option .....	75
deficon option .....	75
snlen option .....	76
Quovadis (quovadis) .....	76
dbname option .....	76
Raymarine Waypoint File (.rwf) (raymarine) .....	76
location option .....	76
See You flight analysis data (cup) .....	77
Sportsim track files (part of zipped .ssz files) (sportsim) .....	77
Suunto Trek Manager (STM) .sdf files (stmsdf) .....	77
index option .....	78
Suunto Trek Manager (STM) WaypointPlus files (stmwpp) .....	78
index option .....	78
Swiss Map # (.xol) format (xol) .....	79
Tab delimited fields useful for OpenOffice, Ploticus etc. (openoffice) .....	79
Textual Output (text) .....	79
nosep option .....	79
encrypt option .....	80
logs option .....	80
degformat option .....	80
altunits option .....	80
splitoutput option .....	80
TomTom Itineraries (.itn) (tomtom_itn) .....	80
TomTom POI file (.asc) (tomtom_asc) .....	80
TomTom POI file (.ov2) (tomtom) .....	81
TopoMapPro Places File (tmpro) .....	81
TrackLogs digital mapping (.trl) (dmtlog) .....	81
index option .....	81
U.S. Census Bureau Tiger Mapping Service (tiger) .....	82
nolabels option .....	82
genurl option .....	82
margin option .....	82
snlen option .....	83
oldthresh option .....	83
oldmarker option .....	83

newmarker option .....	83
suppresswhite option .....	83
unfoundmarker option .....	83
xpixels option .....	83
ypixels option .....	84
iconismarker option .....	84
Universal csv with field structure in first line (unicsv) .....	84
datum option .....	86
grid option .....	86
utc option .....	86
Vcard Output (for iPod) (vcard) .....	86
encrypt option .....	87
VidaOne GPS for Pocket PC (.gpb) (vidaone) .....	87
Vito Navigator II tracks (vitosmt) .....	87
Vito SmartMap tracks (.vtt) (vitovtt) .....	87
WiFiFoFum 2.0 for PocketPC XML (wfff) .....	88
aicicon option .....	88
aioicon option .....	88
ahcicon option .....	89
ahoicon option .....	89
snmac option .....	89
Wintec WBT-100/200 Binary File Format (wbt-bin) .....	89
Wintec WBT-100/200 GPS Download (wbt) .....	89
erase option .....	90
Wintec WBT-201/G-Rays 2 Binary File Format (wbt-tk1) .....	90
Yahoo Geocode API data (yahoo) .....	90
addrsep option .....	90
4. Data Filters .....	91
Include Only Points Inside Polygon (polygon) .....	91
file option .....	92
exclude option .....	92
Include Only Points Within Distance of Arc (arc) .....	92
file option .....	93
distance option .....	93
exclude option .....	93
points option .....	94
Include Only Points Within Radius (radius) .....	94
lat option .....	94
lon option .....	94
distance option .....	94
exclude option .....	95
nosort option .....	95
maxcount option .....	95
asroute option .....	95
Interpolate between trackpoints (interpolate) .....	95
time option .....	96
distance option .....	96
route option .....	96
Manipulate track lists (track) .....	97
move option .....	97
pack option .....	97
split option .....	97
sdistance option .....	98
merge option .....	98

---

name option .....	98
start option .....	99
stop option .....	99
title option .....	99
fix option .....	99
course option .....	100
speed option .....	100
Rearrange waypoints by resorting (sort) .....	100
gcid option .....	100
shortname option .....	100
description option .....	100
time option .....	101
Remove all waypoints, tracks, or routes (nuketypes) .....	101
waypoints option .....	101
tracks option .....	101
routes option .....	101
Remove Duplicates (duplicate) .....	101
shortname option .....	102
location option .....	102
all option .....	102
correct option .....	102
Remove Points Within Distance (position) .....	103
distance option .....	103
all option .....	103
Remove unreliable points with high hdop or vdop (discard) .....	103
hdop option .....	104
vdop option .....	104
hdopandvdop option .....	104
sat option .....	104
Reverse stops within routes (reverse) .....	104
Save and restore waypoint lists (stack) .....	105
push option .....	105
pop option .....	105
swap option .....	106
copy option .....	106
append option .....	106
discard option .....	106
replace option .....	106
depth option .....	106
Simplify routes (simplify) .....	107
count option .....	107
error option .....	107
crosstrack option .....	107
length option .....	107
Transform waypoints into a route, tracks into routes, ... (transform) .....	108
wpt option .....	108
rte option .....	108
trk option .....	108
del option .....	109
A. Supported Datums .....	110
B. Garmin Icons .....	111
C. GPSBabel XCSV Style Files .....	113
Introduction .....	113
Style file overview .....	113

---

Internal Constants .....	114
WHITESPACE .....	115
COMMENTS .....	115
Global Properties of the File .....	115
DESCRIPTION .....	115
EXTENSION .....	115
ENCODING .....	115
DATUM .....	116
DATATYPE .....	116
GPSBabel Behavior Directives .....	116
SHORTLEN .....	116
SHORTWHITE .....	116
Defining the Layout of the File .....	116
FIELD_DELIMITER .....	116
RECORD_DELIMITER .....	117
BADCHARS .....	117
PROLOGUE .....	117
EPILOGUE .....	117
Defining Fields Within the File .....	117
IGNORE .....	118
CONSTANT .....	118
INDEX .....	118
SHORTNAME .....	119
DESCRIPTION .....	119
NOTES .....	119
URL .....	119
URL_LINK_TEXT .....	119
ICON_DESCR .....	119
LAT_DECIMAL .....	120
LON_DECIMAL .....	120
LAT_INT32DEG .....	120
LON_INT32DEG .....	120
LAT_DECIMALDIR / LAT_DIRDECIMAL .....	120
LON_DECIMALDIR / LON_DIRDECIMAL .....	120
LAT_DIR / LON_DIR .....	120
LAT_HUMAN_READABLE .....	121
MAP_EN_BNG .....	121
LON_HUMAN_READABLE .....	121
LATLON_HUMAN_READABLE .....	121
LAT_NMEA .....	121
LON_NMEA .....	122
LAT_10EX / LON_10EX .....	122
ALT_FEET .....	122
ALT_METERS .....	122
HEART_RATE .....	122
CADENCE .....	122
EXCEL_TIME .....	123
TIMET_TIME .....	123
YYYYMMDD_TIME .....	123
GMT_TIME .....	123
LOCAL_TIME .....	123
HMSG_TIME .....	123
HMSL_TIME .....	124
ISO_TIME .....	124

ISO_TIME_MS .....	124
GEOCACHE_DIFF .....	124
GEOCACHE_TERR .....	124
GEOCACHE_CONTAINER .....	125
GEOCACHE_TYPE .....	125
GEOCACHE_PLACER .....	125
GEOCACHE_ISAVAILABLE .....	125
GEOCACHE_ISARCHIVED .....	125
GEOCACHE_LAST_FOUND .....	125
GEOCACHE_HINT .....	125
PATH_DISTANCE_MILES .....	126
PATH_DISTANCE_KM .....	126
PATH_SPEED .....	126
PATH_SPEED_KPH .....	126
PATH_SPEED_MPH .....	126
PATH_SPEED_KNOTS .....	126
PATH_COURSE .....	127
GPS_HDOP / GPS_VDOP / GPS_PDOP .....	127
GPS_SAT .....	127
GPS_FIX .....	127
TRACK_NAME .....	127
ROUTE_NAME .....	127
STREET_NAME .....	127
CITY .....	128
COUNTRY .....	128
FACILITY .....	128
PHONE_NR .....	128
POSTAL_CODE .....	128
Examples .....	128
Miscellaneous Notes .....	129
Default Values .....	129
Glossary .....	130

---

## List of Tables

3.1. Grid values for garmin_txt .....	30
3.2. Supported distance units .....	34
3.3. Supported speed units .....	35
3.4. track file header (8 bytes) .....	87
3.5. track point (32 bytes) .....	88

---

## List of Examples

2.1. Command showing Linux download from Magellan serial and writing to .loc file .....	2
2.2. Command showing Windows download from Magellan serial and writing to .loc file .....	3
2.3. Merging multiple files into one .....	4
2.4. Merging multiple files of differing types. ....	4
2.5. Writing the same data in multiple output formats. ....	4
2.6. Read realtime positioning from Garmin USB, write to Keyhole Markup .....	6
3.1. Example 'csv' file .....	11
3.2. Example for gdb bitcategory option to put all waypoints in categories 1 and 16. ....	27
3.3. Using gdb option roadbook to create simple html roadbook .....	27
3.4. Command showing garmin_txt output with all options .....	29
3.5. Command showing garmin_gpi output example .....	32
3.6. Read GPX file, create GPI to alert when you're 1/2 mile from a speed camera. ....	34
3.7. Example for garmin bitcategory option to put all waypoints in categories 1 and 16. ....	39
3.8. Command showing DG-100 download and erase on Linux .....	42
3.9. Sample BCR command with all options .....	61
3.10. Convert MTK binary trackpoints to GPX .....	64
3.11. Command showing MTK download track and waypoints and erase on Linux .....	64
3.12. Example for splitoutput option to text format .....	80
3.13. Command showing conversion of a Wintec binary file to GPX .....	89
3.14. Command showing WBT-200 download and erase over Bluetooth on Mac OS X .....	89
3.15. Command showing conversion of a Wintec binary file to GPX .....	90
4.1. Using the polygon filter .....	92
4.2. Using the polygon and arc filters to find points in or nearly in a polygon .....	92
4.3. Using the arc filter .....	93
4.4. Using the radius filter to find points close to a given point .....	94
4.5. Using the interpolate filter .....	96
4.6. Time-shifting a track with the track filter .....	97
4.7. Merging tracks with the track filter .....	98
4.8. Extracting a period of time with the track filter .....	99
4.9. Filtering data types with nuketypes .....	101
4.10. Using the duplicate filter to suppress points with the same name and location .....	102
4.11. Using the duplicate filter to implement an "ignore list." .....	102
4.12. Using the duplicate filter to correct the locations of "puzzle" geocaches .....	103
4.13. Using the position filter to suppress close points .....	103
4.14. Using the discard filter for HDOP and VDOP. ....	104
4.15. Using the discard filter to require at least three satellites. ....	104
4.16. Converting a track to a sequence of waypoints .....	108
4.17. Converting a pile of waypoints to a GPX route .....	108
4.18. Converting a pile of waypoints to a GPX track .....	109
4.19. Convert a GPX track to GPX waypoints, tossing the original track .....	109

---

# Introduction

## The Problem: Too many incompatible GPS file formats

There are simply too many gratuitously different file formats to hold waypoint, track, and route information in various programs used by computers and GPS receivers. GPX [<http://www.topografix.com/gpx.asp>] defines a standard in XML to contain all the data, but there are too many programs that don't understand it yet and too much data in alternate formats.

Perhaps you have an Explorist 600 and your friend has a StreetPilot 2720. You've collected a list of your favorite locations as waypoints and you'd like to be able to share them. Unfortunately, his copy of Garmin Mapsource won't read data created by your copy of Magellan Mapsend DirectRoute. What you need is a program that converts data between the two programs.

GPSBabel actually solves that problem for you and much more...

## The Solution

The original author of GPSBabel, Robert Lipe [</people/robertlipe.html>], needed to convert waypoints between a couple of formats, so he whipped up a converter and designed it upon an extensible foundation so that it was easy to add new formats and made the program freely available. Many others [</people/index.html>] have contributed to the program since then.

Most file formats added so far have taken under 200 lines of reasonable ISO C so they can be stamped out pretty trivially. Formats that are ASCII text delimited in some fixed way can be added with no programming at all via our style mechanism.



---

# Chapter 1. Getting it and Building it

GPSTBabel is distributed "ready to run" on most common operating systems via the download page [<http://www.gpsbabel.org/download.html>].

As GPSTBabel runs on a wide variety of operating systems, be sure to visit the OS-Specific notes [<http://www.gpsbabel.org/osnotes.html>] for additional information.

For operating systems where no binary is provided or if you want the latest development version, you will have to build it from source. The code should be compilable on any system with ISO C89 compilers. It's been tested on UnixWare, OpenServer, OS/X, Linux, Solaris, and a variety of processors and compilers.

In most cases, the code is as simple to build as running:

```
./configure && make
```

Expat [<http://expat.sourceforge.net>] is strongly recommended for source builds as it is required for reading all the XML formats such as GPX. Fedora users may need to 'yum install expat-devel'. Ubuntu users may need to 'apt-get install expat libexpat-dev'.

libusb [<http://libusb.sourceforge.net>] is recommended for OS/X and Linux if you want to use a USB Garmin. Fedora users may need to 'yum install expat-devel'. Ubuntu users may need to 'apt-get install libusb-dev'.

There are additional flags that can be passed to configure to customize your build of GPSTBabel.

```
./configure --help
```

lists all the supported options, but of interest we have:

--disable-shapefile Excludes the shapefile support.

--disable-pdb Excludes the Palm database support and all formats that rely on it.

--disable-csv Excludes all support for our something-separated formats.

--disable-filters Excludes all filter support.

--enable-efence Activate debugging mode for gpsbabel-debug.

--with-doc=dir Specifies that the doc should be created and installed in dir.

--without-libusb Disables use of libusb, even if it's available.

--with-zlib=(included) | system | no By default, we use our own version of zlib. If you specify system the system zlib is used. A value of no (or --without-zlib) disables zlib.

---

# Chapter 2. Usage

## Invocation

If you're using GPSTabel, you will need to know how to do at least two things: read data from a file, and write it to another file. There are four basic options you need to know to do those things:

**Command:** `-i format`

**Meaning:** Set input format

**Command:** `-f filename`

**Meaning:** Read file

**Command:** `-o format`

**Meaning:** Set output format

**Command:** `-F filename`

**Meaning:** Write output file

### Important

Case matters. Notably `-f` (lowercase) sets the *input* file. `-F` (uppercase) sets the *output* file.

The *format* parameters in the above list refer to the names of formats or file types supported by GPSTabel.

```
gpsbabel -?
```

will always show you the supported file types. In this document, the various supported formats are listed in Chapter 3, *The Formats*. The name that you would use on the command line follows the format name in parentheses.

Options are *always* processed in order from left to right. In practical terms, this means that things you want to read should appear in the command before things you want to write. This sometimes surprises new users as adding options to turn on debugging at the end, for example, doesn't work as the debugging is turned on after all the interesting work is done. The reason for this strict ordering becomes more apparent once you learn about mixing formats and filters.

The *filename* parameters specify the name of a file to be read or written.

To use GPSTabel in its simplest form, just tell it what you're reading, where to read it from, what you're writing, and what to write it to. For example:

```
gpsbabel -i geo -f /tmp/geocaching.loc -o gpx -F /tmp/geocaching.gpx
```

tells it to read the file `/tmp/geocaching.loc` in `geocaching.com` format and create a new file `/tmp/geocaching.gpx` in GPX format. It's important to note that the names have nothing to do with the formats actually used.

This command will read from a Magellan unit attached to the first serial port on a Linux system (device names will vary on other OSes; typically COMx: on Windows) and write them as a `geocaching loc` file.

**Example 2.1. Command showing Linux download from Magellan serial and writing to .loc file**

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

This second command does the same on Microsoft Windows.

### Example 2.2. Command showing Windows download from Magellan serial and writing to .loc file

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

Optionally, you may specify `-s` in any command line. 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 isn't the lowest common denominator but the source data was written for the lowest common denominator. This is useful for writing data from geocaching.com to a GPS so my waypoints have "real" names instead of the 'GC1234' ones that are optimized for receivers of the lowest common denominator. A geocacher using Linux with a Magellan receiver may thus find commands like this useful.

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

His counterpart on Windows will find this equivalent

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

## Suboptions

Many of the available format options in GPSTools can themselves take options. While we try to make all the formats do the most sensible thing possible without any extra options; this allows great power and flexibility in the operation of the program.

Suboptions are comma separated and immediately follow the option itself. The available suboptions are listed on the individual format pages. We'll make an example from the section called "Google Earth (Keyhole) Markup Language (kml)":

```
gpsbabel -i gpx -f file.gpx -o kml,deficon="file:///myicon.png",lines=0  
-F one.kml -o kml -F two.kml
```

This command will read the GPX file `file.gpx` and create two KML files. `one.kml` will have the given icon and no lines between track and routepoints. `two.kml` will be created with the defaults used in the KML writer.

Suboptions for the various formats allow you to change serial speeds, pass arguments to filters, change the type of file written, override icon defaults, and lots of other things. The suboptions for each filetype are documented on the page in this document that describes the option itself.

## Advanced Usage

Argument are processed in the order they appear on the command line and are translated internally into a pipeline that data flows through when executed. Normally one would:

read from one input  
optionally apply filters

write into one output

but GPSTBabel is flexible enough to allow more complicated operations such as reading from several files (potentially of different types), applying a filter, reading more data, then writing the merged data to multiple destinations.

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:

### Example 2.3. Merging multiple files into one

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

You can merge files of different types:

### Example 2.4. Merging multiple files of differing types.

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

### Example 2.5. Writing the same data in multiple output formats.

You can write the same data in different output formats:

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

If you want to change the character set of input or/and output side you can do this with the option *-c* *<character set>*. You can get a complete list of supported character sets with "gpsbabel -l". To change the character set on both sides you should do this:

```
gpsbabel -i xcsv,style=foo.style -c latin1 -f foo -o xcsv,style=bar.style -c ms-ansi -F bar
```

Note, that some formats have a fixed character set and ignore this option.

## Route and Track Modes

Most formats supported by GPSTBabel will make a reasonable attempt to work transparently with waypoints, tracks, and routes. Some formats, like garmin and magellan require the *-t* flag to work with tracks and *-r* to work with routes. *-w* is for waypoints, and is the default. So if you wanted to read all data from a Magellan Meridian GPS receiver into a gpx file, you might use a command like:

```
gpsbabel -t -r -w -i magellan -f com1: -o gpx -F backup.gpx
```

Tracks and routes are advanced features and don't try to handle every possible hazard that can be encountered during a conversion. If you're merging or converting files of similar limitations, things work very well.

Many of those hazards can be overcome with our filters but there are often compromises to be made. For example, if you have a GPX route that contains 150 turn points but you're sending the route to a

GPS receiver that supports only 30 turnpoints, something has to go. One might use our 'simplify' filter to produce a route that retained the 30 most mathematically significant turnpoints but that may not really be the route you had in mind.

Tracks and routes will sometimes be converted to a list of waypoints when necessary, One example is when writing into one of the CSV formats. The inverse operation is not supported right now, so reading the converted track back from CSV will always result in a list of waypoints, not the original track.

The presence of `-s` on the command line tends to create havoc on tracks and routes since many of these formats rely on internal linkages between such points and renaming them may break those linkages. In general, don't use `-s` when tracks or routes are present.

## Working with predefined options

GPSTabel can read a file on startup to set defaults for options. All module and filter options may be set this way.

The format of the file is identical to the ini-file-format often seen on Windows. Here is an example:

```
[Common format settings]
snupper=Y
snlen=10
[gpx]
gpxver=1.1
[magellan]
baud=115200
[tiger]
[Garmin categories]
; any # from 1 to 16
1=fixed waypoints
2=temporary waypoints
```

Each section of the file starts with a '[section]' header followed by any number of lines formatted `option=value`. Leading and trailing whitespace will be automatically removed from header, option and value items. Lines starting with '#' or ';' will be treated as comments and ignored.

There are three optional sections.

- Common format settings.

Any option from any of the formats listed here will be used by GPSTabel unless explicitly provided on the command line.

- Common filter settings.

As above, but for filters.

- Garmin categories

This allows you to give readable names to the numeric categories used internally in some Garmin devices and the Mapsource formats such as GDB and MPS. This information is also used by our GPX and garmin\_txt formats as well.

By default, GPSTabel tries at startup to load the file named `gpsbabel.ini` from the following locations:

- current working directory

- Windows: all paths "APPDATA", "WINDIR", "SYSTEMROOT" declared in environment.
- Unix like OS'ses: \${HOME}/.gpsbabel/, /usr/local/etc/ and /etc/

If the `-p` option is specified, the above locations are not searched. Only the filename specified by that option will be used.

There may be situations where predefined values are not useable (i.e. wrapper applications using GPSBabel in the background). The inifile mechanism can be disabled with an empty filename.

```
gpsbabel -p "" -i gpx -f something.gpx -o tiger -F -
```

## Realtime tracking

Introduced in GPSBabel 1.3.1, we now have an *experimental* feature for realtime tracking via the new `-T` option. This reads position reports from selected formats and writes an output file when a position report is received.

As of this writing, Garmin's PVT protocol and NMEA are supported inputs. KML, NMEA, and the various XCSV formats are supported on output. Additional formats may be added by interested parties later.

### Example 2.6. Read realtime positioning from Garmin USB, write to Keyhole Markup

```
gpsbabel -T -i garmin -f usb: -o kml -F xxx.kml
```

Will read the USB-connected Garmin and rewrite 'xxx.kml' atomically, suitable for a self-refreshing network link in Google Earth.

## Batch mode (command files)

In addition to reading arguments from the command line, GPSBabel can read directions from batch (or command) files via the `-b` option.

These files are ideal for holding long command lines, long file lists, complex filters and so on. You can use all GPSBabel options and combinations when writing such files. Nesting batch files by using the `-b` option within a batch file is supported.

Here is an example demonstrating segmenting a large command line by placing the input and filtering directives in a file called 'all\_my\_files'.

```
gpsbabel -b all_my_files -o gdb -F all_my_tracks.gdb
```

'all\_my\_files' could look like this:

```
-i gpx
-f saxony_in_summer_2004.gpx -f austria_2005.gpx
-i gdb
-f croatia_2006.gdb
-x nuketypes,waypoints,routes
-x track,pack,split,title="LOG # %Y%m%d"
```

---

# Chapter 3. The Formats

## ? Character Separated Values (xcsv)

This format can...

- read and write waypoints

This format is a very flexible module that can be used to read or write nearly any plain-text record-based waypoint file. This flexibility is achieved by combining this format with "style" files that describe the format of the waypoint files.

There are several formats built in to GPSBabel that use the underlying xcsv machinery. Each of those formats takes the same options as the xcsv format, with the obvious exception of the `style` option. Those formats are all based on style files that can be found in the "style" directory in the GPSBabel source distribution.

### style option

Full path to XCSV style file.

This option specifies the style file that defines the records to be read on input or written on output. This is not a valid option for the various built-in xcsv-based styles; they have prebuilt style definitions.

For information on the format of xcsv style files, see Appendix C, *GPSBabel XCSV Style Files*.

### snlen option

Max synthesized shortname length.

This option specifies the maximum allowable length for a short name on output. This option overrides the style file.

Valid values for this option are 0 (off) and 1 (on).

### snwhite option

Allow whitespace synth. shortnames.

When this option is specified, GPSBabel will allow whitespace (spaces or tabs) in generated short names. This option overrides the style file.

Valid values for this option are 0 (off) and 1 (on).

### snupper option

UPPERCASE synth. shortnames.

When this option is specified, GPSBabel will make all short names contain only UPPERCASE characters. This option overrides the style file.

Valid values for this option are 0 (off) and 1 (on).

## snunique option

Make synth. shortnames unique.

When this option is specified, GPSTBabel will ensure that all short names are unique within the output file. This option overrides the style file.

Valid values for this option are 0 (off) and 1 (on).

## urlbase option

Basename prepended to URL on output.

This option specifies the base name to prepend to a URL on output. This might be useful if an input file contains URLs in a relative format and you need them to be in an absolute format.

## prefer\_shortnames option

Use shortname instead of description.

This option causes GPSTBabel to use the short name of the waypoint instead of the description. This overrides the style file.

Valid values for this option are 0 (off) and 1 (on).

## datum option

GPS datum (def. WGS 84).

This option specifies the GPS datum to be used on read or write. Valid values for this option are listed in Appendix A, *Supported Datums*.

## Alan Map500 tracklogs (.trl) (alantrl)

This format can...

- read and write tracks

GPSTBabel supports .wpr and .trl files for Alan Map500 devices running operating system versions 2.xx.

.trl contain files tracklogs. If you use a CF-Card based operating system, tracklog files must have a .TRL extension when copied to the CF-Card. The default filename is TEMP\_TRK.TRL. Only one .TRL file may be present.

Alan's operating system 3.0 for Map500 is not supported yet. At the time of this writing, OS3 is still beta. Documentation on the new dataformats is sparse.

The Alan Map500 handheld GPSr is identical to the Holux GM101. This GPSTBabel module has only been tested against the Alan Map500. Still, if you use a GM101, GPSTBabel will probably be able to convert your waypoints, routes and tracklogs.

For more information on the Alan Map500 visit Alan Germany [<http://www.alan-electronics.de/gps/gpsuebersicht.php3>]. There is very informative forum [<http://www.alan-germany.de/forum/index.php>], too. The forum language is German but posts in English will be answered, too.



## Alan Map500 waypoints and routes (.wpr) (alanwpr)

This format can...

- read and write waypoints
- read and write routes

GPSTabel supports .wpr and .trl files for Alan Map500 devices running operating system versions 2.xx.

.wpr files contain waypoints and routes. If you use a CF-Card based operating system, waypoint files must have a .WPR extension when copied to the CF-Card. The default filename is TEMPWPRT.WPR. Only one .WPR file may be present.

Alan's operating system 3.0 for Map500 is not supported yet. At the time of this writing, OS3 is still beta. Documentation on the new dataformats is sparse.

The Alan Map500 handheld GPSr is identical to the Holux GM101. This GPSTabel module has only been tested against the Alan Map500. Still, if you use a GM101, GPSTabel will probably be able to convert your waypoints, routes and tracklogs.

For more information on the Alan Map500 visit Alan Germany [<http://www.alan-electronics.de/gps/gpsuebersicht.php3>]. There is very informative forum [<http://www.alan-germany.de/forum/index.php>], too. Forum language is German but posts in English will be answered, too.

## All database fields on one tab-separated line (tabsep)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

This format, like the custom format, is mainly used for the purpose of testing GPSTabel. It is supposed to contain one field for each piece of information supported by the xcsv format writer, but it may not be entirely in sync with the documentation at Appendix C, *GPSTabel XCSV Style Files*.

For a list of fields, see the style/tabsep.style file in the GPSTabel source distribution.

## Brauniger IQ Series Barograph Download (baroiq)

This format can...

- read tracks

Serial download protocol for the Brauniger [<http://www.brauniger.com>] IQ series of barograph recording flight instruments. This format creates a track of altitude vs time which can be merged with a GPS track of the same flight to create a three dimensional IGC file.

## Cambridge/Winpilot glider software (cambridge)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Support for Cambridge [<http://www.cambridge-aero.com/products.htm>] and Winpilot [<http://www.winpilot.com>] flight analysis and planning software for glider pilots.

## CarteSurTable data file (cst)

This format can...

- read waypoints
- read tracks
- read routes

With this format we can read CarteSurTable [<http://phgiraud.free.fr/CarteSurTable/CarteSurTable.htm>] data files. CarteSurTable [<http://phgiraud.free.fr/CarteSurTable/CarteSurTable.htm>] is a shareware program widely used in France. The data inside have to be seen as a mixture of a waypoints list, one route and several tracks.

## Cetus for Palm/OS (cetus)

This format can...

- read and write waypoints
- read tracks

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.

## dbname option

Database name.

This option specifies the database name for the output file. This name is not the same thing as the file name on your computer; this is the name that appears in the file browser on your handheld.

## appendicon option

Append icon\_descr to description.

This option will add the icon description to the end of the waypoint description on output. This can be useful if the icon is used to convey important information about the waypoint. For example, the icon might

be "found geocache" or "unfound geocache"; it might be useful to know that when looking at a list of icons in Cetus.

## CoastalExplorer XML (coastexp)

This format can...

- read and write waypoints
- read and write routes

This is the format used by CoastalExplorer™. The format is XML with items uniquely identified by Windows-style UUIDs. <http://www.rosepointnav.com>

## Comma separated values (csv)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

There are a billion variants of Comma Separated Value data. This is the one specifically that makes Delorme [<http://www.delorme.com>] S&A Deluxe 9™ happy. It's also a very simple program and useful for many other programs like spreadsheets.

CSV is also the correct format for Lowrance MapCreate™, their commercial mapping program, or GDM6 (their free waypoint manager) for iFinder which is available at [lowrance.com](http://www.lowrance.com/Software/GDM6/Default.asp) [<http://www.lowrance.com/Software/GDM6/Default.asp>]

On write, this format writes simple "latitude, longitude" pairs, but on read it will read anything supported by our human readable definition.

For something-separated data that has headers identifying the various fields, see our universal csv format.

### Example 3.1. Example 'csv' file

```
35.97203, -87.13470, Mountain Bike Heaven by susy1313
36.09068, -86.67955, The Troll by a182pilot & Family
35.99627, -86.62012, Dive Bomber by JoGPS & family
36.03848, -86.64862, FOSTER by JoGPS & Family
```

## CompeGPS data files (.wpt/.trk/.rte) (compegps)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

CompeGPS™ data files are "character" separated text files like the pcx format. "Character" means special data lines can have their own separator.

Since release 6.1 of CompeGPS™, GPX is also a supported import/export format for waypoints, routes and tracks.

For more information please have a look at <http://www.compegps.com>

## deficon option

Default icon name.

This option specifies the default icon name on output.

## index option

Index of route/track to write (if more the one in source).

Because this format supports only one route or track, this option may be used on output to select a single route or track from a collection of routes and tracks read from a more expressive format. If you have, say, a gpx file that contains two routes, you may use this option to write them one at a time to individual files.

```
gpsbabel -i gpx -f routes.gpx -o compegps,index=1 -F route1.txt -o  
compegps,index=2 -F route2.txt
```

## radius option

Give points (waypoints/route points) a default radius (proximity).

This option specifies the default proximity for waypoints and route points.

## snlen option

Length of generated shortnames (default 16).

This option specifies the default length for short names generated on output. The default length is 16.

# CoPilot Flight Planner for Palm/OS (copilot)

This format can...

- read and write waypoints

This code is mostly intended to convert CoPilot Flight Planner for Palm/OS" [<http://xcski.com/~ptomblin/CoPilot/>] databases into other formats. You probably should not use this to write CoPilot databases, although the code is there, because GPSTabel doesn't convert magnetic declination values.

This version now reads all CoPilot file versions up to 4, but only writes version 4 files. If you have a need for a version flag, please let me know.

Questions, bug reports, etc, to [ptomblin at xcski.com](mailto:ptomblin@xcski.com)

<http://xcski.com/~ptomblin/CoPilot/> and <http://navaid.com/CoPilot/> [<http://navaid.com/CoPilot/>]

## cotoGPS for Palm/OS (coto)

This format can...

- read and write waypoints
- read tracks

This format supports cotoGPS™, a Palm™ GPS program. It can read both track and marker (waypoint) files. It is currently unable to write track files, so only marker files can be written. The marker categories are written to and read from the icon description. The 'Not Assigned' category leaves the icon description empty on read. Currently geocache info is ignored.

In addition to the documented options, this format also has a debugging option called `internals` which takes an XCSV delimiter value. It writes some internal values (distance, arc, x and y) of the cotoGPS track format to the notes field.

Contributed by Tobias Minich.

cotoGPS [<http://core.de/~coto/projects/cotogps/>]

## zerocat option

Name of the 'unassigned' category.

This option specifies a name for the "Not Assigned" category in the Palm database. The default is "Not Assigned".

## Custom "Everything" Style (custom)

This format can...

- read and write waypoints

This format is derived from the `xcsv` format, so it has all of the same options as that format.

This format is not actually used by any real product. It is most useful for debugging purposes when developing a new format module for GPSTable.

To understand the contents of this file, look at the `style/custom.style` file in the GPSTable source distribution as well as Appendix C, *GPSTable XCSV Style Files*.

## Dell Axim Navigation System (.gpb) file format (axim\_gpb)

This format can...

- read tracks

This format reads the binary (.gpb) track logs recorded on Dell [<http://www.dell.com>] Axim Navigation Systems.

This is a read-only format for now as the format was reverse engineered and there are many unknown bytes. We can successfully extract the common GPS data.

## DeLorme .an1 (drawing) file (an1)

This format can...

- read and write waypoints
- write tracks
- read and write routes

This format supports the DeLorme [<http://www.delorme.com>] ".an1" drawing file format. It can currently be used to either read or write drawing files. If you use this format to create drawing files with routes or waypoints from another source, by default it will create "Red Flag" symbols for waypoints, and thick red lines for routes or tracks. It is possible to merge two drawing layers by doing something like this:

```
gpsbabel -i an1 -f one.an1 -f two.an1 -o an1 -F merged.an1
```

In this case, the merged data will contain all of the properties of the original data.

## type option

Type of .an1 file.

This option specifies the type of the drawing layer to be created. The supported values are "drawing", "road", "trail", "waypoint", or "track". If you do not specify a type, the default will be either the type of the previous an1 file or "drawing" if there is no previous file. This lets you merge, for example, two road layers without having to specify "type=road" for the output.

## road option

Road type changes.

If you are creating a road layer, you may use the "road" option, which allows you to change the types of roads based on their names. You can change multiple roads at the same time. Currently supported types are

**Type:** limited

**Meaning:** Limited-access freeways

**Type:** toll

**Meaning:** Limited-access toll highways

**Type:** ramp

**Meaning:** Access ramps for limited-access highways

**Type:** us

**Meaning:** National highways (e.g. US routes)

**Type:** primary

**Meaning:** Primary State/Provincial routes

**Type:** state

**Meaning:** State/Provincial routes

**Type:** major

**Meaning:** Major Connectors

**Type:** ferry

**Meaning:** Ferry Routes

**Type:** local

**Meaning:** Local Roads

**Type:** editable

**Meaning:** User-drawn Roads

GPSTabel defaults to creating editable roads. These are routed just like local roads, but may be edited with the drawing tools in Street Atlas.

This option has a special format that is best demonstrated by example:

```
"road=I-599!limited!Beecher St.!major"
```

This option will cause any road named "I-599" to become a limited-access highway and any road named "Beecher St." to become a major connector. Note that roads that have had their types changed in this way are not editable in Street Atlas, so make sure they are where you want them before you change them, and make sure to keep a backup of your original road layer. Note that the ! is a shell metacharacter in bash and possibly other shells, so you may have to use single quotes or some other escape mechanism.

There is a tutorial on how to create an onramp for a limited access highway in Street Atlas USA using GPSTabel. [<http://www.gpsbabel.org/formats/delorme-new-interstate/>]

## nogc option

Do not add geocache data to description.

If your original data contains geocaching-specific information such as difficulty and terrain, GPSTabel will automatically include that information in the waypoint descriptions in the generated drawing file. If you do not want that, specify the "nogc" option on the command line:

```
gpsbabel -i gpx -f 12345.gpx -o an1,nogc -F 12345.an1
```

## nourl option

Do not add URLs to description.

If your original waypoint data contains URLs, GPSTabel will include them as links in the generated drawing file. This causes the waypoint symbol to have a blue border, and it causes the waypoint text to be drawn in blue with an underline.

If you do not want this behavior, specify the "nourl" option on the command line:

```
gpsbabel -i gpx -f 12345.gpx -o an1,nourl -F 12345.an1
```

## deficon option

Symbol to use for point data.

This option allows you to specify which symbol to use for points that don't have a symbol already. It defaults to "Red Flag" but it accepts any symbol name you can put in a DeLorme export file. To find the name of a specific symbol in Street Atlas, let the mouse pointer hover over it for a few seconds and the name will be displayed.

## color option

Color for lines or mapnotes.

This option allows you to specify the color for line or mapnote data. It accepts color names of the form "#FF0000" (red) or any of the color names from the Cascading Style Sheets (CSS) specification.

## zoom option

Zoom level to reduce points.

This option specifies at what zoom level Street Atlas will begin showing reduced versions of your symbols. The default is 10. Setting zoom to 0 will disable this feature. Setting it to anything but the default will override the zoom level specified on any waypoints that were read from an existing anl file; this is by design.

## wpt\_type option

Waypoint type.

This option specifies how to represent point data in the draw file. Valid waypoint types are "symbol", "text", "mapnote", "circle", and "image". The default is "symbol".

If you specify a waypoint type of "image", you should make sure that the icon descriptions of your waypoints are the full names, including drive letters and full path, of image files in a format that works with your DeLorme product. Note that this means that the .anl file you generate will not work on any computer that does not have those images in the same place; this is part of the design of the anl format and cannot be avoided.

## radius option

Radius for circles.

If the waypoint type is "circle", the "radius" option specifies the radius of the circles. By default, this is in miles, but it may be specified in kilometers by adding a 'k'. The default radius is 1/10 mile.

## DeLorme GPL (gpl)

This format can...

- read and write tracks

This is the 'gpl' format as used in Delorme mapping products. It is a track format and contains little more than the tracklog of a GPS that was attached while driving. [frontiernet.net \[http://www.frontiernet.net/~werner/gps/\]](http://www.frontiernet.net/~werner/gps/)

## DeLorme Street Atlas Plus (saplus)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.



This format is for Delorme Street Atlas USA 2004 Plus and later.

For geocachers importing data from a tool like GSAK or Spinner, import the file twice in XData. One will create a file with the Cache description as a hyperlink on the flag. This can clutter up the screen and when you try to zoom in, it causes problems. So the second one will only have a flag. Thus you can turn off and on which one you want to view. The first time you import the file, in the assign field types, check the circle above Full Name and then next. The second time you import the file do not check any circle and in the second to last column, change URL to none and then click next. Use the same name you used the first time but add -Flag to it.

## DeLorme Street Atlas Route (saroute)

This format can...

- read tracks

This format reads route files from many Delorme mapping products. It supports the anr, rte, and rtd formats as either tracks or routes.

All options only apply to route files from newer (anr) versions of DeLorme software; older versions didn't store the turn information with the route.

### turns\_important option

Keep turns if simplify filter is used.

This option only makes sense in conjunction with the 'simplify' filter. It ensures that the route simplification process will remove the points corresponding to turns only after it has removed all other route points.

### turns\_only option

Only read turns; skip all other points.

This option causes GPSBabel to read only the waypoints associated with named turns. This should create a list of waypoints that correspond to the itinerary from Street Atlas.

### split option

Split into multiple routes at turns.

This option causes GPSBabel to create separate routes for each street, creating a new route at each turn point. For obvious reasons, 'split' cannot be used at the same time as the 'turns\_only' or 'turns\_important' options.

### controls option

Read control points as waypoint/route/none.

This option lets you read the control points (start, end, vias, and stops) for your route as well as the route itself. The default for this option is 'none', which won't read the control points. You may also specify 'waypoints', which reads the control points as waypoints, or 'route', which creates an extra route named 'control points' containing just the control points in order. Note that if your goal is to create an arc or other CSV file, you should use 'none' (or not use this option, which is the same thing.)

## times option

Synthesize track times.

This option causes GPSBabel to read the route as if it were a track, synthesizing times starting from the current time, using the estimated travel times specified in your route file (you can change your travel speeds in the DeLorme product you used to create the route file.)

## DeLorme XMap HH Native .WPT (xmap)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Delorme TopoUSA/XMap Conduit is one of the bazillion CSV variants mentioned above. It's just like Delorme Streets & Atlas 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 withing TopoUSA. Done with help of Dan Edwards.

## DeLorme XMap/SAHH 2006 Native .TXT (xmap2006)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Delorme XMap2006 Conduit is just like XMap , except there are no spaces between fields and the coordinate format is slightly different. The completely pointless header and footer lines are the same, at least. Use this to create the XMapHHWptsSend.txt file needed to sync to Street Atlas Handheld 2006.

Note that in order to keep from creating duplicates on your handheld, you must first remove the file "XMapWptsDB" from your handheld, restart SAHH2006 on the handheld to create an empty database, and THEN sync the new file.

## DeLorme XMat HH Street Atlas USA .WPT (PPC) (xmapwpt)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Delorme XMapHandHeld Street Atlas USA is another of the billion CSV variants. This is the format used by XmapHH SA USA on (at least) PocketPC O/S.

This XMap is not the same as the simpler XMap format, which is used with Topo USA 4.0 and XMapHH for Palm.

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

XMap on the PocketPC stores its 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, this is not very efficient.

As writing multiple waypoint files is outside of the scope of GPSTabel, GPSTabel 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 chapter.

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.

```
#!/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. xmap1_split.pl points.wpt /tmp/points GPST)\n";
    print " (created GPST0001-GPSTXXXX " .
        " in /tmp/points/ from points.wpt)\n";
    exit;
}

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

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

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

    print OUTFILE $_;

    close(OUTFILE);
}
```

```
}  
  
exit;
```

Contributed to GPSTracker by Alex Mottram.

## Destinator Itineraries (.dat) (destinator\_itn)

This format can...

- read and write routes

Support for Destinator™ itinerary files.

These have (mostly) extension .dat and are binary files. The file structure is undocumented and so this format was reverse engineered from some .dat files. At this time we can read and write name, comment and the coordinates of the route points.

Destinator™ by Destinator Technologies [<http://www.destinatortechnologies.net>] is a software for PNDs, Smartphones and PDAs.

```
gpsbabel -i destinator_itn -f from_A_to_B.dat -o gpx -F from_A_to_B.gpx
```

## Destinator Points of Interest (.dat) (destinator\_poi)

This format can...

- read and write waypoints

Support for Destinator™ binary POI files (.dat).

The basic information was found at mozsoft.com [<http://mozsoft.com/d3log.html>]. GPSTracker can read and write all fields described at this document. Please note that 'house number' isn't supported as a separate field. This field, if available in any source file, will be stored together with 'street' into GPSTracker's internal 'address' field.

Destinator™ by Destinator Technologies [<http://www.destinatortechnologies.net>] is a software for PNDs, Smartphones and PDAs.

```
gpsbabel -i destinator_poi -f interesting_places.dat -o gpx -F interesting_places.gpx
```

## Destinator TrackLogs (.dat) (destinator\_trl)

This format can...

- read and write tracks

Support for Destinator™ binary tracklogs (.dat).

The basic information was found at mozoft.com [<http://mozoft.com/d3log.html>]. In addition to the standard GPS track data of coordinates and timestamp, this format also stores the position fix and the number of satellites seen during recording.

Destinator™ by Destinator Technologies [<http://www.destinatortechnologies.net>] is a software for PNDs, Smartphones and PDAs.

```
gpsbabel -i destinator_tr1 -f last_trip.dat -o gpx -F last_trip.gpx
```

## EasyGPS binary format (easygps)

This format can...

- read and write waypoints

This is the binary file format used by EasyGPS [<http://www.easygps.com/>] format is seemingly being phased out in favor of GPX in newer versions of EasyGPS, but this allows conversions to and from the old binary .loc format.

Information about and sketchy code to implement this file format were provided by Eric Cloninger.

## Embedded Exif-GPS data (.jpg) (exif)

This format can...

- read waypoints

This format reads GPS information embedded in EXIF [<http://www.exif.org>], the Exchangeable Image Format, data. EXIF is a standardized method of encoding data in pictures such as JPEG, TIFF, and WAV and is frequently used by mobile phones with cameras, cameras with built-in GPS.

EXIF is frequently used for Geolocating photographs so their images can be correlated with time and location.

### filename option

Set waypoint name to source filename..

With this default option waypoint names are generated from source filename.

```
gpsbabel -i exif -f "C:\Pictures\IMG_1199.JPG",filename=Y -o gpx -F OUT.GPX
```

The resulting waypoint in OUT.GPX has name IMG\_1199.

## FAI/IGC Flight Recorder Data Format (igc)

This format can...

- read and write tracks
- read and write routes

FAI/IGC Data File -- Used by the international gliding community to record gliding flights. IGC files can be converted to and from tracks representing recorded flights, and routes representing task declarations in other formats.

## IGC Data Format Notes

Refer to Appendix 1 of [http://www.fai.org:81/gliding/gnss/tech\\_spec\\_gnss.asp](http://www.fai.org:81/gliding/gnss/tech_spec_gnss.asp) for the specification of the IGC data format.

A sample list of software applications that use data in IGC format can be found at [http://www.fai.org:81/gliding/gnss/gnss\\_analysis\\_software.pdf](http://www.fai.org:81/gliding/gnss/gnss_analysis_software.pdf)

GPSBabel can be used to translate data in IGC format to and from various other formats.

Routes in other formats are used to represent IGC task declarations.

Tracks in other formats are used to represent IGC recorded flights.

## Converting to IGC format

IGC files generated by GPSBabel will NOT pass security validation tests since the data they contain cannot be proven to originate from an approved flight recorder. For most software applications that use IGC files this is not an issue but for competition scoring, record and badge claims the generated files will not be accepted as proof of a flight.

A track stored in another format (GPX for example) representing a recorded flight can be converted into an IGC file:

```
gpsbabel -i gpx -f mytrk.gpx -o igc -F myflight.igc
```

If multiple track segments are provided in the input file, the one with the most points will be used.

A route stored in another format representing a task declaration can be converted into an IGC file:

```
gpsbabel -i gpx -f myrte.gpx -o igc -F mytask.igc
```

A route and a track in other formats can be included into a single IGC file:

```
gpsbabel -i gpx -f mytrk.gpx -f myrte.gpx -o igc -F myflight.igc
```

A similar result can be obtained by downloading the track log and routes directly from a GPS device connected to a PC. For example to create an IGC file from data recorded in a Garmin GPS connected to the first serial port of a PC running Linux:

```
gpsbabel -t -r -i garmin -f /dev/ttyS0 -o igc -F myflight.igc
```

For Windows operating systems:

```
gpsbabel -t -r -i garmin -f com1 -o igc -F myflight.igc
```

A waypoint file in another format containing a waypoint whose short name is "PILOT" can be merged into an IGC file. The description field of the waypoint will be used for the pilot name in the IGC file header:

```
gpsbabel -i gpx -f mytrk.gpx -f myrte.gpx -f mywpt.gpx -o igc -F myflight.igc
gpsbabel -w -t -r -i garmin -f /dev/ttyS0 -o igc -F myflight.igc
```

Some formats such as GPX allow routes, tracks and waypoints to exist in the same file and can be used to fully populate an IGC file:

```
gpsbabel -i gpx -f myall.gpx -o igc -F myflight.igc
```

## Converting from IGC format

Data in an IGC file can be converted into other formats. For example to generate OziExplorer files containing tracks representing the recorded flight (myozi.plt) and routes representing declared tasks (myozi.rte):

```
gpsbabel -i igc -f myflight.igc -o ozi -F myozi
```

Or to GPX format:

```
gpsbabel -i igc -f myflight.igc -o gpx -F myflight.gpx
```

Header information from the IGC file will be written to the description field of the track(s).

If both pressure altitude and GNSS altitude are recorded in the IGC file, two tracks will be written to the new track file, representing the two altitude tracks. The latitude, longitude and timestamps in the tracks will be identical.

## Merging into IGC format

A route stored in another format can be merged with an existing IGC file that has no task declaration, to generate a new IGC file with a task declaration:

```
gpsbabel -i igc -f myflight.igc -i gpx -f myrte.gpx -o igc -F mynew.igc
```

A two dimensional (lat/lon) track recorded during a flight by a GPS receiver can be merged with a one dimensional (altitude) track recorded during the same flight by a barograph instrument. The result is a three dimensional IGC file representing the flight:

```
gpsbabel -i gpx -f baro.gpx -i igc -f my2D.igc -o igc -F my3D.igc
```

The same can be achieved by downloading directly from a barograph instrument supported by GPSSbabel. For example with a Brauniger IQ Comp GPS variometer:

```
gpsbabel -i baroiq -f /dev/ttyS0 -i igc -f my2D.igc -o igc,timeadj=auto -F my3D.igc
```

or:

```
gpsbabel -i baroiq -f com1 -i igc -f my2D.igc -o igc,timeadj=auto -F my3D.igc
```

(Documentation contributed by Chris Jones, Aug 2004)

## timeadj option

(integer sec or 'auto') Barograph to GPS time diff.

Sometimes there is a discrepancy between the internal clock in the barograph instrument and GPS time which can result in the altitude and ground positions not correlating correctly. This can be corrected manually by passing the time difference in seconds between the two time domains through the "timeadj" parameter. This can be any positive or negative integer:

```
gpsbabel -i gpx -f baro.gpx -i igc -f my2D.igc -o igc,timeadj=27 -F my3D.igc
```

GPSTimeBabel can also attempt to deduce the time difference automatically. This is done by comparing the time that it thinks that you landed on the GPS track and the barograph and adjusting accordingly:

```
gpsbabel -i gpx -f baro.gpx -i igc -f my2D.igc -o igc,timeadj=auto -F my3D.igc
```

## Franson GPSTimeBabel Simulation (gpssim)

This format can...

- write waypoints
- write tracks
- write routes

This is a write-only format used to feed waypoints, tracks, and routes into Franson Technologies' [<http://franson.com/>] GpsGate simulator [<http://franson.com/gpsgate/>].

To use these files in GpsGate, select 'Simulator' and then "File->Open".

## wayptspd option

Default speed for waypoints (knots/hr).

This option specifies the speed of the simulation in knots.

## split option

Split input into separate files.



When this option is specified, GPSTabel will split the output into multiple files using the output filename as a base. For example, if you specify an output file of 'mytrip',

mytrip-waypoints.gpsim - will contain the waypoints.  
mytrip-track0000.gpsim - will contain the first track.  
mytrip-track0001.gpsim - will contain the second track.  
... and so on.  
mytrip-route0000.gpsim - will contain the first route.  
mytrip-route0001.gpsim - will contain the second route.  
... and so on.

Valid values for this option are 0 (off) and 1 (on). The default is '0'.

## Fugawi (fugawi)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

This was a requested CSV format, and is *not* the proprietary binary format used by Fugawi [<http://www.fugawi.com>]. Like any other CSV format, GPSTabel cannot read tracks in this format, but converting a track into it and then importing as track in Fugawi works.

It is known to work with Fugawi V3.1.4.635. When importing/exporting waypoints, one has to specify the order of fields as follows (names of fields may depend on the language used by Fugawi):

- Name
- Comment
- Description
- Latitude
- Longitude
- Altitude (metres)
- Date (yyyymmdd/yymmdd)
- Time of day (hhmmss)

When importing tracks, use "[ignore]" instead of "Name", "Comment" and "Description".

<http://www.fugawi.com/>

## G7ToWin data files (.g7t) (g7towin)

This format can...

- read waypoints
- read tracks
- read routes

Like GPSTabel, G7ToWin [<http://www.gpsinformation.org/ronh/g7towin.htm>] is a program which allows uploading and downloading information from several GPS devices (Garmin, Lowrance/Eagle, Magellan). G7ToWin has its own data format, which is an enhanced format used in Gardown.

This format can read both file types, G7ToWin (.g7t) and Gardown (.gdn).

## Garmin 301 Custom position and heartrate (garmin301)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

This is a very simple format that is most useful for exporting data from units that support heart rate data such as Garmin Forerunner 301™, Garmin Forerunner 305™, and Garmin Edge 305™, and to other programs for analysis. It's a simple comma delimited format that includes the timestamp, 3D position information and heart rate so you can pull it into a spreadsheet or graphing program.

## Garmin Logbook XML (glogbook)

This format can...

- read and write tracks

This is the XML format used by the Garmin Logbook product that ships with Forerunner and Foretrex. As of early 2006, this program is apparently been discontinued in favor of Garmin Training Center. See: <http://www.garmin.com>

## Garmin MapSource - gdb (gdb)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

Support for the "Garmin GPS Database" format used by default in MapSource™ versions since release 6.0 of that product. By default GPSTabel creates gdb files of version 2. Version 2 is used in Mapsource 6.3 and 6.5.

Garmin GPS database is an undocumented file format. The basic info for this module came from the existing MapSource conversion code.

## cat option

Default category on output (1..16).

This option specifies the default category for gdb output. It should be a number from 1 to 16.

## bitcategory option

Bitmap of categories.

This option is closely related to the 'category' option. While category allows you to choose a single category that waypoints should appear in, this options allows you to specify a bitmask to be used for the category. Options may be specified in either decimal or hex.

**Example 3.2. Example for gdb bitcategory option to put all waypoints in categories 1 and 16.**

The following two commands are equivalent. They place a the point in both the first and last of the sixteen available categories.

```
gpsbabel -i gpx -f PocketQuery.gpx -o gdb,bitcategory=32769 -F foo.gdb
gpsbabel -i gpx -f PocketQuery.gpx -o gdb,bitcategory=0x8001 -F foo.gdb
```

## ver option

Version of gdb file to generate (1..3).

This option specifies the data format version for the output file. Version 2 is the default. Currently, the only other valid values for this option are 1 and 3.

## via option

Drop route points that do not have an equivalent waypoint (hidden points).

This option instructs GPSTBabel to drop hidden (calculated) points from routes.

## roadbook option

Include major turn points (with description) from calculated route.

If this option is specified, GPSTBabel drops all calculated route points, with exception of points with a description (i.e. "Make U-turns until you know where you are."). The priority of this option is higher than of the `via` option. A value of 1 or y overwrites the `via` settings.

**Example 3.3. Using gdb option roadbook to create simple html roadbook**

```
gpsbabel -i gdb,roadbook -f sample.gdb -x nuketypes,waypoints,tracks -
x transform,wpt=rte -o html -F roadbook.html
```

Because gdb internally creates a route AND a waypoint list, you have to drop all waypoints and transform the route into waypoints in order to get a well ordered html output. We suggest these steps for all waypoint-only formats as html.

## Garmin MapSource - mps (mapsource)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

This format supports the Garmin Mapsource™ product family.

This format is based on significant reverse-engineering and guesswork. GPSTabel's output appears to be compatible with the various versions of MapSource. Icon mapping is attempted between different MapSource versions. Altitude is supported, but proximity and depth are not.

Naming files \*.mps will allow file->open in Mapsource to find the files more easily.

Versions 3, 4, and 5 of the Mapsource data format are handled automatically on input. By default the output is version 5. (Until 3/2004, it was version 3, but since Mapsource updates are free, the convenience of having modern icon sets outweighs the backward compatibility concern. Users of other versions can either upgrade or specify the switches to get output in a compatible format.) Waypoints, routes, and tracklogs are all handled, but map sets are ignored.

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

## **snlen option**

Length of generated shortnames.

This option specifies the length of generated short names on output. The default is 10 characters.

## **snwhite option**

Allow whitespace synth. shortnames.

This option specifies whether to allow whitespace (space, tab, etc.) in generated short names on output. The default is to not allow whitespace.

## **mpsverout option**

Version of mapsource file to generate (3,4,5).

This option specifies the format version for the output file. The default is version 5, as noted above. Supported versions are 3, 4, and 5.

## **mpsmergeout option**

Merge output with existing file.

This option causes the output to be merged with a pre-existing output file. This allows MapSource sections that aren't handled by GPSTabel (e.g. map sets) to be preserved.

## **mpsusedepth option**

Use depth values on output (default is ignore).

This option causes GPSTabel to write depth values for waypoints. Most input formats do not support depth values, so the default is to not write them.

## **mpsuseprox option**

Use proximity values on output (default is ignore).

This option causes GPSTabel to write proximity values for waypoints. Most input formats do not support proximity values, so the default is to not write them.

## Garmin MapSource - txt (tab delimited) (garmin\_txt)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

This is a textual format that contains nearly all of the information contained in the MapSource™ main format, GDB. This format also contains some computed values such as distances between routepoints and trackpoints, speed, and course (heading).

The main goal of garmin\_txt is to make aviation data more available. Because MapSource™ supports only the export, GPSTabel gives you the possibility to bring aviation data into MapSource™.

During the export with MapSource™, some fields are written using local settings of MapSource™ and Windows. These include grid format, gps datum, distance and temperature units, and the representation of date and time fields. GPSTabel tries to read all items automatically. Problems with date and time format can be solved with the 'date' and 'time' options.

### Example 3.4. Command showing garmin\_txt output with all options

```
gpsbabel -i garmin_txt,date="MM/DD/YYYY",time="hh:mm:ss xx"
-f in.txt -o garmin_txt,date="DD.MM.YYYY",datum="WGS
72",dist=m,prec=6,temp=c,time="HH:mm:ss",utc=+2 -F out.txt
```

## date option

Read/Write date format (i.e. yyyy/mm/dd).

This option specifies the input and output format for the date. The format is written similarly to those in Windows. An example format is "YYYY/MM/DD".

## datum option

GPS datum (def. WGS 84).

This option specifies the datum to be used on output. Valid values for this option are listed in Appendix A, *Supported Datums*.

## dist option

Distance unit [m=metric, s=statute].

This option specifies the unit to be used when outputting distance values. Valid values are M for metric (m/km/kph) or S for statute (ft/mi/mph).

## grid option

Write position using this grid..

This value specifies the grid to be used on write.

**Table 3.1. Grid values for garmin\_txt**

# idx	short	file-header	sample
0	ddd	Lat/Lon hddd.ddddd	S26.25333 E27.92333
1	dmm	Lat/Lon hddd°mm.mm	N33 56.539 W118 24.471
2	dms	Lat/Lon hddd°mm'ss.s	S25 25 26.8 E28 06 07.3
3	bng	British National Grid	TQ 18919 69392
4	utm	Universal Transverse Mercator	33 U 318293 5637154
5	swiss	Swiss grid	776519 167359

Idx or short are valid params for this option.

## prec option

Precision of coordinates.

This option specifies the precision to be used when writing coordinate values. Precision is the number of digits after the decimal point. The default precision is 3.

## temp option

Temperature unit [c=Celsius, f=Fahrenheit].

This option specifies the unit to be used when writing temperature values. Valid values are C for Celsius or F for Fahrenheit.

## time option

Read/Write time format (i.e. HH:mm:ss xx).

This option specifies the input and output format for the time. The format is written similarly to those in Windows. An example format is "hh:mm:ss xx".

## utc option

Write timestamps with offset x to UTC time.

This option specifies the local time zone to use when writing times. It is specified as an offset from Universal Coordinated Time (UTC) in hours. Valid values are from -23 to +23.

## Garmin PCX5 (pcx)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

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. In Mapsource, use file->import to read these files. If you name the files \*.wpt, Mapsource will find them more easily.

In general, you should prefer the "mapsource" file format to this one.

This format has been extended to handle many - but not all - files from GPS Utility [<http://www.gpsu.co.uk/index.html>]. If you encounter something that GPSTools does not handle well, use the free version of GPSTools to read it and save as something more common.

## deficon option

Default icon name.

The deficon option is used to control the icon output when writing to this format. It overrides any icon informatino that might be present in the source data.

## cartoexploreur option

Write tracks compatible with Carto Exploreur.

Carto Exploreur requires a slightly incompatible variation of the PCX format when written. Specifying this option on write tells us to create that strain of PCX.

## Garmin POI database (garmin\_poi)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

The Garmin POI loader [<http://www.garmin.com/support/agree.jsp?id=927>] loads custom points of interest into certain models of Garmin GPS receivers. (As of this writing, only the models introduced in 2005 and later are supported. See Garmin's site for more info.) This is the format readable that that program.

## Garmin Points of Interest (.gpi) (garmin\_gpi)

This format can...

- read and write waypoints

The format garmin\_gpi supports the binary POI (.gpi) files that are useable on newer Garmin GPS receivers. See garmin\_poi for additional information about Garmin's own Poiload program. Garmin POI-Loader [<http://www.garmin.com/support/agree.jsp?id=927>] is the standard application that creates GPI files with all possible features.

The layout of GPI files isn't documented and our module was created via reverse engineering. If you get a problem on reading or writing a GPI file, please provide that file (mailto:gpsbabel-misc@lists.sourceforge.net).

At this time we don't support special features as "Tour-Guide" or links to sounds and pictures.

## Important

Creation timestamp issue: See option sleep !!!

This module does not support direct transfer of .GPI files to receivers in Garmin protocol mode. For units like Nuvi, Zumo, or Streetpilot, just choose a file that's on the drive where your GPS is mounted. For units like the X series (GPSMap 60CSx, GPSMap 60Cx, Legend Hcx, etc.) you must explicitly put the unit in mass storage mode or mount the memory chip in an external reader and transfer the file directly.

### Example 3.5. Command showing garmin\_gpi output example

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,category="Nice Restaurants",bitmap=restaurant.bmp,notes -F "My Points.gpi"
```

## alerts option

Enable alerts on speed or proximity distance.

Because speed isn't a real member of a normal waypoint, you can put the speed values into the waypoint names. "Point@30" will result in a speed value of 30. By default we assume these values are in kilometers per hour.

Proximity distance is also supported by GPX, Garmin GDB, OZI Explorer, CompeGPS and Universal CSV.

```
gpsbabel -i gpx -f "warnings.gpx" -o garmin_gpi,alerts=1 -F "warnings.gpi"
```

## bitmap option

Use specified bitmap on output.

The bitmap (BMP) should be 24x24 (or smaller) and can be in RGB-colors (24- and 32-bit) or 8-bit indexed color format.

A color value of 0xFF00FF (blue=255, green=0, red=255), also called "Magenta", can be used for transparent areas.

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,bitmap="tux.bmp" -F "My Points.gpi"
```

## category option

Default category on output.



With this option you can specify the category which is primary visible on the device (default is "My points").

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,category="Best Restaurants" -F "My Points.gpi"
```

## hide option

Don't show gpi bitmap on device.

For a large list of points (or whatever) it can be useful when no bitmaps are displayed on device. With this option no bitmap is stored and displayed.

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,hide -F "My Points.gpi"
```

## descr option

Write description to address field.

The GPI address field is often visible in lists on the device. Use this option if you want to see the waypoint description (which can be an address too) in this lists.

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,descr -F "My Points.gpi"
```

## notes option

Write notes to address field.

The GPI address field is often visible in lists on the device. Use this option if you want to see the waypoint notes (which can be an address too) in this lists.

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,notes -F "My Points.gpi"
```

## position option

Write position to address field.

The GPI address field is often visible in lists on the device. Use this option if you want to see the waypoint position (coordinates) in this lists.

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,position -F "My Points.gpi"
```

## proximity option

Default proximity.

When no proximity data is available in the source input, GPSTabel uses this as the default proximity value. The parameter has to be in meters, or, when units=s specified, in miles. alerts are automatically enabled.

### Example 3.6. Read GPX file, create GPI to alert when you're 1/2 mile from a speed camera.

```
gpsbabel -i gpx -f "SpeedCameras.gpx" -o garmin_gpi,units=s,proximity=0.5 -F "SpeedCameras.gpi"
```

Its also possible to append a specific distance unit to the parameter.

```
gpsbabel -i gpx -f "SpeedCameras.gpx" -o garmin_gpi,proximity=500m -F "SpeedCameras.gpi"
```

**Table 3.2. Supported distance units**

Unit	Description
fa	Fathoms
feet	Feet
ft	Feet
km	Kilometers
m	Meters
mi	Miles
nm	Nautical miles

## sleep option

After output job done sleep n second(s).

The Garmin units seem to use the creation timestamp of GPI files for internal purposes. In other words, if you load GPI files with same creation timestamp on your device, strange things will happen, such as having missing or repeated POIs. With the sleep option, GPSTBabel waits a given number of seconds after the GPI file was written.

In the normal case of using GPSTBabel from the command line or from the GUI, the chance of creating files with the same timestamp is in the nearly ZERO. In scripts or batch files where you are writing multiple files - even from different GPSTBabel instances - the odds of this happening is rather good. The sleep option forces GPSTBabel to wait after creating a file to ensure the timestamps are unique. Values are specified in seconds and can be 1 or more.

```
gpsbabel -i gpx -f "SpeedCameras.gpx" -o garmin_gpi,sleep=1 -F "SpeedCameras.gpi"
```

## speed option

Default speed.

When no speed data is available in the source input, GPSTBabel uses this as the default speed value. The parameter has to be in kilometers per hour, or, when units=s specified, in miles per hour. alerts are automatically enabled.

```
gpsbabel -i gpx -f "SpeedCameras.gpx" -o garmin_gpi,units=s,speed=30 -F "SpeedCameras.gpi"
```

It's also possible to append a specific speed unit to the parameter.

```
gpsbabel -i gpx -f "SpeedCameras.gpx" -o garmin_gpi,speed=30mph -F "SpeedCameras.gpi"
```

**Table 3.3. Supported speed units**

Unit	Description
km/h	Kilometers per hour
kmh	Kilometers per hour
kph	Kilometers per hour
kt	Knots
knot	Knots
m/s	Meters per second
mps	Meters per second
mi/h	Miles per hour

## unique option

Create unique waypoint names (default = yes).

Don't create unique names sample:

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,unique=0 -F "My Points.gpi"
```

## units option

Units used for names with @speed ('s' for statute or 'm' for metric).

Sample command tells GPSTabel to handle speed values in miles per hour:

```
gpsbabel -i gpx -f "My Points.gpx" -o garmin_gpi,units=s -F "My Points.gpi"
```

## Garmin serial/USB protocol (garmin)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

GPSTabel supports a wide variety of Garmin hardware via serial on most operating systems and USB on Windows, Linux, and OS X.

For serial models, be sure the GPS is set for "Garmin mode" in setup and that nothing else (PDA hotsync programs, gpsd, getty, pppd, etc.) is using the serial port.

Supported Garmin GPS receivers with USB include

Astro	Forerunner 301	GPSMAP 60CSx	StreetPilot 2620
Edge 205	Forerunner 305	GPSMAP 60Cx	StreetPilot 2650
Edge 305	Foretrex 201	GPSMAP 76C	StreetPilot 2720
eTrex Legend C	Foretrex 301	GPSMAP 76CS	StreetPilot 2730
eTrex Legend Cx	GPS 18 <sup>1</sup>	GPSMAP 76CSX	StreetPilot 2820
eTrex Legend HCx	GPSMAP 195	GPSMAP 76Cx	StreetPilot 7200
eTrex Summit Cx	GPSMAP 276C	GPSMAP 96	StreetPilot 7500
eTrex Summit HC	GPSMAP 295	GPSMAP 96C	StreetPilot c310
eTrex Venture C	GPSMAP 296C	Quest	StreetPilot c320
eTrex Venture Cx	GPSMAP 378	Quest II	StreetPilot c330
eTrex Venture HC	GPSMAP 396	Rhino 520	StreetPilot c340
eTrex Vista C	GPSMAP 478	Rhino 530	StreetPilot i2
eTrex Vista Cx	GPSMAP 496	Rhino 520 HCx	StreetPilot i3
eTrex Vista HCx	GPSMAP 60C	Rhino 530 HCx	StreetPilot i5
Forerunner 205	GPSMAP 60CS	StreetPilot 2610	

the following Bluetooth Garmin products:

GPS 10<sup>1</sup>

and most serial Garmin GPS receivers including:

eMap	eTrex H	GPS 12	Rhino 110
eTrex Camo	Forerunner 201	GPS 12XL	Rhino 120
eTrex Legend	Foretrex 201	GPS III	Rhino 130
eTrex Summit	Geko 201	GPS III+	StreetPilot III
eTrex Venture	Geko 301	GPS II	StreetPilot III+
eTrex Vista	GPS 12CX	GPS II+	
eTrex (Basic Yellow)	GPS 12Map	GPS V	

The following Garmin GPS receivers are supported, but they do not support Garmin communication protocol and don't work with the `garmin` option. To use these receivers, read or write GPX files from the mass storage device as mounted on your computer.

Colorado 300 <sup>2</sup>	Nuvi 250W <sup>2</sup>	Nuvi 650 <sup>2</sup>	StreetPilot c510 <sup>2</sup>
Colorado 400c <sup>2</sup>	Nuvi 255W <sup>2</sup>	Nuvi 650FM <sup>2</sup>	StreetPilot c530 <sup>2</sup>
Colorado 400i <sup>2</sup>	Nuvi 260 <sup>2</sup>	Nuvi 660 <sup>2</sup>	StreetPilot c550 <sup>2</sup>
Colorado 400t <sup>2</sup>	Nuvi 260W <sup>2</sup>	Nuvi 670 <sup>2</sup>	StreetPilot c580 <sup>2</sup>
Nuvi 200 <sup>2</sup>	Nuvi 270 <sup>2</sup>	Nuvi 680 <sup>2</sup>	Zumo 450 <sup>2</sup>
Nuvi 205 <sup>2</sup>	Nuvi 300 <sup>2</sup>	Nuvi 750 <sup>2</sup>	Zumo 500 <sup>2</sup>
Nuvi 200W <sup>2</sup>	Nuvi 310 <sup>2</sup>	Nuvi 760 <sup>2</sup>	Zumo 550 <sup>2</sup>
Nuvi 205W <sup>2</sup>	Nuvi 350 <sup>2</sup>	Nuvi 770 <sup>2</sup>	
Nuvi 250 <sup>2</sup>	Nuvi 370 <sup>2</sup>	Nuvi 780 <sup>2</sup>	
Nuvi 255 <sup>2</sup>	Nuvi 600 <sup>2</sup>	Nuvi 880 <sup>2</sup>	

None of the GPSTabel developers has access to every model on that list, but we've received reports of success and/or have reasonable expectations that the above models work. If you succeed with a model

that is not on that list, please send a message to the gpsbabel-misc mailing list with the details so that we may add it.

Not every feature on every model is supported. For example, while we do extract data such as heart rate and temperature from tracks on the sporting models like Edge and Forerunner, GPSTabel is not a fitness program at its core and does not support features like courses or calorie/fitness zone data.

To communicate with a Garmin GPS serially, use the name of that serial port such as COM1 or /dev/cu.serial.

To communicate via USB use `usb:` as the filename on all OSes. Thus, to read the waypoints from a Garmin USB receiver and write them to a GPX file:

```
gpsbabel -i garmin -f usb: -o gpx -F blah.gpx
```

If you have multiple units attached via USB, you may provide a unit number, with zero being the implied default. So if you have three USB models on your system, they can be addressed as `usb:0`, `usb:1`, and `usb:2`. To get a list of recognized devices, specify a negative number such as:

```
gpsbabel -i garmin -f usb:-1
```

When reporting problems with the Garmin format, be sure to include the full unit model, firmware version, and be prepared to offer debugging dumps by adding `-D9` to the command line, like:

```
gpsbabel -D9 -i garmin -f usb: -o gpx -F blah.gpx
```

Custom icons are supported on units that support that. Neither GPSTabel nor your firmware know what is associated with any given slot number. They don't know that the picture you placed in the first slot is a happy face, they only know they're in the lowest numbered slot. GPSTabel names them consistently with Mapsource, so they are named 'Custom 0' through 'Custom 511'.

For models where the connection on the GPS is a serial interface, be sure the GPS is set for "Garmin mode" in setup and that nothing else (PDA hotsync programs, `gpsd`, `getty`, `pppd`, etc.) is using the serial port.

For models connected via USB, we recommend use of the `usb:` filename. For this to work on Windows, you must install the Garmin driver. For Linux, this will fail if have the `garmin_gps` kernel module loaded. See the Operating System Notes [/osnotes.html] for details.

This module also supports realtime tracking which allows realtime position reports from a Garmin GPS receiver over USB or serial.

## Important

The following Garmin units do not follow the standard Garmin communications protocol and are *not supported* by GPSTabel.

Marine plotters:

GPSMap 420	GPSMap 450	GPSMap 530	GPSMap 545
GPSMap 430	GPSMap 520	GPSMap 535	GPSMap 550
GPSMap 440	GPSMap 525	GPSMap 540	GPSMap 555

The PDA products

iQue 3000  
iQue 3200  
iQue 3600  
iQue M3  
iQue M4  
iQue M5

## snlen option

Length of generated shortnames.

This option overrides the internal logic to figure out how many characters an addressed Garmin GPS will support when using the '-s' smartname option. This should be necessary only if you have a receiver type that GPSTabel doesn't know about or if you want to "dumb down" one unit to match another, such as wanting waypoint names in a StreetPilot 2720 (which supports 20 character names) to exactly match those in a 60CS (which supports 10).

## snwhite option

Allow whitespace synth. shortnames.

This options controls whether spaces are allowed in generated smart names when using the '-s' option.

## deficon option

Default icon name.

This option specifies the icon or waypoint type to write for each waypoint on output.

If this option is specified, its value will be used for all waypoints, not just those that do not already have descriptions. That is, this option overrides any icon description that might be in the input file.

Value specified may be a number from the Garmin Protocol Spec or a name as described in the Appendix B, *Garmin Icons*.

This option has no effect on input.

## get\_posn option

Return current position as a waypoint.

This options gets the current longitude and latitude from the attached GPS device and returns it as a single waypoint for further processing. For example, to return the current position from a USB Garmin to a KML file:

```
gpsbabel -i garmin,get_posn -f usb: -o kml -F myposition.kml
```

## power\_off option

Command unit to power itself down.

This command forces an immediate powerdown of the addressed Garmin receiver. It is ignored on hardware that does not support this command. Obviously, further processing once you have sent a "power off" command to a unit that supports it is rather futile, so place this option carefully in your command.

```
gpsbabel -o garmin,power_off -F /dev/ttyS0
```

## resetttime option

Sync GPS time to computer time.

This option is experimental and was added to solve a very specific problem. Certain Garmin units (the original black and white Vista is known to have this) will sometimes scramble their clock crazy far into the future (like 2066). When this happens, the GPS itself may or may not work and later conversations with GPSTBabel may fail as the time overflows the documented range. The use of `resetttime` brings the GPS's internal clock back close enough to reality that the GPS itself can then "fix" it when it has next a lock.

## category option

Category number to use for written waypoints.

This numeric option will force waypoints to be written with that category number when sending to a Garmin receiver that has category support. It is ignored on receivers without that capability.

## bitcategory option

Bitmap of categories.

This option is closely related to the 'category' option. While category allows you to choose a single category that waypoints should appear in, this options allows you to specify a bitmask to be used for the category. Options may be specified in either decimal or hex.

**Example 3.7. Example for garmin bitcategory option to put all waypoints in categories 1 and 16.**

The following two commands are equivalent. They place a the point in both the first and last of the sixteen available categories.

```
gpsbabel -i gpx -f PocketQuery.gpx -o garmin,bitcategory=32769 -F usb:
gpsbabel -i gpx -f PocketQuery.gpx -o garmin,bitcategory=0x8001 -F usb:
```

## Garmin Training Centerxml (gtrnctr)

This format can...

- write tracks

GPSTBabel has limited support for Garmin Training Center files. That program is the successor to Garmin' Logbook program for their workout units. It is a free upgrade.

This format is somewhat underachieving in GPSTBabel. It is a write-only format; we never read it. The bigger problem, however, is a fundamental impedance mismatch between this format and most of what we support. GPSTBabel fundamentally deals in waypoints, tracks, and routes. While we do record things like heart rate and temperature when we know it, the fundamentals of Training Center are different - it deals in concepts like laps and calories which are rather alien to GPSTBabel and most of the formats we

support. As such, while we can describe the tracks pretty accurately, things like calories and heart zone tracking are not supported.

## Geocaching.com .loc (geo)

This format can...

- read and write waypoints

This format supports the Geocaching.com [<http://www.geocaching.com>]/EasyGPS [<http://www.easygps.com>] ".loc" format. This format was created specifically for Geocaching.com and is not the same as the standard EasyGPS .loc format. See the EasyGPS or GPX formats for more general EasyGPS support.

This is a simple XML-based format containing only very basic information about geocaches. If you can use the GPX format instead, you should consider doing so as it is a much richer format.

## deficon option

Default icon name.

This option specifies the icon or waypoint type to write for each waypoint on output.

If this option is specified, its value will be used for all waypoints, not just those that do not already have descriptions. That is, this option overrides any icon description that might be in the input file.

There is no list of valid values for this option.

This option has no effect on input.

## nuke\_placer option

Omit Placer name.

If this option is specified, GPSBabel will not read geocache placer information from a .loc file on input. That is, it will ignore any placeer names in the input file.

This option has no effect on output.

## GeocachingDB for Palm/OS (gcdb)

This format can...

- read and write waypoints

This is format for the GeocachingDB [<http://vip.hyperusa.com/~dougs/geocachingdb/geocachingdb.htm>] program by DougsBrat. It works with v2 and v3 of this program.

## Geogrid Viewer tracklogs (.log) (ggv\_log)

This format can...

- read and write tracks

Binary track logs used by the Geogrid™-Viewer, a very popular product in Germany.



GPSTabel has full support for version 1.0 of this file format.

We can also read some GPS data (including coordinates) from version 2.5. But it seems, that this newer version no longer stores time stamps. This can be a problem when converting to other formats or if you want to use our track filter.

## GEOnet Names Server (GNS) (geonet)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Input support for the GEOnet Names Server (GNS) country file structure. Export to this format is not possible, as this format has too many fields that we never get populated by any other format.

## GeoNiche .pdb (geoniche)

This format can...

- read and write waypoints

Geoniche [<http://www.raydarllc.com/>] is a Palm/OS application oriented for the off-road user. This module was contributed by Rick Richardson.

## dbname option

Database name (filename).

This option specifies the database name for the output file. This name is not the same thing as the file name on your computer; this is the name that appears in the file browser on your handheld.

## category option

Category name (Cache).

This option specifies the name of the category in which to place the waypoints. If this option is not specified, the default category is "Cache".

## GlobalSat DG-100/BT-335 Download (dg-100)

This format can...

- read tracks

Serial download protocol for the GlobalSat DG-100™ GPS data logger. Although untested it is expected that this will also support the BT-335.

While the DG-100 has a button to record waypoints, they seem to be indistinguishable from trackpoints. Therefore, all points will be presented as trackpoints, disregarding whether they were recorded automatically or manually.

GlobalSat DG-100 [[http://www.globalsat.com.tw/eng/product\\_detail\\_00000090.htm](http://www.globalsat.com.tw/eng/product_detail_00000090.htm)]

**Example 3.8. Command showing DG-100 download and erase on Linux**

```
gpsbabel -t -i dg-100,erase -o gpx /dev/ttyUSB0 outputfile.gpx
```

The DG-100 provides a physical USB interface to the host computer, but internally it uses a Prolific PL-2303 chip to do this. So you must have drivers installed on your computer to recognize the PL-2303 and provide that data as a serial port to software like GPSBabel. Such software comes with the unit for Windows. Prolific provides software for Mac OS/X, but unfortunately their driver has a defect which makes it unusable with GPSBabel.

## erase option

Erase device data after download.

This option erases the track log from the device after download.

# Google Earth (Keyhole) Markup Language (kml)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

KML, the Keyhole Markup Language, is used by Keyhole and Google Earth [<http://earth.google.com>].

There are concepts in KML that GPSBabel can't support very well on read because they don't map well into other programs. For example, KML has ideas of camera views and names and descriptions can have arbitrarily complicated HTML in them. KML files may have tiered "Styles" which can identify sizing info and URLs of associated icons. Reading such files with GPSBabel - even if your goal is to output to KML - can often have surprising results. Simple files with waypoints and paths (which GPSBabel represents internally as tracks) work fine.

Google Earth also uses GPSBabel internally for receiver communications and several file format imports and exports.

In general, GPSBabel's KML writer is relatively strong. GPSBabel handles simple KML on read fairly well, but if you're dealing with handcrafted KML that uses extensive features that have no analog in other formats like nested folders, ringgeometry, camera angles, and such, don't expect GPSBabel to do well with them on read.

Google Earth 4.0 and later have a feature that can surprise users of this format. Earth's "time slider" feature controls what timestamped data gets displayed. If you're using data that has timestamps (e.g. GPX points that contain time or almost any track data) this will be important to you. The time slider defaults to the far left position and fully closed. This means that only the first data point will be displayed. You can tweak Earth's settings to "view->show time->never" or you can widen the time slider to show the range of data of interest.

See Google Earth's documentation on timelines [[http://earth.google.com/userguide/v4/ug\\_gps.com#timeline](http://earth.google.com/userguide/v4/ug_gps.com#timeline)] for more info.

## deficon option

Default icon name.

This option specifies the default name for waypoint icons

## lines option

Export linestrings for tracks and routes.

When this option is nonzero, GPSBabel draws lines between points in tracks and routes. The default value for this option is 1, which causes lines to be drawn by default. To disable line-drawing, specify `lines=0`.

## points option

Export placemarks for tracks and routes.

When this option is nonzero, GPSBabel draws placemarks for tracks and routes. The default value for this option is 1, which causes placemarks to be drawn. To disable drawing of placemarks, specify `points=0`.

## line\_width option

Width of lines, in pixels.

This option specifies the width of the drawn lines in pixels. The default value is six pixels.

## line\_color option

Line color, specified in hex AABBGRR.

This option specifies the line color as a hexadecimal number in AABBGRR format, where A is alpha, B is blue, G is green, and R is red.

## floating option

Altitudes are absolute and not clamped to ground.

When this option is nonzero, altitudes are allowed to float above or below the ground surface. By default, this option is zero so that altitudes are clamped to the ground. Specify `floating=1` to allow them to float.

This option is more useful to pilots than to hikers.

## extrude option

Draw extrusion line from trackpoint to ground.

This option is a boolean flag to specify whether Google Earth should draw lines from trackpoints to the ground. It defaults to '0', which means no extrusion lines are drawn. The option of '1' is, of course, most useful for points that aren't actually on the ground such as those captured from planes.

## trackdata option

Include extended data for trackpoints (default = 1).

This is a boolean flag that controls whether GPSTBabel writes extensive data for each trackpoint generated. By default computed speed, timestamps, and so on are written with the default of '1' for this option. If you are writing large tracks and do not value this information, you can reduce the size of the generated file substantially by turning this flag off by setting it to '0'.

## **trackdirection option**

Indicate direction of travel in track icons (default = 0).

If set, this options creates directional icons for trackpoints. Arrows will show the direction of travel on drawn tracks and routes.

## **units option**

Units used when writing comments ('s'tatute or 'm'etric).

Units is a simple option. Specify 's' for "statute" (miles, feet, and other things that don't sensibly convert to each other, but are craved by Americans) or 'm' for "metric".

## **labels option**

Display labels on track and routepoints (default = 1).

When this option is zero, no labels are added for track and route points. This option defaults to one, so labels are added by default.

## **max\_position\_points option**

Retain at most this number of position points (0 = unlimited).

This option allows you to specify the number of points kept in the 'snail trail' generated in the realtime tracking mode.

# **Google Maps XML (google)**

This format can...

- read tracks

This format is designed to read the XML emitted when you tack "&output=js" onto the end of a Google Maps [<http://maps.google.com>>Google Maps] route URL (use the "link to this page" option to get a usable URL.) This allows you to plan a route using Google Maps, then download it and use it in your own mapping program or GPS receiver. To get a file suitable for use with GPSTBabel, plan your route as usual with Google Maps. Once you've got it the way you want it, click the "Link to this page" link in the upper right-hand corner of the Google Maps page. Then, edit the URL that appears in your address bar by adding "&output=js" (without the quotes) onto the end. Hit enter, and the resulting page will be mostly empty. It doesn't look like much, but it contains exactly what GPSTBabel needs. Save it to disk using whatever menu option your web browser provides.

Note that if you are using Microsoft Internet Explorer, you should make sure to save the web page as "Web Page, HTML Only". If you save it as "Web Page, Complete", it will be reformatted into a non-XHTML format that GPSTBabel cannot read.

If you use a Unix-compatible operating system, this shell script might be useful:

```
#!/bin/sh
FROM="233 S. Upper Wacker Dr, Chicago, IL"
TO="1060 W. Addison St, Chicago, IL"
wget -O - "http://maps.google.com/maps?q=$FROM to $TO&output=js" \
2>/dev/null >google_map.js
gpsbabel -i google -f google_map.js -o gpx -F google_map.gpx
```

## GpilotS (gpilots)

This format can...

- read and write waypoints

This is a Palm/OS file format for GPilotS [<http://www.cru.fr/perso/cc/GPilotS/>]. It was tested against version 6.2 of GPilotsS

Neither tracks nor routes are supported at this time.

### dbname option

Database name.

This option specifies the database name for the output file. This name is not the same thing as the file name on your computer; this is the name that appears in the file browser on your handheld.

## GPS TrackMaker (gtm)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

Input and output support for waypoints, tracks and routes in the GPS TrackMaker [<http://www.gpstm.com>] binary format.

Code implemented by Gustavo Niemeyer.

## GPSTBabel arc filter file (arc)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

This format is used by GPSTBabel itself as the input to the arc and polygon filters. See those filters for more information.

The arc format reads two numeric fields, a latitude and a longitude, in any format recognized as human readable and writes as simple degrees decimal. It really is intended for GPSTBabel's own internal use more than general use, though it turns out to be a convenient way of expressing simple polylines and polygons.

## GpsDrive Format (gpsdrive)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

GpsDrive way.txt file format. A space separated format file. Tested against GpsDrive v 1.30 found at [gpsdrive.de](http://www.gpsdrive.de) [http://www.gpsdrive.de]. Contributed by Alan Curry.

## GpsDrive Format for Tracks (gpsdrivetrack)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Format used by GpsDrive to save tracks. Like GPSDRIVE a space separated format file. See above for a link to GpsDrive. Contributed by Tobias Minich.

## GPSman (gpsman)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

GPS Manager [<http://www.ncc.up.pt/gpsman/>] can read and write formats GPSTabel doesn't understand. The format defaults (WGS84, DDD) work reliably. Tracks, routes, and non-default format options are not supported.

This format is documented at the GPS Manager doc site [[http://www.ncc.up.pt/gpsman/gpsmanhtml/manual/html/GPSMandoc\\_30.html](http://www.ncc.up.pt/gpsman/gpsmanhtml/manual/html/GPSMandoc_30.html)].

## GPSPilot Tracker for Palm/OS (gpspilot)

This format can...

- read and write waypoints

The file format for GPSPILLOT [gpspilot.com](http://www.gpspilot.com) [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.

## dbname option

Database name.

This option specifies the database name for the output file. This name is not the same thing as the file name on your computer; this is the name that appears in the file browser on your handheld.

## gpsutil (gpsutil)

This format can...

- read and write waypoints

The format we call `gpsutil` [<http://www.cs.uakron.edu/~hennings/gpsutil>] is a simple file format used by a program that runs on POSIX- compliant OSes like UNIX and Linux. Reads and writes of this format are very reliable. (The lead developer of GPSTabel [<http://people.robertlipe.html>] also contributed to this that 'gpsutil' the early days.)

Note that 'gpsutil' is a different format - and program - than the one called GPS Utility [<http://www.gpsu.co.uk/index.html>]; for that one, you should probably use our PCX module.

## GPX XML (gpx)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

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

GPSTabel's reader of this module attempts to preserve tags it doesn't really understand. It also tries to glean interesting data from pocket queries from Geocaching.com [<http://www.geocaching.com>] and Garmin's "gpxx" GPX extensions.

## snlen option

Length of generated shortnames.

When used with the `-s` to control shortnames, the `snlen` suboption to GPX controls how long the generated smartname will be. This can be useful for cases like writing GPX files to a GPS that has a fixed waypoint name length.

## suppresswhite option

No whitespace in generated shortnames.

When used with the `-s` to generate smart shortnames, this suboption controls whether whitespace is allowed in the generated shortnames.

## logpoint option

Create waypoints from geocache log entries.

When reading Groundspeak Pocket Queries [<http://www.geocaching.com>], the `logpoint` option creates additional waypoints from the log entries.

A typical use for this is to get coordinates read from "corrected coordinates" logs.

## urlbase option

Base URL for link tag in output.

This is a fairly esoteric option. If the GPX file you are reading has only base pathnames (e.g. "foo.html") the value you specify to this argument will be prepended to that. For example, "`-o gpx,urlbase=c:\My Documents\Whatever`" would result in the link to that waypoint being written to refer to `c:\My Document\Whatever\foo.html`

## gpxver option

Target GPX version for output.

This option specifies the version of the GPX specification to use for output. The default version is 1.0. The only other valid value for this option is 1.1.

Notice that this is not a full scale XML schema conversion. In particular, if you have a GPX 1.0 file that has extended namespaces in it (such as a pocket query from Geocaching.com) just writing it with this option will result in a horribly mangled GPX file as we can't convert the schema data.

## HikeTech (hiketech)

This format can...

- read and write waypoints
- read and write tracks

This is the .gps format used by the Mac OS X applications written by HikeTech. These include TopoDraw, Link2GPS, and GPSWrite. More information about these products can be found at [hiketech.com](http://www.hiketech.com) [<http://www.hiketech.com>]

## Holux (gm-100) .wpo Format (holux)

This format can...

- read and write waypoints

The Holux gm-100 (e-fox) gps receiver uses standard compact flash cards. File formats were provided by Holux-Taiwan [holux.com](http://www.holux.com.tw) [<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](mailto:jochen@bauerbahn.net).

When copying the .wpo file to a flash card, the file must be named `tempwprt.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". A practical matter, this loss is only about 1.7 meters (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.

## HSA Endeavour Navigator export File (hsandv)

This format can...

- read and write waypoints

HSA Systems Endeavour Navigator format - will import both the old version 4.x binary files, and the newer XML based ones. Only writes the new XML (5.0 and above) format. (use the .exp extension)

## HTML Output (html)

This format can...

- write waypoints

GPSTBabel's HTML output generates a single HTML file of all of the waypoints in the input file. It supports a number of Groundspeak GPX extensions and filters out potentially harmful HTML from the input file while maintaining almost all of the source HTML formatting. This makes this format well suited for generating HTML to hand to programs like Plucker for putting in a PDA and especially so for "paperless caching" for Geocachers with pocket queries.

This format is similar to the text format.

The following command line reads a GPX file with Groundspeak extensions and writes an HTML file with encrypted hints that is rendered using a custom stylesheet:

```
gpsbabel -i gpx -f 12345.gpx -o html,stylesheet=green.css,encrypt -F 12345.html
```

## stylesheet option

Path to HTML style sheet.

Use this option to specify a CSS style sheet to be used with the resulting HTML file.

## encrypt option

Encrypt hints using ROT13.

Use this option to encrypt hints from Groundspeak GPX files.

## logs option

Include groundpeak logs if present.

Use this option to include Groundspeak cache logs in the created document.

## degformat option

Degrees output as 'ddd', 'dmm'(default) or 'dms'.

When GPSTools writes coordinates, this option is consulted to see if it should write decimal degrees ('ddd') decimal minutes ('dmm') or degrees, minutes, seconds ('dms'). The default is 'dmm'.

## altunits option

Units for altitude (f)eet or (m)etres.

This option should be 'f' if you want the altitude expressed in feet and 'm' for meters. The default is 'f'.

## IGN Rando track files (ignrando)

This format can...

- read and write tracks

This format supports IGN Rando track files. IGN Rando is a program mainly used in France for Topo maps. The files are XML based and are "windows-1252" encoded. Trackpoints do not have time stamps.

## index option

Index of track to write (if more the one in source).

Because the format supports only one track, this option may be used on output to select a single track from a collection of tracks read from a more expressive format. If you have, say, a gpx file that contains two tracks, you may use this option to write them one at a time to individual files.

```
gpsbabel -i gpx -f tracks.gpx -o ignrando,index=1 -F track1.txt -o  
ignrando,index=2 -F track2.txt
```

## Kartex 5 Track File (ktf2)

This format can...

- read and write tracks

This format is derived from the xcsv format, so it has all of the same options as that format.

Support for Kartex 5 trackfiles. For more info see kwf2.

## Kartex 5 Waypoint File (kwf2)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Support for Kartex 5 waypoint files. Kartex is a Swedish map and GPS positioning system. GPSBabel can read and write files from Kartex 4 and 5 with WGS84 coordinates. UTM or Swedish grid are not supported.

## Kompass (DAV) Track (.tk) (kompass\_tk)

This format can...

- read and write tracks

This format is derived from the xcsv format, so it has all of the same options as that format.

This module supports track files used by Kompass and DAV "Deutscher Alpenverein".

Kompass [<http://www.kompass.at>] is a publishing company from Austria. If you want to get more information about DAV, the German alpine association, and if you are familiar with the german language, please have a look at their homepage [<http://www.alpenverein.de>].

## Kompass (DAV) Waypoints (.wp) (kompass\_wp)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

This module supports waypoint files used by Kompass and DAV "Deutscher Alpenverein".

Some more information under kompass\_tk format.

## KuDaTa PsiTrex text (psitrex)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

This is a text format created by KuDaTa's PsiTrex program for the Psion PDAs. The format can't be readily handled by XCSV, so this format is handled explicitly. Waypoints, routes and tracks are all handled, with icon names used corresponding to verison 1.13 of PsiTrex. This module was contributed to GPSBabel by Mark Bradley.

## Lowrance USR (lowranceusr)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

The Lowrance iFinder GPS [<http://www.lowrance.com>] series has the unique capability to output its data to an MMC card. The data is saved to the card as a .USR file and can be read by your computer using a card reader. Waypoints, icons, routes, tracks are supported. Event marker icons contain a symbol, name, latitude and longitude only. By default, Event marker icons are converted to waypoints on read. On write, you are able to create icons from waypoints.

## **ignoreicons option**

Ignore event marker icons on read.

This option instructs GPSTabel to not convert icons to waypoints on input, but to instead disregard them altogether

## **writeasicons option**

Treat waypoints as icons on write.

(USR output) This option converts the waypoint information to an event marker icon.

## **merge option**

(USR output) Merge into one segmented track.

(USR output) This option merges all tracks into a single track with multiple segments.

## **break option**

(USR input) Break segments into separate tracks.

(USR input) Break track segments into separate tracks. file.

# **Magellan Explorist Geocaching (maggeo)**

This format can...

- write waypoints

This format support the on-card format used by the Magellan [<http://www.magellangps.com>] Explorist 400, Explorist 500, Explorist 600, Explorist 210, and Explorist XL to describe geocaches. Notice what while the format can hold an infinite number of geocaches, the unit will read and silently discard all but 200 geocache POIs at a time.

You should name any file created with this format with a ".gs" extension so the firmware can read it.

# **Magellan Mapsend (mapsend)**

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

This format supports the Magellan [<http://www.magellangps.com>] MapSend™ native file format.

Kudos to Magellan for having the foresight to document their file formats, making software like this possible.

## trkver option

MapSend version TRK file to generate (3,4).

This option sets the MapSend version to generate TRK files, since new MapSend versions can't open version 3 files. Valid values are 3 (MapSend v3.0) or 4 (MapSend v4.0 and v4.1).

# Magellan NAV Companion for Palm/OS (magnav)

This format can...

- read and write waypoints

Magellan [<http://www.magellangs.com>] 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 and 3.20.

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.

## Magellan SD files (as for eXplorist) (magellanx)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

This is the SD card format used by the Magellan [<http://www.magellangps.com>] Explorist 400, Explorist 500, Explorist 600, and Explorist XL and internally on those devices plus the Explorist 210. Stored waypoints are identical to the Magellan SD format used by Meridian, but the newer models allow longer waypoint names. Routes are subtly different.

You should name any file containing waypoints created with this format with a ".upt" extension so the firmware can read it. Similarly, routes should be named ".rte" and tracks should be named ".log".

## deficon option

Default icon name.

The deficon option is used to control the icon output when writing to this format. It overrides any icon informatino that might be present in the source data.

## maxcmts option

Max number of comments to write (maxcmts=200).

The maxcmts option allows you to specify the number comments that will be sent to the unit.

Magellan receivers allow a maximum of 200 waypoint comments. Unfortunately, DirectRoute uses waypoint comments to provide next turn directions for navigation pop-ups and that comes from that pool of 200 comments. It is therefore sometimes convenient to limit the number of waypoint comments written to the receiver. For example, a geocacher might want to upload 400 waypoints, but only 190 with comments so that DirectRoute could provide driving directions for the next ten turns.

# Magellan SD files (as for Meridian) (magellan)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

GPSTBabel supports the following Magellan [<http://www.magellangps.com>] receivers:

310	Meridian Color
315	Explorist 100 (with aftermarket cable)
Map330	Explorist 200 (with aftermarket cable)
SporTrak Map Color	Explorist 300 (with aftermarket cable)
SporTrak Map	Explorist 210
SporTrak Map Pro	Explorist 300
SporTrak Map Topo	Explorist 400
Meridian (green or yellow)	Explorist 500
Meridian Gold	Explorist 600
Meridian Platinum	Explorist XL

This format is used for both the serial protocol used on the devices with serial ports such as Map330 and Meridian and for the files stored either in either the unit's internal memory (Explorist 210, Explorist 400, Explorist 500, Explorist 600, Explorist XL) or on removable memory.

If you specify a serial port for the file (.e.g. "COM1", "/dev/ttyS0") to be read or written, GPSTabel will use serial protocol. Specifying a file, either on local filesystem or on a mounted flash card reader, will result in the file-based format being used.

## deficon option

Default icon name.

The deficon option is used to control the icon output when writing to this format. It overrides any icon informatino that might be present in the source data.

## maxcmts option

Max number of comments to write (maxcmts=200).

The maxcmts option allows you to specify the number comments that will be sent to the unit.

Magellan receivers allow a maximum of 200 waypoint comments. Unfortunately, DirectRoute uses waypoint comments to provide next turn directions for navigation pop-ups and that comes from that pool of 200 comments. It is therefore sometimes convenient to limit the number of waypoint comments written to the receiver. For example, a geocacher might want to upload 400 waypoints, but only 190 with comments so that DirectRoute could provide driving directions for the next ten turns.

# Magellan serial protocol (magellan)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

GPSTabel supports the following Magellan [<http://www.magellangps.com>] receivers:

310	Meridian Color
315	Explorist 100 (with aftermarket cable)
Map330	Explorist 200 (with aftermarket cable)
SporTrak Map Color	Explorist 300 (with aftermarket cable)
SporTrak Map	Explorist 210
SporTrak Map Pro	Explorist 300
SporTrak Map Topo	Explorist 400
Meridian (green or yellow)	Explorist 500
Meridian Gold	Explorist 600
Meridian Platinum	Explorist XL

The RoadMate family of products is not supported.

This format is used for both the serial protocol used on the devices with serial ports such as Map330 and Meridian and for the files stored either in either the unit's internal memory (Explorist 210, 400, 500, 600, XL) or on removable memory.

If you specify a serial port for the file (.e.g. "COM1", "/dev/ttyS0") to be read or written, GPSTBabel will use serial protocol. Specifying a file, either on local filesystem or on a mounted flash card reader, will result in the file-based format being used.

Users of the Explorist generation of receivers should probably prefer to use the magellanx format over this one.

## Important

This module does not support the units that do not follow Magellan's documented communications protocols including:

Maestro 3100	RoadMate 800
Maestro 3140	RoadMate 860T
Maestro 3200	RoadMate 1200
Maestro 3210	RoadMate 1400
Maestro 3220	RoadMate 1412
Maestro 3225	RoadMate 1430
Maestro 3250	RoadMate 2000
Maestro 4000	RoadMate 2000
Maestro 4040	RoadMate 2200T
Maestro 4050	RoadMate 3000T
Maestro 4200	RoadMate 3050T
Maestro 4210	RoadMate 6000T
Maestro 4220	RoadMate AAA
Maestro 4250	Triton 200
Maestro 5310	Triton 300
RoadMate 300	Triton 400
RoadMate 360	Triton 500
RoadMate 500	Triton 1500
RoadMate 700	Triton 2000
RoadMate 760	

## deficon option

Default icon name.

This option specifies the icon or waypoint type to write for each waypoint on output.

If this option is specified, its value will be used for all waypoints, not just those that do not already have descriptions. That is, this option overrides any icon description that might be in the input file.

This option has no effect on input.

## maxcmts option

Max number of comments to write (maxcmts=200).

The maxcmts option allows you to specify the number comments that will be sent to the unit.

Magellan receivers allow a maximum of 200 waypoint comments. Unfortunately, DirectRoute uses waypoint comments to provide next turn directions for navigation pop-ups and that comes from that pool of 200 comments. It is therefore sometimes convenient to limit the number of waypoint comments written to the receiver. For example, a geocacher might want to upload 400 waypoints, but only 190 with comments so that DirectRoute could provide driving directions for the next ten turns.



## baud option

Numeric value of bitrate (baud=4800).

This option causes GPSBabel to use the given baud rate for serial communications. It must match the given baud rate on the receiver. The default value matches the default on the receiver, 4800.

Valid options are 1200, 2400, 4800, 9600, 19200, 57600, and 115200.

## noack option

Suppress use of handshaking in name of speed.

Magellan's protocol specification strongly encourages the use of software acknowledgements on every packets. This is a simple "this is what I think I heard. If you agree that I heard it correctly, let's go to the next packet" handshake that is used to ensure the integrity of the data transfer.

Certain firmware versions have problems handling this which makes transfers unnecessarily slow. Transfers on all units at high serial speeds are also severely restricted by this process.

In controlled environments (good cables, low electrical noise, receiving from the unit, not doing donuts with the unit set to "track up" at a 150 mile scale with 500 waypoints on the screen) it is sometimes useful to release that safety belt by using the "noack" suboption.

## nukewpt option

Delete all waypoints.

This option erases all waypoints in the receiver before doing a transfer.

This is a convenient option to use in automated processes where you want to ensure the units starts with a clean state before sending waypoints to it. Using this option on transmit is a better idea than doing it on receive since the latter would erase all the waypoints before asking the unit to send all the waypoints.

## MagicMaps IK3D project file (.ikt) (ik3d)

This format can...

- read waypoints
- read tracks

This is the format for MagicMaps [<http://www.magicmaps.de>] project (.ikt) files.

MagicMaps [<http://www.magicmaps.de>] "Das interaktive Kartenwerk"<sup>TM</sup> is a Software from Germany. It's a route-planning software with a 3-dimensional environment.

The project files are XML based and we can read the main GPS items (names and coordinates). For an output these files are too complex.

## Map&Guide 'TourExchangeFormat' XML (tef)

This format can...

- read routes

TEF, internally called "TourExchangeFormat", is an XML based export format used by Map&Guide Motorrad-Routenplaner 2005/06™.

Because this is only an export format, GPSTabel does not support writing to this format.

GPSTabel also supports the bcr format, which may also be used with this program and supports both reading and writing.

```
gpsbabel -r -i tef,routevia -f in.xml -o gpx -F out.gpx
```

## routevia option

Include only via stations in route.

This option may be used to eliminate calculated route points from the route.

## Map&Guide to Palm/OS exported files (.pdb) (mag\_pdb)

This format can...

- read waypoints
- read routes

With this format we support the Palm/OS export for Map&Guide based products like "PowerRoute", "Motorrad-Routenplaner" and (maybe) other software. The exported files can contain maps and/or route descriptions. The reader for this format has been tested with PowerRoute 5+6, Motorrad-Routenplaner 2002(-2006).

## Mapopolis.com Mapconverter CSV (mapconverter)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Mapconverter is a format that is read by Mapopolis.com's mapconverter application.

Mapconverter is an application used to create userland maps and map data for Mapopolis.com's Mapopolis program. The mapconverter format is essentially waypoint data prepared in a format that the mapconverter application will accept.

The steps for using GPSTabel and Mapconverter go something like this:

Step 1: Create a mapconverter file using gpsbabel.

```
gpsbabel -i geo -f geocaching.loc -o mapconverter -F foo.txt
```

Step 2: Launch mapconverter.exe and choose foo.txt as your input file. Click the begin button to have mapconverter process foo.txt.

If all goes successfully, you should have a file called "foo.pdb" ready for syncing with your PDA. Put it wherever Mapopolis thinks it should be on your PDA.

## Notes

- GPSBabel will write the name of its own output file in the output file it creates as the input for Mapconverter. Mapconverter will replace the extension of this filename with ".pdb".
- The PocketPC version of Mapopolis doesn't notice files with the ".pdb" extension. To make this work, change the extension to ".mlp" when copying the mapconverter output to your PocketPC PDA.
- Mapconverter only works with Mapopolis version 3.x. Mapopolis version 4 will refuse to load mapconverter maps. There is no known work-around for this at the time of this writing.
- Mapconverter is no longer available from the Mapopolis website. If you need a copy of mapconverter, ask on your local GPS Software discussion forum and I'm sure someone will have it. As far as I know, It was never actually acknowledged/supported by Mapopolis to begin with.

## MapTech Exchange Format (mxf)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

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.

## Microsoft AutoRoute 2002 (pin/route reader) (msroute)

This format can...

- read routes

Input support for Microsoft AutoRoute 2002-2006 .axe files and Microsoft Streets and Trips .est files. This is for reading routes created this program and is different than the s\_and\_t format used for writing pushpins.

These files contains only routes. We can extract the coordinates and the names of the points within route. An export to this format will not be supported.

Only the start, stops, and end points are stored in the .est/.axe/.ptm files. Turn-by-turn route data is not stored in the .est/.axe/.ptm files, and is recalculated by the Microsoft title each time on opening the file. This means that the output of GPSBabel will not contain turns needed for driving directions.

One possible approach to achieve similar results is to use the Streets and Trips drawing tools to trace a line over the interesting parts of the route to capture intersections or key turns. GPSBabel will capture stops in the route and insert those as turns, so adding stops at intersections can also improve the results when converting.

## Microsoft Streets and Trips (pin/route reader) (msroute)

This format can...

- read routes

Input support for Microsoft AutoRoute 2002-2006 .axe files and Microsoft Streets and Trips .est files. This is for reading routes created this program and is different than the `s_and_t` format used for writing pushpins.

These files contains only routes. We can extract the coordinates and the names of the points within route. An export to this format will not be supported.

Only the start, stops, and end points are stored in the .est/.axe/.ptm files. Turn-by-turn route data is not stored in the .est/.axe/.ptm files, and is recalculated by the Microsoft title each time on opening the file. This means that the output of GPSBabel will not contain turns needed for driving directions.

One possible approach to achieve similar results is to use the Streets and Trips drawing tools to trace a line over the interesting parts of the route to capture intersections or key turns. GPSBabel will capture stops in the route and insert those as turns, so adding stops at intersections can also improve the results when converting.

## Microsoft Streets and Trips 2002-2007 (s\_and\_t)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

This is a format for creating data to be read by Microsoft Streets and Trips [<http://www.microsoft.com/streets/default.mspix>]. It's been exercised on versions from 2003 through 2008. Detailed instructions on how to use it, including preserving hyperlinks, are at [gpsbabel.org](http://www.gpsbabel.org) [[http://www.gpsbabel.org/formats/s\\_and\\_t/Importing\\_into\\_Microsoft\\_Streets\\_and\\_Trips\\_2003.html](http://www.gpsbabel.org/formats/s_and_t/Importing_into_Microsoft_Streets_and_Trips_2003.html)]

This format has nothing to do with the .est/axe format used by this program to store routes.

## Motorrad Routenplaner (Map&Guide) .bcr files (bcr)

This format can...

- read and write routes

This file format (extension .bcr) is used in Map&Guide Motorrad Routenplaner 2002™ and later versions. BCR is a route-only format. If you own a newer release (2005 or later) you may also use the XML export with GPSBabel's tef input format.

There may be other products from Map&Guide that use this format as well.

Coordinates are stored in a BCR file in a Mercator projection. The conversion from the Mercator projection to polar (latitude/longitude) coordinates and back again may result in visible differences. Experience reports are welcome.

### Example 3.9. Sample BCR command with all options

```
gpsbabel -r -i gpx -f in.gpx -o bcr,index=1,name="From A to B",radius=6371012 -F a_to_b.bcr
```

## index option

Index of route to write (if more the one in source).

Because the format supports only one route, this option may be used on output to select a single route from a collection of routes read from a more expressive format. If you have, say, a gpx file that contains two routes, you may use this option to write them one at a time to individual files.

```
gpsbabel -i gpx -f routes.gpx -o bcr,index=1 -F route1.bcr -o bcr,index=2 -F route2.bcr
```

## name option

New name for the route.

This route specifies the name of the route. This is particularly useful if the route came from an input format that did not support named routes, but it may also be used to rename a route.

## radius option

Radius of our big earth (default 6371000 meters).

This option instructs GPSBabel to use a different value for the radius of the earth when converting between the Mercator projection and geographic coordinates. The default value is 6371000.0 meters.

Careful experimentation with this value may help to reduce conversion errors.

## prefer\_shortnames option

Use shortname instead of description.

This option causes GPSBabel to use the short name of the waypoint instead of the description.

# MS PocketStreets 2002 Pushpin (psp)

This format can...

- read and write waypoints

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.

## Frequently Asked Questions

1. Why should I use GPSBabel/psp to make pushpins when Streets & Trips (S&T) already does that for me?

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).

2. I keep getting a blank (32 byte) PSP file.

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.

3. I've created a PSP file, now what do I do with it?

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".

4. I don't have a map. What do I do now?

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.

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

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.

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

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

7. Does GPSTBabel/psp work with (Autoroute, Mappoint, etc..) .PSP files?

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.

8. Does GPSTBabel/psp work with (S&T 2001, S&T 2002, etc...) files?

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

9. Does GPSTBabel/psp work with (insert your country/location here) maps?

If it doesn't, feel free to inquire on the GPSTBabel-Misc [<https://lists.sourceforge.net/lists/listinfo/gpsbabel-misc>] mailing list.

10. What do you mean S&T writes points with the wrong coordinates?

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.

11. 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?

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 GPSTBabel in the first place.

12. Where can I find documentation for the layout of PSP files?

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).

13. I have some other problem, what do I do?

Ask your question on the GPSTBabel-Misc [<https://lists.sourceforge.net/lists/listinfo/gpsbabel-misc>] mailing list.

## MTK Logger (iBlue 747,...) Binary File Format (mtk-bin)

This format can...

- read tracks

Binary file protocol converter for MTK based GPS loggers. This format reads the raw binary format created by the MTK Windows application and outputs to other formats supported by GPSTBabel When using the csv option a MTK application compatible output file will also be created.

It has been tested with Transystem i-Blue 747™ but other devices should work as well (Qstarz BT-Q1000, iTrek Z1, ...)

All position items (including button push) will be listed as trackpoints in the output. Log items due to button push are presented as waypoints. In theory we would not add waypoints to the list of trackpoints.

But as the MTK logger restart the log session from the button press we would loose a trackpoint unless we include/duplicate it.

Transystem i-Blue 747 [<http://www.transystem.com.tw/p-gps-ibblue747.htm>]

### Example 3.10. Convert MTK binary trackpoints to GPX

```
gpsbabel -t -i mtk-bin, csv=extra.csv -f data.bin -o gpx -F out.gpx
```

Additionally a CSV output file is created.

## csv option

MTK compatible CSV output file.

Specifies a filename into which MTK-compatible CSV output will be written.

Note that this option is a bit of an oddity in the GPSTools arsenal. This should probably be a "real" output type of its own instead of being bolted onto an input type.

## MTK Logger (iBlue 747, Qstarz BT-1000,...) download (mtk)

This format can...

- read tracks

Serial download protocol for the i-Blue 747™ and other MTK based GPS data loggers. Observe that it is only possible to download data using USB cable, Bluetooth requires a hardware modification.

Transystem i-Blue 747 [<http://www.transystem.com.tw/p-gps-ibblue747.htm>] Downloaded data will be stored in data.bin file in the current directory together with the chosen output format.

It has been tested with Transystem i-Blue 747 but other devices should work as well (Qstarz BT-Q1000, iTrek Z1, ...)

See mtk-bin on how trackpoints/waypoints are handled

### Example 3.11. Command showing MTK download track and waypoints and erase on Linux

```
gpsbabel -t -w -i mtk,erase -f /dev/ttyUSB0 -o gpx -F out.gpx
```

For more info and tweaks on MTK based loggers: MTK Tips and Tweaks [[http://www.gpspassion.com/forumsen/topic.asp?TOPIC\\_ID=81990](http://www.gpspassion.com/forumsen/topic.asp?TOPIC_ID=81990)] iBlue 747 Logger [[http://www.gpspassion.com/forumsen/topic.asp?TOPIC\\_ID=81315](http://www.gpspassion.com/forumsen/topic.asp?TOPIC_ID=81315)] For info about the used log format: MTK binary format [<http://spreadsheets.google.com/pub?key=pyCLH-0TdNe-5N-5tBokuOA&gid=5>]

## erase option

Erase device data after download.



This option erases the track log from the device after download.

## csv option

MTK compatible CSV output file.

This option will create an additional CSV output file. The CSV file is compatible with the original MTK logger application.

## National Geographic Topo .tpg (waypoints) (tpg)

This format can...

- read and write waypoints

National Geographic Topo! Waypoint and Route Format. This module reads and writes .TPG files created by various editions of NG Topo! Reading/writing of route data is not supported yet.

Contributed by Alex Mottram.

## datum option

Datum (default=NAD27).

The option 'datum="datum name"' can be used to override the default of NAD27 ("N. America 1927 mean") which is correct for the continental U.S.

Any legal datum supported by GPSTabel may be used. For example, points in Hawaii should use "Old Hawaiian\_mean".

## National Geographic Topo 2.x .tpo (tpo2)

This format can...

- read tracks

This module reads tracks from .TPO files created by National Geographic Topo! version 2.x

Contributed by Steve Chamberlin.

## National Geographic Topo 3.x/4.x .tpo (tpo3)

This format can...

- read waypoints
- read tracks
- read routes

This module reads .TPO files created by National Geographic Topo! version 3.x and 4.x. It will read tracks, routes, waypoints, map notes, symbols, and text notes. The latter three are converted to waypoints.

Contributed by Curt Mills.

## Navicache.com XML (navicache)

This format can...

- read waypoints

This is the XML format that's used by Navicache.com for their geocaching data. There are a number of fields in it that are marked "required" but are Navicache-specific, so GPSTools can not write these files, but we can still read them. navicache.com [<http://www.navicache.com/cgi-bin/ib312a/ikonboard.cgi?act=ST;f=23;t=334>]

## noretired option

Suppress retired geocaches.

If this option is present, retired (archived) caches will be suppressed on write.

## Navigon Mobile Navigator .rte files (nmn4)

This format can...

- read and write routes

Support for Navigon Mobile Navigator route (.rte) files. This is a very simple text format that only requires coordinates, but has fields for many other things. We only write coordinates as fields like 'city' and 'street' cannot typically be populated from other formats. [www.navigon.com](http://www.navigon.com) [<http://www.navigon.com>]

## index option

Index of route to write (if more than one in source).

Because the format supports only one route, this option may be used on output to select a single route from a collection of routes read from a more expressive format. If you have, say, a gpx file that contains two routes, you may use this option to write them one at a time to individual files.

```
gpsbabel -i gpx -f routes.gpx -o nmn4,index=1 -F route1.rte -o  
nmn4,index=2 -F route2.rte
```

## NaviGPS GT-11/BGT-11 Download (navilink)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

GPSTBabel supports the Navilink protocol used by the Locosys GT-11 [<http://www.locosystech.com/product.php?zln=en&id=5>] GPS receivers. These are sold under a variety of names including:

NaviGPS	GT-11	Amaryllo
NaviGPS-BT	BGT-11	

This format is used for both the serial protocol used on the USB link and for the files which can be copied from the internal memory to the SD card using recent firmware versions.

If you specify a serial port for the file (.e.g. "COM1", "/dev/ttyUSB0") to be read or written, GPSTBabel will use the serial protocol. Specifying a file, either on local filesystem or on a mounted flash card reader, will result in the file-based format being used.

To access the device using the serial protocol over USB the device needs to be in Navilink mode, which can be activated from the main menu of the device.

Details of the Navilink serial protocol can be found here [<http://wiki.splitbrain.org/navilink>].

## **nuketrk option**

Delete all track points.

This option erases all track data in the receiver before doing a transfer.

This is a convenient option to use in automated processes where you want to ensure the units starts with a clean state before sending track data to it. Using this option on transmit is a better idea than doing it on receive since the latter would erase all the waypoints before asking the unit to send all the waypoints.

## **nukerte option**

Delete all routes.

This option erases all routes in the receiver before doing a transfer.

This is a convenient option to use in automated processes where you want to ensure the units starts with a clean state before sending routes to it. Using this option on transmit is a better idea than doing it on receive since the latter would erase all the waypoints before asking the unit to send all the waypoints.

## **nukewpt option**

Delete all waypoints.

This option erases all waypoints in the receiver before doing a transfer.

This is a convenient option to use in automated processes where you want to ensure the units starts with a clean state before sending waypoints to it. Using this option on transmit is a better idea than doing it on receive since the latter would erase all the waypoints before asking the unit to send all the waypoints.

## **power\_off option**

Command unit to power itself down.

This options powers down the Navilink receiver once any transfers are complete.

## Navitrak DNA marker format (dna)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

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.

## NetStumbler Summary File (text) (netstumbler)

This format can...

- read waypoints

This format reads summary files from NetStumbler™ 0.4 or MacStumbler™.

The default behavior when creating waypoints is to use the SSID for the short name, and information about the access point for the description. When the SSID is not unique, is not available, or consists of whitespace, a short name is synthesized.

Different icons are assigned to encrypted, non-encrypted, stealth, and non-stealth access points; these may be changed with options.

NetStumbler [<http://www.netstumbler.com/>]

MacStumbler [<http://www.macstumbler.com/>]

### nseicon option

Non-stealth encrypted icon name.

This option specifies the name of the icon to use for non-stealth, encrypted access points.

### nsneicon option

Non-stealth non-encrypted icon name.

This option specifies the name of the icon to use for non-stealth, non-encrypted access points.

### seicon option

Stealth encrypted icon name.

This option specifies the name of the icon to use for stealth, encrypted access points.

### sneicon option

Stealth non-encrypted icon name.

This option specifies the name of the icon to use for stealth, non-encrypted access points.

## snmac option

Shortname is MAC address.

This option causes GPSBabel to use the MAC address as the short name for the waypoint. The unmodified SSID is included in the waypoint description.

## NIMA/GNIS Geographic Names File (nima)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

This is a CSV format from the National Imagery and Mapping Agency.

## NMEA 0183 sentences (nmea)

This format can...

- read and write waypoints
- read and write tracks

This format is the file representation of the NMEA (National Marine Electronics Association) 0183 log and waypoint format for GPS devices. Some hardware and software that work with NMEA-0183 formatted data include:

GPS Data Logger	[ <a href="http://VisualGPS">http://VisualGPS</a>	[ <a href="http://SparkFun GPS Datalogger">http://SparkFun GPS Datalogger</a>
<a href="http://homepages.tig.com.au/~robk/datalogger.html">homepages.tig.com.au/~robk/datalogger.html</a> ]	<a href="http://www.visualgps.net/VisualGPSce/default.htm">www.visualgps.net/VisualGPSce/default.htm</a> ]	<a href="http://www.sparkfun.com/">www.sparkfun.com/</a> ]
GPS TrackMaker	[ <a href="http://GPS">http://GPS</a>	[ <a href="http://Sony">http://Sony</a>
<a href="http://www.gpstm.com/eng/features_eng.htm">www.gpstm.com/eng/features_eng.htm</a> ]	Utility <a href="http://www.gpsu.co.uk/">www.gpsu.co.uk/</a> ]	GPS-CS1 <a href="http://www.sonystyle.com/is-bin/INTERSHOP.enfinity/eCS/Store/en/-/USD/SY_DisplayProductInformation-Start?ProductSKU=GPSCS1">www.sonystyle.com/is-bin/INTERSHOP.enfinity/eCS/Store/en/-/USD/SY_DisplayProductInformation-Start?ProductSKU=GPSCS1</a> ]
GPSPMaster	[ <a href="http://GeoConv">http://GeoConv</a>	[ <a href="http://">http://</a>
<a href="http://www.gpsmaster.nl/">www.gpsmaster.nl/</a> ]	<a href="http://www.kolumbus.fi/eino.uikkanen/geoconvgb/index.htm">www.kolumbus.fi/eino.uikkanen/geoconvgb/index.htm</a> ]	
NMEAlog	[ <a href="http://CommLinx">http://CommLinx</a>	[ <a href="http://">http://</a>
<a href="http://www.silcom.com/~rwhately/index.html">www.silcom.com/~rwhately/index.html</a> ]	GPS recorder <a href="http://www.commlinx.com.au/GPS_recorder.htm">www.commlinx.com.au/GPS_recorder.htm</a> ]	

This module also supports realtime tracking which allows realtime position reports from a GPS, such as one connected serially, over Bluetooth, or a USB module emulating a serial port, to be used with selected output formats.

When used in realtime tracking mode, if GPSBabel does not sense incoming NMEA sentences arriving from the port, it will send Sirf "reset to NMEA" commands to the port at a variety of speeds in an attempt

to communicate with an attached GPS. This lets devices like the Microsoft GPS or Pharos GPS that are Sirf chips with an integrated USB/Serial adapter work with this input format.

## **snlen option**

Max length of waypoint name to write.

This option specifies the maximum length to be used for waypoint names in the GPWPL sentence. Longer names will be shortened to no more than this length, but all waypoint names will remain unique.

## **gprmc option**

Read/write GPRMC sentences.

This option tells GPSBabel whether to read (on input) or write (on output) GPRMC sentences. The default is to read or write GPRMC sentences. To disable GPRMC sentences, specify `gprmc=0`.

GPRMC sentences contain the "recommended minimum" positional information, including date and time, heading, and velocity. Note that they do not include altitude. For altitude, you will have to include GPGGA sentences.

## **gpgga option**

Read/write GPGGA sentences.

This option tells GPSBabel whether to read (on input) or write (on output) GPGGA sentences. The default is to read or write GPGGA sentences. To disable GPGGA sentences, specify `gpgga=0`.

GPGGA sentences contain the location and quality of the GPS position fix.

## **gpvtg option**

Read/write GPVTG sentences.

This option tells GPSBabel whether to read (on input) or write (on output) GPVTG sentences. The default is to read or write GPVTG sentences. To disable GPVTG sentences, specify `gpvtg=0`.

GPVTG sentences contain information about the heading and the speed at the time of the fix. They do not contain any location information; for that you will need either or both of GPGGA or GPRMC.

## **gpgsa option**

Read/write GPGSA sentences.

This option tells GPSBabel whether to read (on input) or write (on output) GPGSA sentences. The default is to read or write GPGSA sentences. To disable GPGSA sentences, specify `gpgsa=0`.

GPGSA sentences contain information on the quality of the positional fix and the individual satellites from which it was derived. However, GPSBabel neither reads nor writes the individual satellite data. On input, the satellite fields are ignored and on output they are left blank.

## **date option**

Complete date-free tracks with given date (YYYYMMDD)..

On input, track points with times but no dates will have this date applied.

This is necessary because some NMEA sentences contain times but no dates. If this option is not specified and the date cannot be determined from one or more of the available NMEA sentences, the tracks will be discarded.

## **get\_posn option**

Return current position as a waypoint.

This options, when specified, returns the current position as a single waypoint.

## **pause option**

Decimal seconds to pause between groups of strings.

This option tells GPSBabel to pause between individual track records when used on output. This may be used with appropriate external software or hardware to simulate a GPS receiver for testing purposes. On Unix, for example, you may use a named pipe to feed the output from GPSBabel to gpsd.

If a value for this option is specified, it is in seconds and it may be either a whole number of seconds or a fraction (e.g. 0.5 for a 1/2 second pause between trackpoints.)

If this option is specified without a value, the time between adjacent trackpoints will be computed and used for the length of the pause. That is, if your trackpoints are 5 seconds apart, GPSBabel will pause 5 seconds between trackpoints.

Note that very long tracks may be subject to clock drift, as GPSBabel does not take into account the amount of time it may take to write the NMEA sentences. Also, there is no guarantee that it will pause for exactly the specified number of seconds between samples; different operating systems will allow greater or lesser precision for timers, so actual precision may be as much as plus or minus 100 milliseconds.

If you are using this option with compressed or simplified tracks from your handheld GPS receiver, you might find the interpolate filter useful.

## **append\_positioning option**

Append realtime positioning data to the output file instead of truncating.

When writing NMEA realtime positioning data, append to the output file instead of truncating it on each successive position fix.

## **baud option**

Speed in bits per second of serial port (baud=4800).

To the "nmea" module, the "baud" option specifies the baud rate of the serial connection when used with the real-time tracking option.

## **gisteq option**

Write tracks for Gisteq Phototracker.

This option writes the Gisteq format - which has the extension of .GPS - to allow third-party GPS hardware with the Gisteq PhotoTrackr software.

The Gisteq PhotoTrackr is a GPS data logger hardware and software package that allows one to easily record the locations of where the user has taken photos. The PhotoTrackr software works by comparing EXIF timestamps in digital photos with the timestamps in the tracking data. In doing so, the software plots the locations of the photos using Google Maps. The logging format used by the Gisteq hardware is very close to NMEA format, but with a few small quirks.

More information can be found at the Gisteq [<http://www.gisteq.com/>] site.

## Nokia Landmark Exchange (Imx)

This format can...

- read and write waypoints

This format supports Nokia Landmark Exchange (LMX) files used by several Nokia phones. GPSBabel supports only the traditional XML format and not the compressed binary format.

## OpenStreetMap data files (osm)

This format can...

- read and write waypoints
- write tracks
- read and write routes

This format is used to exchange data with the OpenStreetMap [<http://www.openstreetmap.org>] project. The main goal of this collaborative project is to create free editable maps.

These data files are XML based. Every GPS element (way or node) described by the files has a unique number as identifier. When we write OSM data files and don't know something about the id's, negative numbers will be used as identifier. This has been tested with JOSM [<http://wiki.openstreetmap.org/index.php/JOSM>].

Because the resulting timestamps of OSM ways differ from real GPS tracks, we read OSM ways into routes. On the output side we write all available routes and tracks into the osm target file.

### tag option

Write additional way tag key/value pairs.

With this option you can preset OSM features [[http://wiki.openstreetmap.org/index.php/Map\\_Features](http://wiki.openstreetmap.org/index.php/Map_Features)] (tags) on all exported ways.

```
gpsbabel -i gdb -f ways.gdb -o osm,tag="highway:motorway" -F ways.osm
```

### tagnd option

Write additional node tag key/value pairs.



With this option you can preset OSM features [[http://wiki.openstreetmap.org/index.php/Map\\_Features](http://wiki.openstreetmap.org/index.php/Map_Features)] (tags) on every written nodes.

```
gpsbabel -i gdb -f nodes.gdb -o osm,tagnd="amenity:pub;building:yes" -F nodes.osm
```

## OziExplorer (ozi)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

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

### pack option

Write all tracks into one file.

In normal case GPSTools creates for each track a separate file (track.plt, track-1.plt, ...). With this option all tracks will be written into one file. A '1' in the third field of the trackpoint record signals the beginning of a new track.

```
gpsbabel -i gpx -f tracks.gpx -o ozi,pack -F track
```

### snlen option

Max synthesized shortname length.

This option allows you to specify the length of waypoint names written to this format when used with the `-s` option.

### snwhite option

Allow whitespace synth. shortnames.

This option forces waypoint names generated with `-s` to allow whitespace in the names.

### snupper option

UPPERCASE synth. shortnames.

When specified, this option will force generated shortnames to be in all uppercase letters.

### snunique option

Make synth. shortnames unique.

When specified, this option will force the generated waypoint names to be unique.

## wptfgcolor option

Waypoint foreground color.

This option allows you to specify a foreground color of a waypoint. You can specify it as either a decimal number or one of the standard web colors.

## wptbgcolor option

Waypoint background color.

This option allows you to specify a background color of a waypoint. You can specify it as either a decimal number or one of the standard web colors.

## proximity option

Proximity distance.

This option, specified in meters, allows you to set the proximity of written in waypoints.

# PalmDoc Output (palmdoc)

This format can...

- write waypoints

PalmDoc output is similar to Text output, except that it generates a Palm Database (PDB) file suitable for use with programs like CSpotRun, TealDoc, AportisDoc, Palm Reader, and others. The resulting file also contains bookmarks to make it easy to jump to a particular waypoint.

The following command line reads a GPX file with Groundspeak extensions and writes a Palm document with encrypted hints and logs:

```
gpsbabel -i gpx -f 12345.gpx -o "palmdoc,dbname=Unfound  
Geocaches,encrypt,logs" -F 12345.pdb
```

## nosep option

No separator lines between waypoints.

To suppress the dashed lines between waypoints, use this option.

## dbname option

Database name.

This option specifies the internal name for the document. This is the name that appears in your document reader, not the name of the file that is created on your computer.

## **encrypt option**

Encrypt hints with ROT13.

Use this option to encrypt hints from Groundspeak GPX files.

## **logs option**

Include groundspeak logs if present.

Use this option to include Groundspeak cache logs in the created document.

## **bookmarks\_short option**

Include short name in bookmarks.

If you would like the generated bookmarks to start with the short name for the waypoint, specify this option.

This is particularly useful when used in combination with the 'sort' filter.

# **PathAway Database for Palm/OS (pathaway)**

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

PathAway is a Palm software designed for handling "most" GPS devices (including BlueTooth). In this time (I mean 2005) a free tool to convert this database is located on the homepage of PathAway ([www.pathaway.com](http://www.pathaway.com)). But I've read there ... for windows and the output formats are also very limited.

## **date option**

Read/Write date format (i.e. DDMMYYYY).

This option specifies the input and output format for the date. The format is written similarly to those in Windows. An example format is "YYMMDD".

## **dbname option**

Database name.

This option specifies the database name for the output file. This name is not the same thing as the file name on your computer; this is the name that appears in the file browser on your handheld.

## **deficon option**

Default icon name.

The `deficon` option is used to control the icon output when writing to this format. It overrides any icon informatino that might be present in the source data.

## **snlen option**

Length of generated shortnames.

This option allows you to specify the length of waypoint names written to this format when used with the `-s` option.

## **Quovadis (quovadis)**

This format can...

- read and write waypoints

QuoVadis for Palm OS [marcosoft.com](http://www.marcosoft.com/) [<http://www.marcosoft.com/>] is a program for Palm/OS. Working with record definitions provided by MarcoSoft and further experimentation by Bruce Thompson and "Fuzzy" from the Geocaching Forums to nail down the format precisely.

Should work fine for import and export.

One thing of note, QuoVadis stores all waypoints in a single Palm Database without using categories. This means that it may be difficult to keep personal waypoints separate from generated waypoints. What Bruce recommends is taking the QuoVadisMarkerDB.PDB file synced down from your Palm Powered device and extract the waypoints you personally set to a GPX file. Then using GPSBabel's joining capabilities generate a new PDB file from the personal file and the other waypoint files of interest.

Currently the selection of icons to display and the scale at which to display them is hardcoded. Also there is no support for notes associated with waypoints. This will be addressed in a future revision.

## **dbname option**

Database name.

This option specifies the database name for the output file. This name is not the same thing as the file name on your computer; this is the name that appears in the file browser on your handheld.

## **Raymarine Waypoint File (.rwf) (raymarine)**

This format can...

- read and write waypoints
- read and write routes

This format supports the "Raymarine Waypoint File" format (.rwf). More information to Raymarine you'll find at their homepage [<http://www.raymarine.com/>].

Known limits: max. 16 characters for waypoint names and max. 50 waypoints per route.

## **location option**

Default location.

With this option you can specify the name of the folder where the waypoints are placed.

This name is also limited to 16 characters.

## See You flight analysis data (cup)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

This format supports flight analysis data from the See You [<http://www.seeyou.ws/>] program.

Position information is preserved, but the aviation-specific information such as runway length and airport frequency, are written as blanks and ignored on read.

Tasks are not supported.

## Sportsim track files (part of zipped .ssz files) (sportsim)

This format can...

- read and write tracks

This format is derived from the xcsv format, so it has all of the same options as that format.

With this format we support Sportsim trackfiles located in zipped .ssz archives.

Currently we cannot read zipped files directly with GPSTabel. So you have to extract the archive before you can use any file. The trackfiles have .txt extensions.

From the Sportsim homepage: [<http://www.sportsim.com/web/index.php?url=about>]

Sportsim provide software applications and web-based graphically simulated performance information and image solutions to outdoor active people.

## Suunto Trek Manager (STM) .sdf files (stmsdf)

This format can...

- read and write tracks
- read and write routes

This format supports the .sdf files from the Suunto product family 'Suunto Trek Manager', 'Suunto Ski Manager' and 'Suunto Sail Manager'. The contents of the sdf file depends on the used product and can be one route or one track. That's why when you want to use sdf on the output side you have to use the -r OR the -t option. This will tell GPSTabel which type of data should be written.

Currently we can read the following file types:

4 = M9 TrackLog  
 5 = Route  
 28 = X9 TrackLog

```
gpsbabel -i gpx -f some-routes.gpx -r -o stmsdf,index=3 -F single-  
route.sdf
```

Suunto Website [<http://www.suunto.fi>]

## index option

Index of route (if more the one in source).

Convert route number 'index' from source into sdf format.

We have a lot of more expressive formats thats support more than one route. At this place sdf files are limited to only one single route. With option index you can specify which route from source should be converted.

Our default index is 1.

This example will convert route number two and three into separate sdf files:

```
gpsbabel -i gdb -f routes.gdb -r -o stmsdf,index=2 -F route-one.sdf -  
r -o stmsdf,index=3 -F route-three.sdf
```

## Suunto Trek Manager (STM) WaypointPlus files (stmwpp)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

This format supports the Suunto Trek Manager (STM) WaypointPlus format. This is a simple format with coordinates and a time stamp. Route points also have a short name. A single file may only contain one route or one track.

Suunto Website [<http://www.suunto.fi>]

## index option

Index of route/track to write (if more the one in source).

Because the format supports only one route or track, this option may be used on output to select a single route or track from a collection of routes and tracks read from a more expressive format. If you have, say, a gpx file that contains three routes, you may use this option to write them one at a time to individual files.

```
gpsbabel -i gpx -f routes.gpx -o stmwpp,index=1 -F route1.txt -o  
stmwpp,index=2 -F route2.txt -o stmwpp,index=3 -F route3.txt
```

## Swiss Map # (.xol) format (xol)

This format can...

- read and write waypoints
- read and write tracks

This module reads and writes xml based (.xol) files used by Swiss Map [<http://www.swisstopo.ch>] software.

These files uses the "Swiss national grid" (CS-1903) to store coordinates.

## Tab delimited fields useful for OpenOffice, Ploticus etc. (openoffice)

This format can...

- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

Tab seperated export-all (except geocaching data) file format. Intended to serve as source for number-processing applications like OpenOffice, Ploticus and others. Tab was chosen as delimiter because it is a) supported by both OpenOffice and Ploticus and b) is not ',', so you can use

```
sed -i "s/./,/g" <x>.csv'
```

to adapt it to locales where ',' is used as decimal seperator. Contributed by Tobias Minich.

## Textual Output (text)

This format can...

- write waypoints

This is a simple human readable version of the data file, handy for listings of any type of waypoint files.

The following command line reads a GPX file with Groundspeak extensions and writes a text file with encrypted hints:

```
gpsbabel -i gpx -f 12345.gpx -o text,encrypt -F 12345.txt
```

## nosep option

Suppress separator lines between waypoints.

To suppress the dashed lines between waypoints, use this option.

## encrypt option

Encrypt hints using ROT13.

Use this option to encrypt hints from Groundspeak GPX files.

## logs option

Include groundspeak logs if present.

Use this option to include Groundspeak cache logs in the created document.

## degformat option

Degrees output as 'ddd', 'dmm'(default) or 'dms'.

When GPSTabel writes coordinates, this option is consulted to see if it should write decimal degrees ('ddd') decimal minutes ('dmm') or degrees, minutes, seconds ('dms'). The default is 'dmm'.

## altunits option

Units for altitude (f)eeet or (m)etres.

This option should be 'f' if you want the altitude expressed in feet and 'm' for meters. The default is 'f'.

## splitoutput option

Write each waypoint in a separate file.

Splits output into separate files for each waypoint by appending a decimal number to the output filename.

### Example 3.12. Example for splitoutput option to text format

If "MyPQ.gpx" contains five waypoints,

```
gpsbabel -i gpx -f MyPocketQuery -o text,split -F blah
```

will result in files named blah1 ... blah5, each containing info from one of those waypoints.

## TomTom Itineraries (.itn) (tomtom\_itn)

This format can...

- read and write routes

This format is derived from the xcsv format, so it has all of the same options as that format.

tomtom\_itn can be used to read and write TomTom [<http://www.tomtom.com>] Navigator Itineraries (Routes).

## TomTom POI file (.asc) (tomtom\_asc)

This format can...



- read and write waypoints

This format is derived from the xcsv format, so it has all of the same options as that format.

With this format you can read and write TomTom [<http://www.tomtom.com>] Points of Interest - POI (ascii) files. It is a simple text (csv) format with only latitude, longitude and a short name.

## TomTom POI file (.ov2) (tomtom)

This format can...

- read and write waypoints

This format can read and write TomTom [<http://www.tomtom.com>] .ov2 (POI) files, as used by the TomTom GO and TomTom Navigator. It has been tested with an original TomTom GO running version 5.00 of the TomTom software. There may be some records that confuse the input module - if you have an example of such a record "in the wild", and you aren't restricted from sharing it, we encourage you to post to the gpsbabel-misc mailing list to contact a developer.

Note that in addition to the .ov2 file, you will need a .bmp file for the icon. It should be 22x22 and 16 colors, and have the same name (not including the extension) as the .ov2 file.

## TopoMapPro Places File (tmpro)

This format can...

- read and write waypoints

Reads and writes places files for use in TopoMapPro places files [<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.

## TrackLogs digital mapping (.trl) (dmtlog)

This format can...

- read and write waypoints
- read and write tracks

This format can be used to convert files from TrackLogs Digital Mapping [<http://www.tracklogs.co.uk>]. The files have extension .trl and can contain waypoints and tracks.

We have seen three different types of this format. Two are binary and one is an XML based format. All three types are supported by our reader.

## index option

Index of track (if more the one in source).

Convert track number 'index' from source into dmtlog format.

The known variants of Tracklog 'digital mapping' files supports only one track per file. If you have more than one track in source (f.e MapSource and many others can do such heavy things), you can specify which track should be used for the conversion.

The default index is 1 (the first track of a possible list of tracks).

An example usage you can find at the `ignrando` format, which uses option `index` in same manner.

## U.S. Census Bureau Tiger Mapping Service (tiger)

This format can...

- read and write waypoints

The U.S. Census Bureau provides online mapping facilities. This format is described at: [tiger.census.gov](http://tiger.census.gov) [<http://tiger.census.gov/instruct.html>]. Do notice that this format is not the actual Tiger line mapping records, but rather the interface to their online mapping program.

### **nolabels option**

Suppress labels on generated pins.

This option tells GPSBabel to not generate labels on the pins. If this is true, the description of the incoming waypoints are ignored and not placed on the pins.

### **genurl option**

Generate file with lat/lon for centering map.

`genurl` is a convenience option for generating the scaling parameters when accessing the Tiger servers. It will output the latitude, longitude, height, and width parameters in a form suitable for use in the URL to generate a map that will hold all the points to be displayed and is suitably scaled and centered.

For example:

```
gpsbabel -i geo -f geocaching.loc -o tiger,genurl=tiger.ctr -F tiger.dat
```

may create `tiger.ctr` with

```
lat=36.042108&lon=-86.877408&ht=0.161172&wid=0.591771&iwd=768&iht=768
```

After uploading `tiger.dat` to a public server, a request to

```
http://tiger.census.gov/cgi-bin/mapgen?url=$THATFILE$(cat tiger.ctr)
```

will return a gif file from the tiger server that's suitably scaled.

### **margin option**

Margin for map. Degrees or percentage.

This option specifies a margin around the maps for the `genurl` options. The margin may be specified in either decimal degrees or as a percentage.

This option is most useful for ensuring there is adequate space for the label around the markers when generating automatically scaled maps.

## **snlen option**

Max shortname length when used with `-s`.

The `snlen` option controls the maximum length of names generated by the `'-s'` option. It's particularly useful in Tiger maps to avoid the amount of clutter generated by potentially lengthy labels on the markers.

## **oldthresh option**

Days after which points are considered old.

This options allows you to control the threshold in days between whether a pin is considered "new" (and thus potentially governed by the `'newmarker'` option) or "old" (and thus potentially governed by the `'oldmarker'` option).

## **oldmarker option**

Marker type for old points.

This option specifies the pin to be used if a waypoint has a creation time newer than `'oldthresh'` days.

The default is `"redpin"`.

## **newmarker option**

Marker type for new points.

This option specifies the pin to be used if a waypoint has a creation time older than `'oldthresh'` days.

The default is `"greenpin"`.

## **suppresswhite option**

Suppress whitespace in generated shortnames.

When set, this options tells the `'-s'` smartname generator to not allow any spaces in the labels generated for markers.

## **unfoundmarker option**

Marker type for unfound points.

## **xpixels option**

Width in pixels of map.

The `xpixels` argument lets you specify the number of pixels to be generated by the Tiger server along the horizontal axis when using the `'genurl'` option.

## **ypixels option**

Height in pixels of map.

The `ypixels` argument lets you specify the number of pixels to be generated by the Tiger server along the vertical axis when using the `'genurl'` option.

## **iconismarker option**

The icon description is already the marker.

This options signifies that the icon in the incoming format is to be used without change in the generated Tiger output file. Without this option, GPSBabel tries to color pins based on their creation time and certain Geocaching traits when available.

# **Universal csv with field structure in first line (unicsv)**

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

Unicsv examines the first line of a file to determine the field order and field separator in that file. On write, it tries to figure out what data it has and writes headers and all the data it can.

If the first line contains any tabs, the data lines are assumed to be tab separated. Otherwise the fields are assumed to be separated by commas.

The list of keywords include:

```
alt =      Elevation (in meters) of the point
bng_e =    British National Grid's easting
bng =      full coordinate in BNG format (zone easting northing)
bng_pos =  full coordinate in BNG format (zone easting northing)
bng_n =    British National Grid's northing
bng_z =    British National Grid's zone
caden =    Cadence
comment =  Notes
cour =     Heading / Course true
date =     Date (yyyy/mm/dd)
depth =    Depth
desc =     Description
ele =      Elevation (in meters) of the point
e/w =      'e' for eastern hemisphere, 'w' for western
fix =      3d, 2d, etc.
geschw =   Geschwindigkeit (speed)
```

hdop =	Horizontal dilution of precision
head =	Heading / Course true
heart =	Heartrate
height =	Elevation (in meters) of the point
icon =	Symbol (icon) name
lat =	Latitude
lon =	Longitude
name =	Waypoint name ("Shortname")
n/s =	'n' for northern hemisphere, 's' for southern
notes =	Notes
pdop =	Position dilution of precision
prox =	Proximity
sat =	Number of sats used for fix
speed =	Speed
symb =	Symbol (icon) name
tempf =	Temperature (degrees Fahrenheit)
temp =	Temperature (degrees Celsius)
time =	Time (hh:mm:ss[.msec])
url =	URL
utc_d =	UTC date
utc_t =	UTC time
utm_c =	UTM zone character
utm_e =	UTM easting
utm =	full coordinate in UTM format (zone zone-ch easting northing)
utm_pos =	full coordinate in UTM format (zone zone-ch easting northing)
utm_n =	UTM northing
utm_z =	UTM zone
vdop =	Vertical dilution of precision
x =	Longitude
x_pos =	Longitude
y =	Latitude
y_pos =	Latitude
z =	Altitude (elevation)

We support some enhanced Garmin attributes. They are also available in gpx, gdb, garmin\_gpi and partly garmin\_txt. These entities are currently not visible in MapSource™ (6.12.4), but are NOT dropped when working with GDB (version 3) or GPX files.

Please note, that these do NOT provide a geocoding service; don't expect to "convert" a street address to a latitude and longitude.

addr =	Street address
city =	City
country =	Country
faci =	Facility (not available in GPX)
phone =	Phone number
post =	Postal code
state =	State

Fuller spellings (i.e. "longitude") may be used. You can also use keywords with a whitespace instead of an underscore.

A typical file may be:

```
Name, Latitude, Longitude, Description
GCEBB, 35.972033, -87.134700, Mountain Bike Heaven by susy1313
GC1A37, 36.090683, -86.679550, The Troll by a182pilot & Family
```

On the output side unicsv writes fixed number of columns (waypoint index, latitude and longitude) followed by a variable column list depending on internal data.

With at least ONE valid timestamp in data a unicsv output may look like that:

```
No, Name, Latitude, Longitude, Description, Date, Time
1, "GCEBB", 35.972033, -87.134700, "Mountain Bike Heaven by susy1313", 2003/06/29,
2, "GC1A37", 36.090683, -86.679550, "The Troll by a182pilot & Family", ,
```

## datum option

GPS datum (def. WGS 84).

This option specifies the datum to be used on output. Valid values for this option are listed in Appendix A, *Supported Datums*.

## grid option

Write position using this grid..

This value specifies the grid to be used on write. It is similar to the grid option of garmin\_txt (see Table 3.1, “Grid values for garmin\_txt”). The only difference is that unicsv does not write a degree sign (°) into the output file.

Without this option unicsv writes the coordinates as simple numbers like in the samples above.

## utc option

Write timestamps with offset x to UTC time.

This option specifies the local time zone to use when writing times. It is specified as an offset from Universal Coordinated Time (UTC) in hours. Valid values are from -23 to +23.

## Vcard Output (for iPod) (vcard)

This format can...

- write waypoints

The vCard output is intended to be in a format that enables waypoints to be viewed with an Apple iPod. This is achieved by mapping waypoint fields into vCard fields that can be displayed as 'Contacts' on the iPod. With the iPod mounted as a hard disk (see your iPod manual for instructions), the resulting VCF file should be moved into the iPod 'Contacts' folder. As an alternative, Mac OS X users may prefer to drag the VCF file into their address book and synchronize with the iPod using iSync.

## encrypt option

Encrypt hints using ROT13.

By default geocaching hints are unencrypted; use this option to encrypt them.

## VidaOne GPS for Pocket PC (.gpb) (vidaone)

This format can...

- read and write tracks

This format supports the VidaOne [<http://www.vidaone.com>] GPS for pocket PC GPB file type.

This is have a very simple binary format which stores only the coordinates in the tracklog (.gpb) files.

## Vito Navigator II tracks (vitosmt)

This format can...

- read and write waypoints
- read and write tracks
- read and write routes

Vito Navigator II is a Pocket PC GPS application. This format reads a Vito Navigator II .SMT track file and can work in either waypoint or track mode. The speed, heading and Dilution of Position data is written in the notes field.

Support for writing .SMT tracks is very experimental and may crash VitoNavigator II on the Pocket PC.

## Vito SmartMap tracks (.vtt) (vitovtt)

This format can...

- read tracks

This format reads the binary (.vtt) track logs recorded by VITO SmartMap for Nokia Series 60 [[http://vitotechnology.com/en/products/smartmap\\_nokia\\_series60.html](http://vitotechnology.com/en/products/smartmap_nokia_series60.html)] 1.0, a GPS application for smartphones connected to NMEA 0183-compatible Bluetooth GPS receivers. It may work with .vtt files produced by versions of VITO SmartMap for other platforms.

This format was reverse engineered from a .vtt file. Currently, the coordinates, altitude, and time are available for each point recorded in a track. The course speed and heading fields have been identified, but the units are not certain and so those fields are ignored. The rest of the entry has not yet been decoded. The format uses little-endian byte ordering. The application displays metric units by default. Time is UTC.

**Table 3.4. track file header (8 bytes)**

Position	Field info
bytes 0-3	Probably a version field. Int value is 3 in sample file.
bytes 4-7	Number of points in file as int.

**Table 3.5. track point (32 bytes)**

Position	Field info
bytes 0-3	Decimal latitude multiplied by 20000000 as int.
bytes 4-7	Decimal longitude multiplied by 10000000 as int.
bytes 8-11	Altitude in meters as float.
bytes 12-13	Year, with century, as int.
byte 14	Month, ranging 1-12.
byte 15	Day of month, ranging 1-31.
byte 16	Hour, ranging 0-23.
byte 17	Minute, ranging 0-59.
bytes 18-21	Decimal second multiplied by 30000000 as int.
bytes 22-25	Probably speed in meters per second as float. Ranges 0~3 in file, seems reasonable since sample file was acquired on foot.
bytes 26-27	Probably decimal heading multiplied by something. Ranges between min and max values possible when decoded as integer. Doesn't change when speed field is 0. Doesn't change smoothly, jumps around a bit.
bytes 28-31	Status field of some kind. Changes only twice in file. May contain satellite count or PDOP info, as both are reported by the application's GUI.

## WiFiFoFum 2.0 for PocketPC XML (wfff)

This format can...

- read waypoints

WFFF is the export format for Aspecto Software's WiFiFoFum [<http://www.aspecto-software.com/rw/applications/wififofum/index.html>] 2.0 for Windows Mobile PCs.

It is a simple XML format that is read-only to GPSBabel and stores information about a WiFi stumbling session.

All WiFi-specific elements are written in the description field, similar to the netstumbler format.

### aicon option

Infrastructure closed icon name.

This option lets you specify an icon for infrastructure closed points.

### aioicon option

Infrastructure open icon name.

This option lets you specify an icon for infrastructure open points.



## ahcicon option

Ad-hoc closed icon name.

This options lets you specify an icon for an Ad-hoc, closed, waypoint.

## ahoicon option

Ad-hoc open icon name.

This options lets you specify an icon for an Ad-hoc, open, waypoint.

## snmac option

Shortname is MAC address.

This options lets you specify that the shortname of the waypoint is the MAC address.

# Wintec WBT-100/200 Binary File Format (wbt-bin)

This format can...

- read tracks

File protocol for the Wintec WBT-200™ and Wintec WBT-201™ (sometimes called the G-Rays 2™)GPS data loggers. This format reads the binary file format created by Wintec's Windows application.

Wintec WBT-201 [<http://www.semsons.com/wiwbmu3gpsre2.html>]

### Example 3.13. Command showing conversion of a Wintec binary file to GPX

```
gpsbabel -i wbt-bin -f tracks.bin -o gpx -F out.gpx
```

# Wintec WBT-100/200 GPS Download (wbt)

This format can...

- read waypoints
- read tracks

Serial download protocol for the Wintec WBT-200™ and Wintec WBT-201™ GPS data loggers. Although untested it is expected that this will also support the WBT-100.

Wintec WBT-201 [<http://www.semsons.com/wiwbmu3gpsre2.html>]

### Example 3.14. Command showing WBT-200 download and erase over Bluetooth on Mac OS X

```
gpsbabel -t -w -i wbt,erase -f /dev/cu.WBT200-SPPslave-1 -o gpx -F out.gpx
```

Internally, this is actually a serial device that has a serial/USB adapter built into it. It uses the CP210x chip by Silicon labs. You will probably need a driver for this chip. The product ships with one for Windows. The Linux 210x driver seems to work fine. Mac users will need to download the Mac driver for CP210x [[http://www.silabs.com/tgwWebApp/public/web\\_content/products/Microcontrollers/USB/en/mcu\\_vcp.htm](http://www.silabs.com/tgwWebApp/public/web_content/products/Microcontrollers/USB/en/mcu_vcp.htm)].

GPSTabel does not try to offer an interface to configure these units. That is left to the Windows software that comes with it or tools like the WBT 201 Viewer [<http://www.daria.co.uk/gps>] for Mac OS/X and Linux.

## erase option

Erase device data after download.

This option erases the track log from the device after download.

## Wintec WBT-201/G-Rays 2 Binary File Format (wbt-tk1)

This format can...

- read tracks

File protocol for the Wintec WBT-201 / G-Rays 2™ GPS data logger. This format reads the binary file format created by Wintec's Time Machine X application.

Wintec WBT-201 [<http://www.semsons.com/wiwbmu3gpsre2.html>]

### Example 3.15. Command showing conversion of a Wintec binary file to GPX

```
gpsbabel -w -t -i wbt-tk1 -f tracks.tk1 -o gpx -F out.gpx
```

## Yahoo Geocode API data (yahoo)

This format can...

- read waypoints

This format reads output from the Yahoo geocoding API [<http://developer.yahoo.com/maps/rest/V1/geocode.html>]. This feature of GPSTabel makes it easy to get geocoded results from Yahoo into your favorite mapping program, GPS receiver, or other format.

## addrsep option

String to separate concatenated address fields (default=", ").

This option specifies the string GPSTabel should use to separate the parts of the street address. Since most other formats supported by GPSTabel do not support street addresses, the street address fields from the Yahoo file are concatenated into the waypoint "notes" field.

The default value for this option is a comma followed by a space (", ").

---

# Chapter 4. Data Filters

GPSTabel supports data filtering. Data filters are invoked from the command line via the '-x' option. It should be noted that data filters are invoked in the internal pipeline at the point that corresponds to their position on the command. This implies that specifying a filter before reading any data ('-x <filter> -f <file>'), despite being legal, will not have any effect. The advantage is that filters 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.

Beware that most filters only apply to a certain kind of data. This is usually indicated below by referring to points, tracks or routes in the first sentence which describes each filter or in the table at [gpsbabel.org](http://www.gpsbabel.org/capabilities.html) [<http://www.gpsbabel.org/capabilities.html>] .

## Include Only Points Inside Polygon (polygon)

The polygon filter includes points if they are inside of a polygon. A polygon file looks like an arc file, except that the arc it describes must be a closed cycle. That is, for a simple polygon, the first and last points must be the same. Here's a square:

```
# A square (not really) polygon
41.0000      -85.0000
41.0000      -86.0000
42.0000      -86.0000
42.0000      -85.0000
41.0000      -85.0000
```

Polygons may include islands and holes. To include an island or a hole, just append it to the main polygon.

```
# A square polygon with a triangular hole
41.0000      -85.0000
41.0000      -86.0000
42.0000      -86.0000
42.0000      -85.0000
41.0000      -85.0000
# The hole begins here
41.5000      -85.5000
41.6000      -85.5000
41.6000      -85.6000
41.5000      -85.5000
```

As with the arc filter, you define a polygon by giving the name of the file that contains it, using the `file` option.

Note that this filter currently will not work properly if your polygon contains one or both poles or if it spans the line of 180 degrees east or west longitude.

### Example 4.1. Using the polygon filter

Suppose you have a polygon file that defines the border of your county, called `mycounty.txt`. This command line will give you only the points in your county:

```
gpsbabel -i geo -f 1.loc -x polygon,file=mycounty.txt -o mapsend -F 2.wpt
```

### Example 4.2. Using the polygon and arc filters to find points in or nearly in a polygon

Because the polygon and arc filters use the same file format, you can use them together to find all points that are "in or nearly in" a polygon. This can be useful if your waypoints or the boundaries of your polygon are not quite perfect, so you want to provide a buffer zone around it in case there are points nearby that should be in the polygon but aren't quite.

```
gpsbabel -i gpx -f points.gpx -x stack,push -x polygon,file=mycounty.txt  
-x stack,swap -x arc,file=mycounty.txt,distance=1k -x stack,pop,append  
-x duplicate,shortname -o gpx -F nearmycounty.gpx
```

This command makes a copy of the points, finds the ones that are in your county, swaps that result with the copy of the original set of points, finds the ones from that set that are within 1 km of the border of the county, puts the two lists together, and then filters out any points that appear twice (This step is necessary because points inside the county but near the county line will be kept by both the polygon and the arc filter.)

## file option

File containing vertices of polygon.

This option is required.

This option specifies the name of the file containing the polygon to use for filtering. The format of the file is as described above.

GPSTabel supports converting any route or track to a file usable by this filter; simply read it in the normal way and write it using the arc file format. Afterward, you will need to make sure that the first point and the last point in the file are the same, as the polygon filter depends on that. You can do so with any text editor.

## exclude option

Exclude points inside the polygon.

When this option is specified, the usual sense of the polygon filter is reversed. That is, points that are inside the polygon are discarded while points that are further away are kept.

## Include Only Points Within Distance of Arc (arc)

This filter keeps or removes waypoints based on their proximity to an arc, which is a series of connected line segments similar to a route or a track but without any associated data other than the coordinates.

The arc is defined in a file whose name must be provided with the `file`. That file contains pairs of coordinates for the vertices of the arc, one coordinate pair per line. Comments may be included by preceding them with a '#' character. An arc file looks something like this sample:

```
# Lima Road/SR3 north of Fort Wayne, Indiana
41.150064468 -85.166207433
41.150064468 -85.165371895
41.149034500 -85.165157318
41.147832870 -85.164771080
41.146631241 -85.164384842
41.144270897 -85.163655281
41.141953468 -85.162882805
```

An arc file may optionally contain gaps in the arc. You may specify such a gap by inserting a line containing "#break" either on a line by itself or after the coordinates of the starting point of the new arc segment.

### Example 4.3. Using the arc filter

Assuming the arc above is in a file called `lima_rd.txt`, the following command line would include only points within one mile of the section of Lima Road covered by the arc.

```
gpsbabel -i geo -f 1.loc -x arc,file=lima_rd.txt,distance=1 -o mapsend
-F 2.wpt
```

## file option

File containing vertices of arc.

This option is required.

This option specifies the name of the file containing the arc to use for filtering. The format of the file is as described above.

GPSTabel supports converting any route or track to a file usable by this filter; simply read it in the normal way and write it using the arc file format.

## distance option

Maximum distance from arc.

This option is not required, but if it is not specified the distance defaults to zero miles, which isn't very useful.

This option specifies the maximum distance a point may be from the arc without being discarded. Points that are closer to the arc are kept, while points that are further away are discarded.

Distances may be specified in miles (3M) or kilometers (5K). If no units are specified, the distance is assumed to be in miles.

## exclude option

Exclude points close to the arc.

When this option is specified, the usual sense of the arc filter is reversed. That is, points that are closer than `distance` are discarded while points that are further away are kept.

## points option

Use distance from vertices not lines.

When this option is specified, only points that are within the specified distance of one of the vertices of the arc are kept. This differs from the normal mode of operation in that in the normal mode, points that are close to the lines between points are also kept.

This option makes the arc filter act like a multi-point version of the radius filter.

## Include Only Points Within Radius (radius)

This filter includes or excludes waypoints based on their proximity to a central point. All waypoints more than the specified distance from the specified point will be removed from the dataset.

By default, all remaining points are sorted so that points closer to the center appear earlier in the output file.

### Example 4.4. Using the radius filter to find points close to a given point

This example command line would include only points within 1 1/2 miles of N30.000 W 90.000

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

## lat option

Latitude for center point (D.DDDDD).

This option is required.

This option specifies the latitude of the central point in decimal degrees. South latitudes should be expressed as a negative number. Valid values for this option are from -90 to 90.

## lon option

Longitude for center point (D.DDDDD).

This option is required.

This option specifies the longitude of the central point in decimal degrees. West longitudes should be expressed as a negative number. Valid values for this option are from -180 to 180.

## distance option

Maximum distance from center.

This option is required.

This option specifies the maximum distance a point may be from the central point in order to remain in the dataset. Points closer than this distance will be kept and points further away will be removed (unless the `exclude` option is specified.)

Distances may be expressed in miles (3M) or kilometers (4K). If no units are provided, the distance is assumed to be in miles.

## exclude option

Exclude points close to center.

If this option is included, the action of the radius filter will be reversed: points within the given distance will be removed, and points further away will be kept.

## nosort option

Inhibit sort by distance to center.

If this option is specified, the radius filter will not sort the remaining points by distance from the center. They will remain in whatever order they were originally.

## maxcount option

Output no more than this number of points.

This option specifies the maximum number of points that the radius filter may keep. If there are more than this number of points within the specified distance of the center, the more distant points will be discarded even though they are within the specified distance. If this option is not specified, all points are kept regardless of how many there are.

Note that if the `nosort` option is also specified, this option will instead keep points based on their position within the input file rather than on their distance from the center. This may or may not be what you want.

Note, too, that this option may be used with the `exclude` option, but the results might not be what you expect. In particular, the results will not be the same as if you had kept all of the points you'd otherwise throw away. You will still get no more than `maxcount` points, but they will all be at least `distance` away from the center. (And possibly sorted.)

## asroute option

Put resulting waypoints in route of this name.

This option specifies the name of a route. If this option is specified, the radius filter puts all points that are kept into a route with the given name. The order of points in the route is by distance from the center (unless the `nosort` option is also specified.)

Note that this route is not necessarily the most efficient route to visit all of the points. In fact, for some data sets, it might be the least efficient route.

## Interpolate between trackpoints (interpolate)

This filter modifies any tracks so that either the distance or the time between consecutive points is no less than the specified interval. Where points are missing, the filter fills them in by following a straight

line (actually a great circle) between the adjacent points. You must specify either the `distance` or the `time` option.

### Example 4.5. Using the interpolate filter

This command line reads `track.gpx` and inserts points wherever two adjacent trackpoints are more than 10 seconds apart:

```
gpsbabel -i gpx -f track.gpx -x interpolate,time=10 -o gpx -F newtrack.gpx
```

This command reads `track.gpx` and inserts points wherever two adjacent trackpoints are more than 15 kilometers apart:

```
gpsbabel -i gpx -f track.gpx -x interpolate,distance=15k -o gpx -F newtrack.gpx
```

This command reads `track.gpx` and inserts points wherever two adjacent trackpoints are more than 2 miles apart:

```
gpsbabel -i gpx -f track.gpx -x interpolate,distance=2m -o gpx -F newtrack.gpx
```

## time option

Time interval in seconds.

This option specifies the maximum allowable time interval between points in the track. If two points in the track are further apart than this value, new points will be inserted between them.

This value is always specified in units of seconds.

Either this option or the `distance` must be specified.

## distance option

Distance interval in miles or kilometers.

This option specifies the maximum allowable distance between points in the track. If two points in the track are further apart than this value, new points will be inserted between them.

This value may be specified in units of miles (3M) or kilometers (5K). If no units are specified, the units are assumed to be miles.

Either this option or the `time` must be specified.

## route option

Interpolate routes instead.

If this option is specified, the interpolate filter interpolates routes rather than tracks. Because route points do not have time stamps, it is an error to use this option with the `time` option.



# Manipulate track lists (track)

WARNING: This filter always drops empty tracks.

This filter performs various operations on track data.

## move option

Correct trackpoint timestamps by a delta.

This option changes the time of all trackpoints. This might be useful if your track must be moved by one or more hours because of an incorrect time zone.

### Example 4.6. Time-shifting a track with the track filter

The following command line will shift all tracks to be one hour later.

```
gpsbabel -t -i gpx -f in.gpx -x track,move=+1h -o gpx -F out.gpx
```

## pack option

Pack all tracks into one.

This option causes all tracks to be appended to one another to form a single track. This option does not work if any two tracks overlap in time; in that case, consider using the merge option.

This option is most useful for rejoining tracks that might have been interrupted by an equipment malfunction or an overnight stop.

If no other option is given to the track filter, this option is assumed.

## split option

Split by date or time interval (see README).

The input track will be split into several tracks depending on date of track points. If there is more than one track, use the pack option before using this. To split a single tracks into separate tracks for each day and name them, use this:

```
gpsbabel -t -i gpx -f in.gpx -x track,split,title="ACTIVE LOG # %Y%m%d" -o gpx -F out.gpx
```

If the input has multiple tracks, pack them together before splitting them back apart per day thusly:

```
gpsbabel -t -i gpx -f in.gpx -x track,pack,split,title="ACTIVE LOG # %D" -o gpx -F out.gpx
```

Additionally you can add an interval to the split option. With this the track will be split if the time between two points is greater than this parameter. The interval must be numeric and can be int days, hours, minutes

or seconds, expressed as one of the character "d", "h", "m", or "s". If no trailing character is present, the units are assumed to be in seconds.

For example, to split a track based on an four hour interval, use this:

```
gpsbabel -t -i gpx -f in.gpx -x track,pack,split=4h,title="LOG # %c"
-o gpx -F out.gpx
```

## sdistance option

Split by distance.

The input track will be split into several tracks if the distance between successive track points is greater than the distance given as a parameter. The distance must be numeric and can be in miles or kilometers, expressed as one of the character "k", or "m". If `sdistance` is given no parameters, this option has the same effect as the `split` option without parameters. If there is more than one track, use the `pack` option before using this.

For example, to split the track if the distance between points is greater than 100 meters, use this:

```
gpsbabel -t -i gpx -f in.gpx -x track,pack,sdistance=0.1k" -o gpx -
F out.gpx
```

The `sdistance` option can be combined with the `split` option. The track then will be split only if both time and distance interval exceeds the supplied values. This technique can be used to filter out gaps from the tracklog. The gap is kept only if the gps device is without signal for longer time than that given and during that time it moves a distance over that given. This example splits the track if the device is without signal for at least 5 minutes and during this time moves more than 300 meters:

```
gpsbabel -t -i gpx -f in.gpx -x track,pack,sdistance=0.3k,split=5m -
o gpx -F out.gpx
```

## merge option

Merge multiple tracks for the same way.

This option puts all track points from all tracks into a single track and sorts them by time stamp. Points with identical time stamps will be dropped.

### Example 4.7. Merging tracks with the track filter

Suppose you want to merge tracks recorded with two different GPS devices at the same time. To do that, use this command line:

```
gpsbabel -t -i gpx -f john.gpx -i gpx -f doe.gpx -x
track,merge,title="COMBINED LOG" -o gpx -F john_doe.gpx
```

## name option

Use only track(s) where title matches given name.

With the name option you can filter out a track by title.

The comparison is always non-case-sensitive. Wildcards are allowed.

## start option

Use only track points after this timestamp.

This option is used along with the `stop` to discard trackpoints that were recorded outside of a specific period of time. This option specifies the beginning of the time period.

If this option is not specified, the time period is assumed to begin at the dawn of time or January 1, 1970, whichever was later. The time for this option is expressed in UTC.

The value of this option must be in the form of `YYYYMMDDHHMMSS`, but it is not necessary to specify the smaller time units if they are not needed. That is, if you only care about points logged between 10 AM and 6 PM on a given date, you need not specify the minutes or seconds.

### Example 4.8. Extracting a period of time with the track filter

To get only the parts of a track that were mapped on 20 July 2005 between 10 AM and 6 PM, use this command line:

```
gpsbabel -t -i gpx -f in.gpx -x track,start=2005072010,stop=2005072018  
-o gpx -F out.gpx
```

## stop option

Use only track points before this timestamp.

This option is used in conjunction with the `start` option to discard all trackpoints outside of a given period of time. This option defines the end of the time period.

If this option is not specified, the time period is assumed to end at the end of civilization as we know it or the year 2038, whichever comes first. The time for this option is expressed in UTC.

See the `start` option for the format of this value and an example of usage.

## title option

Basic title for new track(s).

This option specifies a title for tracks generated by the track filter. By default, the title of the new track is composed of the start time of the track appended to this value.

If this value contains a percent (%) character, it is treated as a format string for the POSIX `strftime` function, allowing custom time-based track names.

## fix option

Synthesize GPS fixes (PPS, DGPS, 3D, 2D, NONE).

This option sets the GPS fix status for all trackpoints to the specified value. Valid values for this option are PPS, DGPS, 3D, 2D, or NONE.

This option is most useful when converting from a format that doesn't contain GPS fix status to one that requires it.

## course option

Synthesize course.

This option computes (or recomputes) a value for the GPS heading at each trackpoint. This is most useful with trackpoints from formats that don't support heading information or for trackpoints synthesized by the interpolate filter. The heading at each trackpoint is simply the course from the previous trackpoint in the track. The first trackpoint in each track is arbitrarily assigned a heading of 0 degrees.

## speed option

Synthesize speed.

This option computes a value for the GPS speed at each trackpoint. This is most useful with trackpoints from formats that don't support speed information or for trackpoints synthesized by the interpolate filter. The speed at each trackpoint is the average speed from the previous trackpoint (distance divided by time). The first trackpoint in each track is assigned a speed of "unknown."

# Rearrange waypoints by resorting (sort)

This filter sorts waypoints into alphabetical order by the selected field. You must specify exactly one of the options.

## gcid option

Sort by numeric geocache ID.

If the data contains Groundspeak geocache IDs, this option causes the waypoints to be sorted in alphabetical order by geocache ID.

This option is not valid in combination with any other option.

## shortname option

Sort by waypoint short name.

This option causes the waypoints to be sorted in alphabetical order by short name.

This option is not valid in combination with any other option.

## description option

Sort by waypoint description.

This option causes the waypoints to be sorted in alphabetical order by description.

This option is not valid in combination with any other option.

## time option

Sort by time.

This option causes the waypoints to be sorted in chronological order by creation time.

This option is not valid in combination with any other option.

## Remove all waypoints, tracks, or routes (nuketypes)

There are three main types of data that GPSTools deals with: waypoints, tracks, and routes. The nuketypes filter allows removing all the data of any or all of those three types.

### Example 4.9. Filtering data types with nuketypes

If you have a GPX file that contains routes, tracks, and waypoints and you want a GPX file that contains only tracks, you may use this filter to remove the waypoints and the routes with this command:

```
gpsbabel -i gpx -f bigfile.gpx -x nuketypes,waypoints,routes -o gpx -F trackonly.gpx
```

## waypoints option

Remove all waypoints from data stream.

This option causes the nuketypes filter to discard all waypoints that are not associated with a track or route.

## tracks option

Remove all tracks from data stream.

This option causes the nuketypes filter to discard all track data.

## routes option

Remove all routes from data stream.

This option causes the nuketypes filter to discard all route data.

## Remove Duplicates (duplicate)

The duplicate filter is designed to remove duplicate points based on their short name (traditionally a waypoint's name on the GPS receiver), and/or their location (to a precision of 6 decimals). This filter supports two options that specify how duplicates will be recognized, `shortname` and `location`. Generally, at least one of these options is required.

### Example 4.10. Using the duplicate filter to suppress points with the same name and location

This command line removes points that have duplicate short names and duplicate locations. The result would be a gpx file that more than likely contains only unique points and point data.

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

## shortname option

Suppress duplicate waypoints based on name.

This option is the one most often used with the duplicate filter. This option instructs the duplicate filter to remove any waypoints that share a short name with a waypoint that has come before. This option might be used to remove duplicates if you are merging two datasets that were each created in part from a common ancestor dataset.

## location option

Suppress duplicate waypoint based on coords.

This option causes the duplicate filter to remove any additional waypoint that has the same coordinates (to six decimal degrees) as a waypoint that came before. This option may be used to remove duplicate waypoints if the names are not expected to be the same. It also might be used along with the `shortname` option to remove duplicate waypoints if the names of several unrelated groups of waypoints might be the same.

## all option

Suppress all instances of duplicates.

When this option is specified, GPSTabel will remove all instances of a duplicated waypoint, not just the second and subsequent instances. If your input file contains waypoints A, B, B, and C, the output file will contain waypoints A, B, and C without the `all` option, or just A and C with the `all` option.

### Example 4.11. Using the duplicate filter to implement an "ignore list."

This option may be used to implement an "ignore list." In the following example, the duplicate filter is used to remove a list of waypoints to be ignored from a larger collection of waypoints:

```
gpsbabel -i gpx -f waypoints.gpx -i csv -f to_ignore.csv -x  
duplicate,shortname,all -o gpx -F filtered.gpx
```

## correct option

Use coords from duplicate points.

This option is used to change the locations of waypoints without losing any of the other associated information. When this option is specified, the latitude and longitude from later duplicates will replace the latitude and longitude in the original waypoint.

As an example, this option may be used to adjust the locations of "puzzle" geocaches in a Groundspeak pocket query:

**Example 4.12. Using the duplicate filter to correct the locations of "puzzle" geocaches**

```
gpsbabel -i gpx -f 43622.gpx -i csv -f corrections.csv -x
duplicate,shortname,correct -o gpx -F 43622-corrected.gpx
```

After this command is run, the waypoints in the output file will have all of the descriptive information from 43622.gpx, but waypoints that were also found in corrections.csv will have their coordinates replaced with the coordinates from that file.

## Remove Points Within Distance (position)

This filter removes points based on their proximity to each other. A point is removed if it is within the specified distance of a point that has come before.

**Example 4.13. Using the position filter to suppress close points**

The following command removes multiple points that are within one foot of each other, leaving just one.

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

### distance option

Maximum positional distance.

This option specifies the minimum allowable distance between two points. If two points are closer than this distance, only one of them is kept.

Distances may be expressed in feet (30f) or meters (10m). If no unit is specified, the distance is assumed to be in feet.

### all option

Suppress all points close to other points.

This option causes the position filter to remove all points that are within the specified distance of one another, rather than leaving just one of them.

This option may be used to entirely remove clusters of points.

## Remove unreliable points with high hdop or vdop (discard)

This filter is used to "fix" unreliable GPS data by discarding points that are believed to be unreliable. You may specify an HDOP and/or VDOP above a specified limit, a minimum number of satellites that must have been in view for a fix to be considered, or both.

HDOP and VDOP are measures of the best possible horizontal or vertical precision for a given configuration of GPS satellites. Higher numbers indicate a higher dilution of precision and therefore mathematically less useful.

**Example 4.14. Using the discard filter for HDOP and VDOP.**

```
gpsbabel -i gpx -f in.gpx -x discard,hdop=10,vdop=20,hdopandvdop -o  
gpx -F out.gpx
```

You may specify a minimum number of satellites.

**Example 4.15. Using the discard filter to require at least three satellites.**

```
gpsbabel -i gpx -f in.gpx -x discard,sat=3 -o gpx -F out.gpx
```

Contributed by Tobias Minich and Serge Droz.

## hdop option

Suppress waypoints with higher hdop.

This option specifies the maximum allowable Horizontal Dilution of Precision (HDOP). By default, any point with an HDOP in excess of this value will be discarded regardless of its VDOP, but see `hdopandvdop`.

## vdop option

Suppress waypoints with higher vdop.

This option specifies the maximum allowable Vertical Dilution of Precision (VDOP). By default, any point with an VDOP in excess of this value will be discarded regardless of its HDOP, but see `hdopandvdop`.

## hdopandvdop option

Link hdop and vdop suppression with AND.

If this option is used, only points that exceed both the maximum allowable HDOP and the maximum allowable VDOP will be discarded. This option requires that both the `hdop` and `vdop` options be specified.

## sat option

Minimum sats to keep waypoints.

This option specifies the minimum required number of satellites.

## Reverse stops within routes (reverse)

The reverse filter is used to reverse tracks and routes. It's mostly useful for those few formats where track/route sequence matters and there isn't a way to reverse them using the program itself.



The reversal is performed in the laziest way possible. Timestamps are kept with the original waypoints so the resulting track or route will have the interesting characteristic that time runs backwards. This tends to make Magellan Mapsend, in particular, do a wierd thing and place each waypoint on a separate day.

Additionally, if you're using this to reverse a route that navigates, say, an exit ramp or a one way street, you will be in for unpleasant ride. application cares about timestamps

## Save and restore waypoint lists (stack)

This filter is designed to solve advanced problems that involve shuffling multiple lists of waypoints, tracks, or routes.

The stack filter can be used to save the current state of the entire collection of data. That state is placed on top of a stack of collections, so you can simultaneously have as many stored collections of data as you can fit in your computer's memory.

The stack filter can be used in conjunction with other filters to implement a "union" or "logical or" functionality. The basic idea is to use the stack to store copies of the original list of waypoints, then use the 'swap' function to replace each copy with a filtered list. Finally, append all of the filtered lists to create one big list, which is then output. The following example finds a list of all points that are either inside county A or inside county B. Any points that are inside both counties are duplicated (but the duplicates can be removed with the DUPLICATE filter; see above.)

```
gpsbabel -i gpx -f in.gpx -x stack,push,copy -x
polygon,file=county_a.txt -x stack,swap -x polygon,file=county_b.txt -
x stack,pop,append -o gpx -F out.gpx
```

This example reads a large list of waypoints and extracts the points within 20 miles of each of two cities, writing the waypoint descriptions into two different PalmDoc files and exporting all of the points to the GPS receiver:

```
gpsbabel -i gpx -f indiana.gpx -x
stack,push,copy -x radius,lat=41.0765,lon=-85.1365,distance=20m -
o palmdoc,dbname=Fort\ Wayne -F fortwayne.pdb -x
stack,swap -x radius,lat=39.7733,lon=-86.1433,distance=20m -o
palmdoc,dbname=Indianapolis -F indianapolis.pdb -x stack,pop,append -
o magellan -F fwaind.wpt
```

## push option

Push waypoint list onto stack.

This is one of three "primary" options to the stack filter.

When this option is specified, the current state is pushed onto the top of the stack. By default, the current state is then cleared, but the `copy` option can be used to cause it to be saved.

## pop option

Pop waypoint list from stack.

This is one of three "primary" options to the stack filter.

This option "pops" the collection of data from the top of the stack. By default, the saved state replaces the current state, but see the `discard` and `append` options for alternatives.

## swap option

Swap waypoint list with `<depth>` item on stack.

This is one of three "primary" options to the stack filter.

When this option is specified, the current state is swapped with a saved state from the stack. By default, it is swapped with the top of the stack, but the `depth` can be used to specify a different saved state.

## copy option

(push) Copy waypoint list.

This option is only valid when used with the `push` option. When this option is specified, a copy of the current state is pushed onto the stack but the current state is left unchanged. Otherwise, the push operation clears the current data collection.

## append option

(pop) Append list.

This option is only valid in conjunction with the `pop`. When it is specified, the topmost collection of data from the stack is appended to the current collection of data.

## discard option

(pop) Discard top of stack.

This option is only valid when used with the `pop` option. When this option is specified, the popped state is discarded and the current state remains unchanged.

## replace option

(pop) Replace list (default).

This option is only valid when used with the `pop` option. This is the default behavior of the `pop` option, so you should never need to specify it, but it is included for the sake of readability. When this option is specified, the popped state replaces the current state.

## depth option

(swap) Item to use (default=1).

This option is only valid when used along with the `swap` option. If specified, it indicates which item on the stack should be swapped with the current state. The default value is 1, which corresponds to the top of the stack.

## Simplify routes (simplify)

The Simplify filter is used to simplify routes and tracks for use with formats that limit the number of points they can contain or just to reduce the complexity of a route.

The filter attempts to remove points from each route until the number of points or the error is within the given bounds, while also attempting to preserve the shape of the original route as much as possible.

The quality of the results will vary depending on the density of points in the original route and the length of the original route.

For example, suppose you have a route from Street Atlas 2003 that you wish to use with a Magellan GPS receiver that only supports up to 50 points in a route:

```
gpsbabel -r -i saroute -f RoadTrip.anr -x simplify,count=50 -o magellan  
-F grocery.rte
```

### count option

Maximum number of points in route.

This option specifies the maximum number of points which may appear in the simplified route. For example, if you specify "count=50", all resulting routes will contain 50 points or fewer.

You must specify either this option or the error option.

### error option

Maximum error.

This option specifies the maximum allowable error that may be introduced by removing a single point. The value of this option is a distance, specified in miles by default. You may also specify the distance in kilometers by adding a 'k' to the end of the number.

How the error is determined depends on whether the `length` or `crosstrack` method is used. If you are using the `length` method, the error is the change in the length of the route introduced by removing a point. If you are using the `crosstrack` method, the error is the distance from the point to the line that results if that point is removed.

### crosstrack option

Use cross-track error (default).

This option instructs GPSBabel to remove points that have the smallest overall effect on the overall shape of the route. Using this method, the first point to be removed will be the one that is closest to a line drawn between the two points adjacent to it.

If neither this option nor the `length` option is specified, this is the default.

### length option

Use arclength error.

This option instructs GPSTransform to simplify by removing points that cause the smallest change in the overall length of the route first.

## Transform waypoints into a route, tracks into routes, ... (transform)

This filter can be used to convert GPS data between different data types.

Some GPS data formats support only some subset of waypoints, tracks, and routes. The transform filter allows you to convert between these types. For example, it can be used to convert a pile of waypoints (such as those from a CSV file) into a track or vice versa.

The following example show you how to create a route from a waypoint table.

```
gpsbabel -i csv waypoints.txt -x transform,rte=wpt -o gpx -F route.gpx
```

Only the first letter of option value decides which transformation will be done. Depending on the used option it can be only 'W' for waypoints, 'R' for routes or 'T' for tracks.

### wpt option

Transform track(s) or route(s) into waypoint(s) [R/T].

This option selects the destination type of this filter to be waypoints. Choose this when you want to convert tracks or routes into waypoints.

#### Example 4.16. Converting a track to a sequence of waypoints

Say you you have a KML file that contains a track but you want to convert it to a CSV file that can contain only waypoints, perhaps to import into a spreadsheet. Use the following command:

```
gpsbabel -i kml -f blah.kml -x transform,wpt=trk -o csv -F blah.txt
```

### rte option

Transform waypoint(s) or track(s) into route(s) [W/T].

This option selects the destination type of this filter to be routes. Choose this when you want to convert tracks into waypoints routes. A single route will be created in the sequence they appear in the input.

#### Example 4.17. Converting a pile of waypoints to a GPX route

Say you you have a data file that came from CSV file that you want to convert to a GPX route that can be loaded into Mapsource. Use the following command:

```
gpsbabel -i csv -f blah.txt -x transform,rte=wpt -o gdb -F blah.gdb
```

### trk option

Transform waypoint(s) or route(s) into tracks(s) [W/R].

This option selects the destination type of this filter to be tracks. Choose this when you want to create tracks from a list of waypoints or routes. A single track will be created in the sequence they appear in the input.

#### **Example 4.18. Converting a pile of waypoints to a GPX track**

Say you have a data file that came from CSV file that you want to convert to a GPX track that can be loaded into Mapsource. Use the following command:

```
gpsbabel -i csv -f blah.txt -x transform,trk=wpt -o gdb -F blah.gdb
```

## **del option**

Delete source data after transformation.

This option, when used in connection with the wpt, rte, or trk options, tells GPSTBabel to delete the source data after conversion. This is most useful if you are trying to avoid duplicated data in the output.

#### **Example 4.19. Convert a GPX track to GPX waypoints, tossing the original track**

```
gpsbabel -i gpx -f blah.gpx -x transform,wpt=trk,del -o gpx -F converted.gpx
```

---

# Appendix A. Supported Datums

Some formats in GPSBabel support multiple datums. For example, the `datum` option to the `garmin_txt` format allows you to specify a datum for the output file.

The following is a list of the datums supported by GPSBabel.

Adindan	Cuba NAD27	La Reunion	Qornoq
AFG	Cyprus	Liberia 1964	Quatar National
Ain-El-Abd	Djakarta(Batavia)	Luzon	Rome 1940
Alaska-NAD27	DOS 1968	Mahe 1971	S-42(Pulkovo1942)
Alaska-Canada	Easter Island 1967	Marco Astro	S.E.Asia_(Indian)
Anna-1-Astro	Egypt	Masirah Is. Nahrwan	SAD-69/Brazil
ARC 1950 Mean	European 1950	Massawa	Santa Braz
ARC 1960 Mean	European 1950 mean	Merchich	Santo (DOS)
Asc Island 58	European 1979 mean	Mexico NAD27	Sapper Hill 43
Astro B4	Finnish Nautical	Midway Astro 61	Schwarzeck
Astro Beacon E	Gandajika Base	Mindanao	Sicily
Astro pos 71/4	Geodetic Datum 49	Minna	Sierra Leone 1960
Astro stn 52	Ghana	Montjong Lowe	S. Am. 1969 mean
Australia Geo 1984	Greenland NAD27	Nahrwan	South Asia
Bahamas NAD27	Guam 1963	Naparima BWI	Southeast Base
Bellevue IGN	Gunung Segara	North America 83	Southwest Base
Bermuda 1957	Gunung Serindung 1962	N. America 1927 mean	Tananarive Obs 25
Bukit Rimpah	GUX1 Astro	Observatorio 1966	Thai/Viet (Indian)
Camp_Area_Astro	Herat North	Old Egyptian	Timbalai 1948
Campo_Inchauspe	Hjorsey 1955	Old Hawaiian_mean	Tokyo mean
Canada_Mean(NAD27)	Hong Kong 1963	Old Hawaiian Kauai	Tristan Astro 1968
Canal_Zone_(NAD27)	Hu-Tzu-Shan	Old Hawaiian Maui	United Arab Emirates
Canton_Island_1966	Indian	Old Hawaiian Oahu	Viti Levu 1916
Cape	Iran	Oman	Wake Eniwetok 60
Cape_Canaveral_mean	Ireland 1965	OSGB36	WGS 72
Carribean NAD27	ISTS 073 Astro 69	Pico De Las Nieves	WGS 84
Carthage	Johnston Island 61	Pitcairn Astro 67	Yacare
Cent America NAD27	Kandawala	S. Am. 1956 mean(P)	Zanderij
Chatham 1971	Kerguelen Island	S. Chilean 1963 (P)	Sweden
Chua Astro	Kertau 48	Puerto Rico	
Corrego Alegre	L.C. 5 Astro	Pulkovo 1942	

# Appendix B. Garmin Icons

Following is a list of the valid values for the `garmin def icon` option. These values are also used internally by the GDB, BCR, Mapsource, Geoniche, GPilotS, PCX, and PSITrex formats.

ATV	Contact, Glasses	Hunting Area	Number 0, Green	Scales
Airport	Contact, Goatee	Ice Skating	Number 0, Red	Scenic Area
Amusement Park	Contact, Kung-Fu	Information	Number 1, Blue	School
Anchor	Contact, Panda	Intersection	Number 1, Green	Seafood
Anchor Prohibited	Contact, Pig	Intl freeway hwy	Number 1, Red	Seaplane Base
Animal Tracks	Contact, Pirate	Intl national hwy	Number 2, Blue	Shipwreck
Asian Food	Contact, Ranger	Italian food	Number 2, Green	Shopping Center
Bait and Tackle	Contact, Smiley	Large intersection	RampNumber 2, Red	Short Tower
Ball Park	Contact, Spike	Large exit without services	Number 3, Blue	Shower
Bank	Contact, Sumo	Letter A, Blue	Number 3, Green	Ski Resort
Bar	Controlled Area	Letter A, Green	Number 3, Red	Skiing Area
Beach	Convenience Store	Letter A, Red	Number 4, Blue	Skull and Crossbones
Beacon	Cover	Letter B, Blue	Number 4, Green	Small City
Bell	Covey	Letter B, Green	Number 4, Red	Small Game
Big Game	Crossing	Letter B, Red	Number 5, Blue	Soft Field
Bike Trail	Dam	Letter C, Blue	Number 5, Green	Square, Blue
Blind	Danger Area	Letter C, Green	Number 5, Red	Square, Green
Block, Blue	Deli	Letter C, Red	Number 6, Blue	Square, Red
Block, Green	Department Store	Letter D, Blue	Number 6, Green	Stadium
Block, Red	Diamond, Blue	Letter D, Green	Number 6, Red	State Hwy
Blood Trail	Diamond, Green	Letter D, Red	Number 7, Blue	Steak
Boat Ramp	Diamond, Red	Letterbox Cache	Number 7, Green	Street Intersection
Border Crossing (PortDiver Down Flag 1 Of Entry)	Diver Down Flag 1	Levee	Number 7, Red	Stump
Bottom Conditions	Diver Down Flag 2	Library	Number 8, Blue	Summit
Bowling	Dock	Light	Number 8, Green	Swimming Area
Bridge	Dot, White	Live Theater	Number 8, Red	TACAN
Building	Drinking Water	Localizer Outer Marker	Number 9, Blue	Tall Tower
Buoy, White	Dropoff	Locationless (Reverse)Cache	Number 9, Green	Telephone
Campground	Elevation point	Lodge	Number 9, Red	Tide/Current PRediction Station
Car	Event Cache	Lodging	Oil Field	Toll Booth
Car Rental	Exit	Man Overboard	Open 24 Hours	TracBack Point
Car Repair	Exit without services	Marina	Oval, Blue	Trail Head
Cemetery	Fast Food	Medical Facility	Oval, Green	Tree Stand
Church	First approach fix	Micro-Cache	Oval, Red	Treed Quarry
Circle with X	Fishing Area	Mile Marker	Parachute Area	Triangle, Blue
Circle, Blue	Fishing Hot Spot	Military	Park	Triangle, Green
Circle, Green	Fitness Center	Mine	Parking Area	Triangle, Red
Circle, Red	Flag	Missed approach point	Pharmacy	Truck
City (Capitol)	Flag, Blue	Movie Theater	Picnic Area	Truck Stop
City (Large)	Flag, Green	Multi-Cache	Pin, Blue	Tunnel
City (Medium)	Flag, Red	Multi-Cache	Pin, Green	U Marina
City (Small)	Food Source	Museum	Pin, Red	U stump

---

Garmin Icons

---

City Hall	Forest	Navaid, Amber	Pizza	US hwy
Civil	Furbearer	Navaid, Black	Police Station	Ultralight Area
Coast Guard	Gambling/casino	Navaid, Blue	Post Office	Unknown Cache
Contact, Afro	Gas Station	Navaid, Green	Post Office	Upland Game
Contact, Alien	Geocache	Navaid, Green/Red	Private Field	VHF Omni-range
Contact, Ball Cap	Geocache Found	Navaid, Green/White	Puzzle Cache	VOR-DME
Contact, Big Ears	Geographic name, Man-made	placeNavaid, Orange	RV Park	VOR/TACAN
Contact, Biker	Geographic name, land	placeNavaid, Red	Radio Beacon	Virtual cache
Contact, Blonde	Geographic name, water	placeNavaid, Red/Green	Ramp intersection	Water Hydrant
Contact, Bug	Ghost Town	Navaid, Red/White	Rectangle, Blue	Water Source
Contact, Cat	Glider Area	Navaid, Violet	Rectangle, Green	Waterfowl
Contact, Clown	Golf Course	Navaid, White	Rectangle, Red	Waypoint
Contact, Dog	Ground Transportation	Navaid, White/Green	Reef	Webcam Cache
Contact, Dreadlocks	Heliport	Navaid, White/Red	Residence	Weed Bed
Contact, Female1	Horn	Non-directional beacon	Restaurant	Winery
Contact, Female2	Hotel	Null	Restricted Area	Wrecker
Contact, Female3	House	Number 0, Blue	Restroom	Zoo



---

# Appendix C. GPSTBabel XCSV Style Files

## Introduction

Often it is desirable to add a new file format for "one-off" work (perhaps you want to export something to a spreadsheet or graphing program) or to read a format that GPSTBabel does not yet support. For suitably simple formats, this can be done by a user with no programming experience by providing a GPSTBabel style file.

For a format to be described by a style file, it must be predictable and generally readable by human. Formats with binary or unreadable content are not good fits for this scheme. It should have:

A fixed header at the beginning, if it has any at all. This is called a 'prologue'.

Waypoints that are grouped by fixed separators, often a newline. In style file parlance, this is called a 'record'.

Traits of that waypoint described in that record. In the style files, these are called 'fields' and examples may include longitude or a name.

Fields that are grouped by fixed separators, often a comma or a tab. In the style files, this is called the field separator.

A fixed footer at the end, if it has any at all. This is called the 'epilogue'.

Once you have created a style file that describes the file format you have or want, you must tell GPSTBabel to use the xcsv format and have the xcsv format use that file. If you created a new style file called "mystyle.style" and you want to write the waypoints from a GPX file named "mine.gpx" to it, you would issue a command like:

```
gpsbabel -i gpx -f mine.gpx -o xcsv,style=mystyle.style -F mine.new
```

You might then examine mine.new to see if it met your expectations. If not, you could continue to tweak mystyle.style until it did, rerunning the above command each time. If 'mystyle' is a format that describes a popular program or is likely to be of use to others, you can then share mystyle.style with other GPSTBabel users. Send it along with a coherent description to the GPSTBabel-misc mailing list for consideration to be included in a future version.

## Style file overview

The first and foremost important step is understanding how the style file is laid out itself. The format is:

DIRECTIVE<whitespace>VALUE

Where <whitespace> is one or more spaces or tabs. There should be no spaces or tabs at the beginning of the line; all directives start at the left edge in column zero.

An example style format is shown here:

```
# Format: MS S&T 2002/2003
# Author: Alex Mottram
```

```
# Date: 12/09/2002
#

DESCRIPTION  Microsoft Streets and Trips 2002-2006
EXTENSION    txt

#
# FILE LAYOUT DEFINITIONS:
#
FIELD_DELIMITER TAB
RECORD_DELIMITER NEWLINE
BADCHARS  ,

PROLOGUE Name Latitude Longitude Description URL Type Container Diff Terr

#
# INDIVIDUAL DATA FIELDS, IN ORDER OF APPEARANCE:
# NOTE: MS S&T ONLY IMPORTS DATA, IT DOESN'T
#   EXPORT THIS ANYWHERE SO WE CAN HAVE OUR
#   WAY WITH THE FORMATTING.
#
IFIELD SHORTNAME, "", "%s" # Name
IFIELD LAT_DECIMAL, "", "%f" # Latitude
IFIELD LON_DECIMAL, "", "%f" # Longitude
IFIELD DESCRIPTION, "", "%s" # Name 2 (Big Description)
IFIELD URL, "", "%s" # URL
IFIELD GEOCACHE_TYPE, "", "%s" # Geocache Type
IFIELD GEOCACHE_CONTAINER, "", "%s" # Geocache Type
IFIELD GEOCACHE_DIFF, "", "%3.1f" # Geocache Type
IFIELD GEOCACHE_TERR, "", "%3.1f" # Geocache Type
```

Each of these lines will be explained in the following sections.

## Internal Constants

A few internal constants are defined in the XCSV parser to make the style file simpler. They may or may not be used and are optional in most cases. Note that only certain style file directives map these constants.

**Style Constant:** COMMA  
**Maps to Char(s):** ,  
**Style Constant:** COMMASPACE  
**Maps to Char(s):** ,<space>  
**Style Constant:** SINGLEQUOTE  
**Maps to Char(s):** '  
**Style Constant:** DOUBLEQUOTE  
**Maps to Char(s):** "  
**Style Constant:** COLON  
**Maps to Char(s):** :  
**Style Constant:** SEMICOLON  
**Maps to Char(s):** ;  
**Style Constant:** NEWLINE  
**Maps to Char(s):** \n  
**Style Constant:** CR  
**Maps to Char(s):** \r

**Style Constant:** CRNEWLINE  
**Maps to Char(s):** \r\n  
**Style Constant:** TAB  
**Maps to Char(s):** \t  
**Style Constant:** SPACE  
**Maps to Char(s):** <space>  
**Style Constant:** HASH  
**Maps to Char(s):** #  
**Style Constant:** PIPE  
**Maps to Char(s):** |  
**Style Constant:** WHITESPACE  
**Maps to Char(s):** see below

## WHITESPACE

The WHITESPACE constant has special properties. When reading data, WHITESPACE refers to sequential runs of SPACES and/or TABS. When writing data, WHITESPACE is always a single SPACE.

For example, the following line:

```
SOME_NAME          30.1208 -91.1365      SOME OTHER NAME
```

Parses into the following data fields:

```
SOME_NAME , 30.1208 , -91.1365 , SOME , OTHER , NAME
```

## COMMENTS

Anything after a hash (#) on a line is not parsed. For example:

```
#THIS ENTIRE LINE IS A COMMENT.  
#FIELD LAT_DECIMAL, " ", "%f"    THIS ENTIRE LINE IS A COMMENT  
FIELD LAT_DECIMAL, " ", "%f"    # ONLY THIS SENTENCE IS A COMMENT.
```

## Global Properties of the File

There are a few available directives to describe general traits of the file being described and not specific data within the file itself.

## DESCRIPTION

This is the description of the file format being described. This text appears in the help screens and in menus used by the various GUI wrappers.

## EXTENSION

This directive gives the filename extension generally associated with this file.

## ENCODING

Describes the character set used by this format. The value given must be one listed by 'gpsbabel -l'.  
example:

ENCODING UTF-8 # Use UTF-8 for input and output.

## DATUM

This value specifies the GPS datum to be used on read or write. Valid values for this option are listed in Appendix A, *Supported Datums*.

DATUM European 1950

## DATATYPE

Specifies the kind of data we have to read or write.

By default all data are seen as waypoint data. With DATATYPE you are now able to bind a specific type to this format. Possible values are WAYPOINT, ROUTE or TRACK.

DATATYPE ROUTE # route-only format

## GPSBabel Behavior Directives

There are a few available directives to control some of the internal processing functions of GPSbabel.

## SHORTLEN

This sets the maximum allowed shortname length when using the internal shortname synthesizer.

example:

```
SHORTLEN 16 # shortnames will be at most 16 characters long.
```

## SHORTWHITE

This tells the shortname synthesizer whether or not to allow whitespace in the synthesized shortnames. Allowed values are zero and one.

example:

```
SHORTWHITE 0 # Do not allow whitespace in shortname.  
SHORTWHITE 1 # Allow whitespace in shortname.
```

## Defining the Layout of the File

The first few directives define the layout the physical file itself:

## FIELD\_DELIMITER

The field delimiter defines the character(s) that separate the fields in the rows of data inside the XCSV file. Common field delimiters are commas and tabs. (referred to as "comma separated values" and "tab separated values")

examples:

FIELD_DELIMITER	COMMA
FIELD_DELIMITER	~

The directive FIELD\_DELIMITER is parsed for STYLE CONSTANTS as defined in the table above.

## RECORD\_DELIMITER

The record delimiter defines that character(s) that separate ROWS of data (FIELDS) in the XCSV file. The most common record delimiters are NEWLINE and CR (carriage return).

examples:

RECORD_DELIMITER	NEWLINE
RECORD_DELIMITER	

The directive RECORD\_DELIMITER is parsed for STYLE CONSTANTS as defined in the table above.

## BADCHARS

Bad characters are things that should *\*never\** be written into the XCSV file as data on output. GPSBabel automatically includes any non-blank FIELD\_DELIMITER and RECORD\_DELIMITER characters as BADCHARS by default.

examples:

BADCHARS	COMMA
BADCHARS	~

The directive BADCHARS is parsed for STYLE CONSTANTS as defined in the table above.

## PROLOGUE

A prologue is basically constant data that is written to the output file BEFORE any waypoints are processed. PROLOGUE can be defined multiple times in the style file, once for each "line" before the data begins. This is commonly used in XCSV files as a "header" row.

examples:

```
PROLOGUE OziExplorer Waypoint File Version 1.1
PROLOGUE WGS 84
PROLOGUE Symbol,Name,Latitude,Longitude
```

## EPILOGUE

An Epilogue is the same as a prologue, except this data is written at the END of the file. See the examples for PROLOGUE above.

## Defining Fields Within the File

A field defines data. There are two different classifications of FIELDS, IFIELD (file input) and OFIELD (file output). In the absence of any OFIELDS, IFIELDS are use as both input and output. The existence of OFIELDS is primarily to allow more flexible mapping of GPSBabel data to output data (say, for instance,

to map the internal GPSBabel "description" variable to two or more fields on output). For all practical purposes, IFIELDS and OFIELDS are defined the same way in the style file.

The following per-field options are defined:

- "no\_delim\_before" is supported on in OFIELD tags to specify that this field should be written without a field delimiter before it. It's useful for limited field concatenation.
- "absolute" is supported on OFIELD tags for lat and lon to indicate that only absolute values (never negative) are to be printed.
- "optional" is supported only OFIELD tags and indicates that the field may or may not be available in the source data. If the field is absent, no trailing field separator is written.

This attribute is most useful when paired with "no\_delim\_before" as it allows you to concatenate fields without concern for whether those fields are actually populated or not.

There are several different types of fields that may be defined. Each field consists of three pieces of information: the FIELD TYPE, a DEFAULT VALUE, and a PRINTF CONVERSION (for output). In many cases, not all pieces are used, but all 3 pieces are required. Additionally, an fourth field is supported that modifies the behaviour of the field being described.

FIELDS should be defined in the style file in the logical order that they appear in the data, from left to right. This is the order in which they are parsed from input and written to output.

The fields used by the XCSV parser are as follows:

## IGNORE

IGNORE fields are, guess what, ignored on input. Internally, IGNORE fields are treated as CHARACTER data, and as such, require a printf conversion for a character array.

examples:

```
IFIELD IGNORE, "", "%14.14s"    # (writes a 14 character blank field)
IFIELD IGNORE, "", "%s"        # (writes a blank field on output)
```

## CONSTANT

CONSTANT fields are, of course, constant. They are ignored on input, however they write CONSTANT data on output. As such, they require a DEFAULT VALUE and a printf conversion for a character array.

examples:

```
IFIELD CONSTANT, "FFFFFF", "%s"    # (writes "FFFFFF" in the field)
IFIELD CONSTANT, "01/01/70", "%s"  # (a constant date field)
```

## INDEX

An INDEX field is used ONLY on output. The INDEX constant defines a field that, at output, contains the sequence number of the waypoint being written, starting at 0. An index is managed internally as an INTEGER and requires an INTEGER printf conversion. An INDEX has one special property. The DEFAULT VALUE of the index is added to the index on each iteration (to allow indexes starting at 1, 100, etc..).

examples:

```
IFIELD INDEX,"0","%04d"    # (Starts counting at zero)
IFIELD INDEX,"","%04d"     # (Starts counting at zero)
IFIELD INDEX,"1","%04d"    # (Starts counting at one)
```

## SHORTNAME

A SHORTNAME is generally the waypoint name of the data being processed. SHORTNAME maps directly to the GPSBabel variable ->shortname. A SHORTNAME is CHARACTER data and requires a character array printf conversion.

example:

```
IFIELD SHORTNAME,"","%s"
```

## DESCRIPTION

A DESCRIPTION is generally a long description of the waypoint. A DESCRIPTION maps to the GPSBabel variable ->description and is otherwise handled exactly like a SHORTNAME.

examples:

```
IFIELD DESCRIPTION,"","%s"
```

## NOTES

NOTES are generally everything else about a waypoints. NOTES map to the GPSBabel variable ->notes and is otherwise handled exactly like a SHORTNAME.

## URL

URL is a URL for the waypoint. URL maps to the GPSBabel variable ->url and is otherwise handled exactly like a SHORTNAME.

example:

```
IFIELD URL,"","%s"
```

## URL\_LINK\_TEXT

URL\_LINK\_TEXT is a textual description of where a URL points. URL\_LINK\_TEXT maps to the GPSBabel variable ->url\_link\_text and is otherwise handled exactly like a SHORTNAME.

example:

```
IFIELD URL_LINK_TEXT,"","%s"
```

## ICON\_DESCR

ICON\_DESCR is a textual description of an icon type for a waypoint. ICON\_DESCR maps to the GPSBabel variable ->icon\_desc and is otherwise handled exactly like a SHORTNAME.

example:

```
IFIELD ICON_DESCR, " ", "%s"
```

## LAT\_DECIMAL

LAT\_DECIMAL defines LATITUDE in DECIMAL format. Note that this is a PURE signed decimal format (i.e. -91.0000). This data is handled internally as a DOUBLE PRECISION FLOAT and requires a FLOATING POINT printf conversion.

example:

```
IFIELD LAT_DECIMAL, " ", "%f"
```

## LON\_DECIMAL

See LAT\_DECIMAL, except LON\_DECIMAL defines LONGITUDE.

## LAT\_INT32DEG

LAT\_INT32DEG defines LATITUDE in what I call INT32DEGREES. This value is a signed LONG INTEGER and requires a LONG INTEGER printf conversion. (This format is only used by some DeLorme products.)

example:

```
IFIELD LAT_INT32DEG, " ", "%ld"
```

## LON\_INT32DEG

See LON\_INT32DEG except LON\_INT32DEG defines LONGITUDE.

## LAT\_DECIMALDIR / LAT\_DIRDECIMAL

LAT\_DECIMALDIR and LAT\_DIRDECIMAL define LATITUDE in DECIMAL format with the added bonus of a 'N/S' or 'E/W' direction character. This data is handled internally as a DOUBLE PRECISION FLOAT and a single CHARACTER and requires a FLOATING POINT as well as a CHARACTER printf conversion. The only difference between the two is whether the directional character appears before (LAT\_DIRDECIMAL) or after (LAT\_DECIMALDIR) the decimal number.

examples:

```
IFIELD LAT_DECIMALDIR, " ", "%f %c"      # (writes 31.333 N)
IFIELD LAT_DIRDECIMAL, " ", "%c %f"      # (writes N 31.333)
```

## LON\_DECIMALDIR / LON\_DIRDECIMAL

Same as LAT\_DECIMALDIR / LAT\_DIRDECIMAL except LON\_ defines LONGITUDE.

## LAT\_DIR / LON\_DIR

LAT\_DIR returns the single character 'N' or 'S' depending on the hemisphere of the latitude. LON\_DIR returns 'E' or 'W' depending on the hemisphere of the longitude.



## LAT\_HUMAN\_READABLE

LAT\_HUMAN\_READABLE defines LATITUDE in a human-readable format. This format is probably the most expressive format. It is similar to LAT\_DECIMALDIR in that it requires multiple printf conversions, but it is far more flexible as to the contents of those conversions. On read, the printf conversions are ignored and GPSBabel attempts to determine the latitude and longitude based on what is in the file.

examples:

```
# (writes N 31 40.000)
IFIELD LAT_HUMAN_READABLE, "", "%c %d %f"
# (writes "31 deg 40.000 min N")
IFIELD LAT_HUMAN_READABLE, "", "%d deg %f min %c"
# Note that this string will confuse the reading routine due
# to the letter "n" in "min" and the letter "e" in "deg."
# (writes 31 40 00.000N)
IFIELD LAT_HUMAN_READABLE, "", "%d %d %f%c"
```

## MAP\_EN\_BNG

MAP\_EN\_BNG converts coordinates from/to British National Grid (BNG).

The only supported order of the items is: Map,Easting,Northing. During output all coordinates have to be located within this limited area.

examples:

```
IFIELD MAP_EN_BNG, "", "%s%5d %5d"      # (writes i.e. "SJ00001 00001")
IFIELD MAP_EN_BNG, "", "%s %d %d"       # (writes i.e. "TQ 888 999")
```

## LON\_HUMAN\_READABLE

See LAT\_HUMAN\_READABLE except LON\_HUMAN\_READABLE defines LONGITUDE.

## LATLON\_HUMAN\_READABLE

LATLON\_HUMAN\_READABLE is like LAT\_HUMAN\_READABLE and LON\_HUMAN\_READABLE except that it reads and writes both latitude and longitude as a single field. On write, the same format specifier is used for both coordinates. On read, GPSBabel does exactly the same thing it does for LAT\_HUMAN\_READABLE or LON\_HUMAN\_READABLE.

example:

```
IFIELD LATLON_HUMAN_READABLE, "", "%c %d %f"
# (writes "N 31 40.126 W 85 09.62" as a single field)
```

## LAT\_NMEA

Defines the latitude in the format used by the NMEA standard which is degrees multiplied by 100 plus decimal minutes.

example:

```
IFIELD LAT_NMEA, "%f", "%08.3f" # (writes 3558.322)
```

## LON\_NMEA

Defines the longitude in the format used by the NMEA standard which is degrees multiplied by 100 plus decimal minutes.

example:

```
IFIELD LON_NMEA, "%f", "%010.3f" # (writes -08708.082)
```

## LAT\_10EX / LON\_10EX

Defines the latitude or longitude in the format used i.e. by TomTom Navigator itinerary files. It is degrees multiplied by 10 power X. X have to be replaced with a valid decimal value. A factor of 10000 would be generated by LAT\_10E5 as shown in the examples below.

examples:

```
IFIELD LAT_10E5, "%f", "%.f" # (writes 3558322)
```

```
IFIELD LON_10E5, "%f", "%.f" # (writes -8708082)
```

## ALT\_FEET

ALT\_FEET is the position's ALTITUDE in FEET. This value is treated as a SIGNED DOUBLE PRECISION FLOAT and requires a FLOATING POINT printf conversion.

example:

```
IFIELD ALT_FEET, " ", "%.0f"
```

## ALT\_METERS

ALT\_METERS is identical to ALT\_FEET with the exception that the altitude is in METERS.

## HEART\_RATE

Heart rate, measured in beats per minute. Only valid for units with heart rate monitor features (i.e. Garmin Forerunner 301).

example:

```
IFIELD HEART_RATE, " ", "%d"
```

## CADENCE

Cadence in revolutions per minute. Only valid for units with heart rate monitor features (i.e. Garmin Edge 305).

example:

```
IFIELD CADENCE, " ", "%d"
```

## EXCEL\_TIME

EXCEL\_TIME is the waypoint's creation time, if any. This is actually the decimal days since 1/1/1900 and is handled internally as a DOUBLE PRECISION FLOAT and requires a FLOATING POINT printf conversion.

example:

```
IFIELD EXCEL_TIME, " ", "%11.5f"
```

## TIMET\_TIME

TIMET\_TIME is the waypoint's creation time, if any. This is actually the integer seconds since 1/1/1970 (let's not start the holy war) and is handled internally as a LONG INTEGER and requires a LONG INTEGER printf conversion.

example:

```
IFIELD TIMET_TIME, " ", "%ld"
```

## YYYYMMDD\_TIME

YYYYMMDD\_TIME is the waypoint's creation time, if any. It's a single decimal field containing four digits of year, two digits of month, and two digits of date. Internally it is a LONG INTEGER and thus requires a LONG INTEGER printf conversion.

example:

```
IFIELD YYYYYMDD_TIME, " ", "%ld"
```

## GMT\_TIME

GMT\_TIME is the waypoint's creation time, in UTC time zone. It uses the strptime conversion format tags.

example:

```
IFIELD GMT_TIME, " ", "%m/%d/%Y %I:%M:%D %p"
```

Search the web for 'strptime man page' for details strptime, but one such page can be found at <http://www.die.net/doc/linux/man/man3/strptime.3.html>

## LOCAL\_TIME

LOCAL\_TIME is the waypoint's creation time, in the local time zone. It uses strptime conversion format tags. See GMT\_TIME for a reference.

example:

```
IFIELD LOCAL_TIME, " ", "%Y-%m-%d"
```

## HMSG\_TIME

HMSG\_TIME parses up to three time parts and am/pm string to add this value to the previously parsed \*\_TIME field that contains only a date. On output, will print the time in UTC.

example:

```
IFIELD HMSG_TIME, " ", "%d:%d:%d %s"
```

## HMSL\_TIME

HMSG\_TIME parses up to three time parts and am/pm string to add this value to the previously parsed \*\_TIME field that contains only a date. On output, will print the time in local time.

example:

```
IFIELD HMSL_TIME, " ", "%dh%dm"
```

## ISO\_TIME

ISO\_TIME is the waypoint's creation time, in ISO 8601 format, which include time zone information. It is expected to be in the format yyyy-mm-ddThh:mm:sszzzzz where zzzzzz is the local time offset or the character Z for UTC time. On output, UTC 'Z' time zone will always be used.

example:

```
IFIELD ISO_TIME, " ", "%s"
```

## ISO\_TIME\_MS

ISO\_TIME\_MS is much like ISO\_TIME, but expresses milliseconds at the end of the timestamp. It is thus in the format yyyy-mm-ddThh:mm:ss.SSSzzzzz where 'SSS' is milliseconds and zzzzzz is the local time offset or the character Z for UTC time. On output, UTC 'Z' time zone will always be used.

example:

```
IFIELD ISO_TIME_MS, " ", "%s"
```

## GEOCACHE\_DIFF

GEOCACHE\_DIFF is valid only for geocaches and represents a DOUBLE PRECISION FLOAT. This is the geocache "difficulty" rating as defined by Groundspeak. A "three and a half star" cache would therefore be "3.5"

example:

```
IFIELD GEOCACHE_DIFF, " ", "%3.1f"
```

## GEOCACHE\_TERR

GEOCACHE\_TERR is valid only for geocaches and represents a DOUBLE PRECISION FLOAT. This is the geocache "terrain" rating as defined by Groundspeak. A "three and a half star" cache would therefore be "3.5"

example:

```
IFIELD GEOCACHE_TERR, " ", "%3.1f"
```

## GEOCACHE\_CONTAINER

GEOCACHE\_CONTAINER is valid only for geocaches and is heavily influenced by the Groundspeak container types. Examples would include "Micro" and "Virtual".

example:

```
GEOCACHE_CONTAINER, " ", "%s"
```

## GEOCACHE\_TYPE

GEOCACHE\_TYPE is valid only for geocaches and is heavily influenced by the Groundspeak cache types. Examples would include "Event cache" and "Multi-Cache".

example:

```
GEOCACHE_TYPE, " ", "%s"
```

## GEOCACHE\_PLACER

GEOCACHE\_PLACER is a string containing the name of the placer of a geocache.

example:

```
GEOCACHE_PLACER, " ", "%s"
```

## GEOCACHE\_ISAVAILABLE

GEOCACHE\_ISAVAILABLE is a string containing "True" or "False" indicating whether a geocache is currently available or not.

example:

```
GEOCACHE_ISAVAILABLE, " ", "%s"
```

## GEOCACHE\_ISARCHIVED

GEOCACHE\_ISARCHIVED is a string containing "True" or "False" indicating whether a geocache has been archived.

example:

```
GEOCACHE_ISARCHIVED, " ", "%s"
```

## GEOCACHE\_LAST\_FOUND

A long integer in format YYYYMMDD containing the last time this geocache was found.

example:

```
GEOCACHE_LAST_FOUND, " ", "%ld"
```

## GEOCACHE\_HINT

The hint for this geocache. No additional transformation (such as rot13) will be performed on this string.

example:

```
GEOCACHE_HINT, " ", "%s"
```

## **PATH\_DISTANCE\_MILES**

PATH\_DISTANCE\_MILES outputs the total length of the route or track from the start point to the current point, in miles. This and the altitude could be used to create an elevation profile. PATH\_DISTANCE\_MILES is a DOUBLE PRECISION FLOAT.

PATH\_DISTANCE\_MILES is not valid as an input field.

PATH\_DISTANCE\_MILES is only meaningful if the data comes from a track or a route; waypoint data will generate essentially meaningless output.

example:

```
PATH_DISTANCE_MILES, " ", "%f"
```

## **PATH\_DISTANCE\_KM**

PATH\_DISTANCE\_KM is like PATH\_DISTANCE\_MILES except it outputs the length in kilometers.

## **PATH\_SPEED**

Speed in meters per second. Gpsbabel does NOT calculate this data by default; it is read from the input file if present. (If not present, it may be calculated with the track filter.)

example:

```
PATH_SPEED, " ", "%f"
```

## **PATH\_SPEED\_KPH**

Like PATH\_SPEED but means kilometers per hour.

example:

```
PATH_SPEED_KPH, " ", "%.1f"
```

## **PATH\_SPEED\_MPH**

Like PATH\_SPEED but means miles per hour.

example:

```
PATH_SPEED_MPH, " ", "%.1f"
```

## **PATH\_SPEED\_KNOTS**

Like PATH\_SPEED but means knots (nautical).

example:

```
PATH_SPEED_KNOTS, " ", "%.1f"
```

## PATH\_COURSE

Course in degrees. Gpsbabel does not calculate this data by default; it is read from the input file if present. (If not present, it may be calculated with the track filter.)

example:

```
PATH_COURSE, " ", "%f "
```

## GPS\_HDOP / GPS\_VDOP / GPS\_PDOP

GPS horizontal / vertical / positional dilution of precision parameters. Needs float conversion.

example:

```
GPS_HDOP, " ", "%f "
```

## GPS\_SAT

Number of satellites used for determination of the position. Needs integer conversion.

example:

```
GPS_SAT, " ", "%d "
```

## GPS\_FIX

Type of fix (see GPX spec or track filter). Needs string conversion.

example:

```
GPS_FIX, " ", "%s "
```

## TRACK\_NAME

The name of the track currently being operated on. Needs string conversion.

example:

```
TRACK_NAME, " ", "%s "
```

## ROUTE\_NAME

The name of the route currently being operated on. Needs string conversion.

example:

```
ROUTE_NAME, " ", "%s "
```

## STREET\_NAME

Street address including house number. Notice that this is not used for any geocoding, it's merely textual description associated with a position.

example:

```
STREET_ADDR, " ", "%s"
```

## CITY

The name of a city. Sometimes part of "Points of Interest". This is simple textual data associated with a position, no geocoding will be done..

example:

```
CITY, " ", "%s"
```

## COUNTRY

The name of a country associated with a position.

example:

```
COUNTRY, " ", "%s"
```

## FACILITY

The name of a facility to associate with a position.

example:

```
FACILITY, " ", "%s"
```

## PHONE\_NR

A phone number associated with a position. This is just textual data attached for convenience.

example:

```
PHONE_NR, " ", "%s"
```

## POSTAL\_CODE

A postal code to associate with a position. It is freeform text and is not used by GPSBabel for any geocoding or such.

example:

```
POSTAL_CODE, " ", "%s"
```

## Examples

Here is one example style file from the GPSBabel source.

```
# gpsbabel XCSV style file  
#
```



```
# Format: Garmin POI
# Author: Robert Lipe
# Date: 10/07/2005
# Reference: http://forums.groundspeak.com/GC/index.php?showtopic=110641&st=0&#entry1752204
#
DESCRIPTION Garmin POI database
#
#
# FILE LAYOUT DEFINITIONS:
#
FIELD_DELIMITER COMMA
RECORD_DELIMITER NEWLINE
BADCHARS COMMA
SHORTLEN 24

#
# INDIVIDUAL DATA FIELDS, IN ORDER OF APPEARANCE:
#
IFIELD LON_HUMAN_READABLE, "", "%08.5f"
IFIELD LAT_HUMAN_READABLE, "", "%08.5f"
IFIELD SHORTNAME, "", "%s"
IFIELD DESCRIPTION, "", "%s"

OFIELD LON_DECIMAL, "", "%08.5f"
OFIELD LAT_DECIMAL, "", "%08.5f"
OFIELD SHORTNAME, "", "%-.24s"
OFIELD GEOCACHE_TYPE, "", "%-.4s", "no_delim_before,optional"
OFIELD GEOCACHE_CONTAINER, "", "%-.4s", "no_delim_before,optional"
OFIELD GEOCACHE_DIFF, "", "(%3.1f", "no_delim_before,optional"
OFIELD GEOCACHE_TERR, "", "(%3.1f)", "no_delim_before,optional"
OFIELD DESCRIPTION, "", "%-.50s"
```

When used on a Groundspeak Pocket Query, it will output lines that look like:

```
-76.76234,38.39123,GC5370 Loca/Virt (1.0/1.0),Dude.. Wheres my Limo??
-90.42345,38.55234,GCC8B Trad/Regu (2.0/2.0),Sweet Reward
-90.81456,38.62456,GC3091 Trad/Regu (1.5/2.0),Matson Hill
```

that are suitable for Garmin's POI loader.

For additional examples, please see the \*.style files in the style/ subdirectory of GPSBabel or at the online source tree [<http://gpsbabel.cvs.sourceforge.net/gpsbabel/gpsbabel/style/>].

## Miscellaneous Notes

### Default Values

Default values are supported for any output fields that contain pure character data output such as URL and NOTES. Default values are only written on output and are not used to supplement missing input. When using default values your mileage will vary greatly depending on the input formats used to populate waypoint data.

---

# Glossary

Terms that are used in conjunction with GPSBabel.

## G

Geocaching                                      GPS based "paper chase", see <http://en.wikipedia.org/wiki/Geocaching>

## I

Itinerary                                        same as a Route (e.g. used by TomTom)

## P

Points of Interest (POI)                    a collection of gas stations, post boxes, shops and like.

## R

Route    a list of geopoints (often with names) connected in a specific order. Usually a collection of geopoints defining the route you want to pass while traveling, created by PC software, or generated inside a GPS device. They can be composed of existing waypoints, or new "routepoints" might be generated.

## T

Track    a collection of geopoints recorded by your GPS device while traveling -- "breadcrumb trails". The order of trackpoints within the track is important. Usually a trackpoint doesn't have a name or comment, but a timestamp. This distinguishes a trackpoint from a waypoint.

## W

Waypoints                                      are geopoints that are not necessarily connected to other points, and their order is unimportant. They can be entered before, while or after you actually visit the place and might have tags like name, comment and the like. Usually used to mark special locations as your home, a hotel or a geocache.