GPS Manager User Manual

Miguel Filgueiras

DCC-FC & LIACC, Universidade do Porto
R. do Campo Alegre 823, 4150-180 Porto, Portugal
Phone: +351-226078830 Fax: +351-226003654
email: mig@ncc.up.pt

16 July 2004

Contents

Chapter 1

Introduction

GPS Manager (GPSMan) is a graphical manager of GPS data that makes possible the preparation, inspection and edition of GPS data in a friendly environment. GPSMan supports communication and real-time logging with Garmin, Lowrance and Magellan receivers and accepts real-time logging information in NMEA 0183 from any GPS receiver. GPSMan can also be used in command mode (with no graphical interface).

GPSMan is a stand-alone Tcl/Tk program. Its use in real-time is at the sole risk of the user.

 See Appendix D for a list of new features.

1.1 Contributors

GPSMan incorporates code contributed by

- Brian Baulch (baulchb@hotkey.net.au): communication with Lowrance receivers, support for the wheelmouse, real-time logging (variant for the Lowrance);
- Rogério Reis (Universidade do Porto): Debian Linux package and the utilities to configure and lock the serial port;
- Andreas Lange (Andreas.C.Lange@GMX.de): support for German (versions below 6.0);
- Alessandro Palmas (alpalmas@tin.it): implementation of elevation graphs for tracks and routes, both in 2 and 3 dimensions, support for Italian, and exportation of data in OziExplorer format;
- Niki Hammler (http://www.nobaq.net): Perl script for reading waypoint data in Fugawi export format which was translated into Tcl and incorporated in GPSMan;
- Martin Ostermann (Aachen University of Technology): conversion of waypoints listed in HTML pages of the MapBlast site into GPSMan data;

- Valère Robin (valere.robin@wanadoo.fr): support for French, importation of EasyGPS export and GPX formats, and exportation of GPX format;
- David Wolfskill (david@catwhisker.org): FreeBSD package;
- Rob Buitenhuis (rob@buitenhs.demon.nl): support for Dutch;
- Frank Kujawski (Frank@Kujawski.org): conversion of routes listed in HTML pages of the MapsOnUS site into GPSMan data;
- Tri Agus Prayitno (acuss@bk.or.id): support for Indonesian;
- Matt Martin (matt.martin@ieee.org): communication with Magellan receivers;
- Stefan Heinen (Stefan.Heinen@synopsys.com): new data structures for datums, the procedure to access them and changes to improve the focus policy and bindings under MS-Windows;
- Heiko Thede (Heiko.Thede@gmx.de): shell- and Tcl-scripts that convert export files from Map&Guide and related software (Falk, Power Route) to files in GPSMan format and which were incorporated in GPSMan.
- Sabine Broda (sbb@ncc.up.pt): support for German (since version 6.0).
- Alberto Morales (amd77@gulic.org): support for Spanish;
- Martin Buck (m@rtin-buck.de): resizing of 2D graphs, change in track edit window;
- David Kaplan (dmkaplan@ucdavis.edu): RPM packages;
- Jean H. Theoret (ve2za@rac.ca): code for changing the symbol of each waypoint in a group;
- Paul Scorer (p.scorer@leedsmet.ac.uk): a Tcl-script implementing importation of British Gliding Association turnpoint (DOS) files.

Kind help from many other people should be mentioned and is acknowledged below (6.3).

1.2 Main features

What GPSMan does when in graphical mode:

- GPSMan keeps lists of data items (waypoints, routes, tracks, polylines and groups) whose information can be written to and read from text files, or, except for polylines, got from and put into supported GPS receivers; ▌↓*new* ▌*new*↑
- GPSMan lets the user create new data items, as well as modify or delete those already defined; groups (sets) of data items are very helpful in keeping and classifying the existing information, as well as in the selection of information to be processed;
- GPSMan makes conversions of
  - routes into tracks;
  - tracks into routes, tracks or polylines, by keeping a certain number (fixed by the user) of track points; ▌↓*new* ▌*new*↑
  - polylines into tracks; ▌↓*new* ▌*new*↑

  - tracks into a waypoint taking the averages of the latitudes, longitudes and altitudes of the track points;
  - waypoints in a group into a waypoint taking the averages of their latitudes, longitudes and altitudes;

- GPSMan records real-time track logging information that can be displayed on a moving map and used to create a track (that may be then converted into a route or a polyline); ↓*new* *new*↑
- GPSMan makes computations of,
  - for waypoints: distance and bearing to another waypoint, nearest waypoints (in fact, distances and bearings to all other waypoints ordered from nearest to furthest), clusters of waypoints with given centres,
  - for routes: distances, azimuths and differences in altitude between consecutive points, total distance, and enclosed area (under certain conditions),
  - for tracks: distances, differences in time, speed and azimuths between consecutive points, cumulative distance and altitude at each point, total distance and average speed for the track, distance from first to last point, maximum distance from first point to any track point;

- GPSMan can make a map to scale, using one of a choice of projections, showing waypoints, routes, tracks and polylines; the map can be saved or printed in Postscript or other graphics formats if the `Img` Tcl/Tk library is available; images may be used as background for the map and geo-referenced; waypoints can be represented in different ways (any combination of symbol and name or comment); an animation of the movement along the real-time track or of a recorded track can be shown on the map; elevation graphs, as side-views or perspectives, of routes and tracks can be plotted and saved or printed in Postscript or other formats if `Img` is available; speed and climb rate graphs for tracks can be plotted, saved and printed in a similar way; ↓*new* *new*↑
- GPSMan allows for data items to be searched by:
  - patterns matching the item name, comment and/or remark,
  - distance to a given waypoint or location (given by its coordinates), for waypoints and tracks,
  - symbol, for waypoints,
  - waypoints, for routes,
  - start date, for tracks.

- GPSMan provides conversion between different position formats (latitude/longitude in DMS, DMM or DDD, and several grid coordinates, including UTM/UPS) and/or different datums; there is support for user-defined datums, ellipsoids, projections and related coordinate grids.
- GPSMan allows the user to change its configuration, providing a choice of languages (Dutch, English, French, German, Indonesian, Italian, Spanish and Portuguese in the current distribution), and accepting new values for parameters related to the GPS receiver, default settings, and concerning interface appearance (colours, dimensions, positions). ↓*new* *new*↑

What GPSMan does when in command-line mode:

- GPSMan makes availability and connection checks.
- GPSMan provides information on its version, the supported protocols, data file formats, projections, coordinates transformations, datums, and available commands. ↓*new* *new*↑

- GPSMan connects to the receiver, gets data or the real-time log and stores them in a file.
- GPSMan reads data from a file in a user selected format, connects to the receiver and transfers the loaded information to it.
- GPSMan reads data from a file in a user selected format and writes it to another file in another selected format. ▌$\downarrow new$
- GPSMan prints the projected coordinates of a point using a given projection.
- GPSMan produces an image information file from geo-referencing data.
- GPSMan executes Tcl/Tk scripts and scripts made up of GPSMan commands. ▌$new\uparrow$

Chapter 2

Programs

2.1 Current version

The current version is: GPSMan version 6.1.2, a stand-alone Tcl/Tk program that communicates directly with the GPS receiver.

2.2 Known problems

The list of known problems (on 16 July 2004) is:

1. GPSMan relies on the use of a serial port to communicate with the GPS receiver. Some Tcl/Tk installations (e.g., in SuSe, Red Hat and Mandrake Linux distributions), operating system drivers and even hardware (in some laptops) were reported not to work correctly with the serial port. Namely, the Tcl error: "bad option -mode" is a problem of bad configuration of the Tcl/Tk installation and upgrading to a newer version normally solves the problem. To help debugging input from a serial port a Tcl/Tk program is available from the WWW page at `http://www.ncc.up.pt/gpsman`. It must be edited before use to set the correct path to the serial port.
2. GPSMan support for Garmin receivers may need to convert between bytes and floating point numbers. Tcl/Tk has no machine-independent way to do these conversions and GPSMan only implements them for little- or big-endian architectures that follow the IEEE floating point standard. Some Garmin receivers do not use protocols having floating point numbers and are not affected by this.
3. GPSMan releases 5.3 and later cannot be run under Tcl/Tk 8.0 due to bugs in the latter. The recommended version of Tcl/Tk is 8.4.
4. Tcl/Tk 8.4 conflicts with the `Img` library versions below 1.3, making it impossible to load/save images in formats other than GIF and PNM.
5. due to limitations imposed by Tcl/Tk, that does not implement communication with serial ports on some Macintosh platforms, GPSMan will not be able to dialog with the GPS receiver on these systems. See below (2.6.4) for information on how to use a USB port under MacOS X. ↓*new* *new*↑

2.3 Downloading GPSMan

GPSMan (version 6.1.2) is available for downloading from `http://www.ncc.up.pt/gpsman`. Both Debian, RPM and FreeBSD packages are available for easy installation in Linux. Updates for the current release can also be found there.

In order to run GPSMan, Tcl/Tk (preferably version 8.4) must be previously installed. It can be got from the Active State site at `http://tcl.activestate.com`.

The following Tcl/Tk libraries will be used if they are installed:

- `Img`, a library implementing enhanced support for graphics and graphic formats; it is part of the latest Tcl/Tk distributions and is provided by the Debian Linux package `libtk-img`; there is an `Img` site at `http://www.xs4all.nl/~nijtmans/img.html`;
- `gpsmanshp`, a library that provides the means for creating and reading files in the ESRI Shapefile format; there is a `gpsmanshp` site at `http://www.ncc.up.pt/gpsmanshp`. Version 1.2 or newer should be installed. ▌↓*new* ▌*new*↑

2.4 Separate Utilities

The GPSMan distribution includes (in the `util` directory) some other utilities that can also be downloaded from the GPSMan site.

The first two below were not integrated into GPSMan because the format of the HTML pages they read does not follow any known specification and may change at any time. They must be edited for configuration and GPSMan must be installed before they can be used.

2.4.1 MapBlast waypoints

`mb2gmn.tcl` converts the waypoints listed in HTML pages of the MapBlast site into GPSMan files. At its core is code contributed by Martin Ostermann (Aachen University of Technology). It translates HTML pages produced at the MapBlast site in answer to queries under the `Directions` section; the pages should be saved locally and then opened from the program.

2.4.2 MapsOnUS routes

`mou2gmn.tcl` converts the waypoints listed in HTML pages of the MapsOnUS site into GPSMan files. At its core is code contributed by Frank Kujawski (Frank@Kujawski.org). It translates HTML pages created at the MapsOnUS site as follows: go to the "General Options" (under the "Tools" menu), select "Show Latitude & Longitude", plan a route, "jump to turn-by-turn directions", hit the "non-tabular format" link and save locally the HTML page. These files are then opened from the program and the resulting files will contain the routes and their waypoints. The remark fields of the waypoints will have the directions in the HTML pages.

2.4.3 BGA (DOS) turnpoints

▌↓*new*

`dos2gpsman.tcl` converts BGA (British Gliding Association) DOS turnpoints files to GPSMan. It was contributed by Paul Scorer (p.scorer@leedsmet.ac.uk). It reads from and writes to the standard input and standard output unless otherwise specified by the arguments which are:

```
[-p feature] [-f findability] [-a air_activity] \
         [-i in_file] [-o out_file] [-h] [--help]
```

where `findability` is a letter in A to D or G, and `air_activity` is a 1 or 2-characters string.

2.4.4 Splitting a Shapefile into Quadrants

`shape2quadr.tcl` reads a Shapefile and from a list of quadrangles (defined by extreme latitude and longitude) produces GPSMan files one for the set of items in the file that belong to each quadrangle. Quadrangles may overlap. Items may belong to more than one quadrangle. This utility may prove helpful for dealing with large Shapefiles. It can be called in two ways:

- giving a name and the extreme coordinates of each quadrangle, the arguments in call being

  ```
  FILE PREFIX [NAME LONGMIN LONGMAX LATMIN LATMAX]+
  ```

  where `NAME` is the name of each quadrangle

- defining a grid of quadrangles, the arguments being

  ```
  FILE PREFIX -d QPREFIX LONG LONGRANGE STEPLONG MAXLONG \
                         LAT LATRANGE STEPLAT MAXLAT
  ```

  where `QPREFIX` is used in generating names for the quadrangles, each quadrangle having an extent of `LONGRANGE` by `LATRANGE`, the first has the southwest corner at `LONG,LAT`, and the southwest corners of the others follow at steps of `LONGSTEP` and `LATSTEP` and will not be greater than `MAXLONG` and `MAXLAT`

the common arguments being: `FILE`, the Shapefile to be read, and `PREFIX` that is used for generating a file name for each quadrangle.

Shapefiles with points are not supported and altitudes are not kept. The coordinates in the Shapefile are assumed to be in decimal degrees (DDD) in the "WGS 84" datum (it can easily be changed in the utility source).

All items read are saved as GPSMan polylines (LNs) in DDD format. Each quadrangle file will also have a group (GR) with all its polylines, so that it will be easy to display/hide all of them at once; the file starts with a comment giving the boundaries of the quadrangle. The file name is of the form `PREFIX_QUAD` where `QUAD` is the quadrangle name; each file is created in the current directory destroying any file under the same name. ▌*new*↑

2.5 Data and examples

The following data and examples are available from the GPSMan site:

- the *Waypoints in Portugal* data set,
- data concerning some paragliding sites in Portugal,
- track file for one sample flight,
- computation results for that track, and,
- a map of the same track, in GIF or Postscript.

2.6 Installation

In Unix and Linux systems access to the serial port is restricted. This means that either there is a program to lock the port with super-user privileges, or the permissions of the port are changed to give read/write access to a group of users or to all users. The latter is dangerous in that it creates security problems. In any case super-user privileges are needed to install the software or to give access to the serial port group.

2.6.1 Debian and other Linux distribution packages

The installation is done as with other such packages and there is no need for manual configuration. Users of GPSMan must belong to the group that owns the serial port to be used (normally `dialout`).

2.6.2 Other Unix and Linux systems

After unpacking the files (use `tar xzvf gpsman-6.1.2.tgz`) the GPSMan main file, `gpsman.tcl`, should be edited for configuration (see below (3)) and put where it can be executed. The other GPSMan files should go into the directory whose path is given at the beginning of `gpsman.tcl`.

The package includes a file `gpsman.sh` in the `util` directory that is a shell script to call `gpsman.tcl` in graphical mode with no need to give the serial port as argument. This file should be edited for configuration and placed where its can be executed.

2.6.3 Launching the program in a Unix/Linux system

The program is launched by calling `gpsman` (or `gpsman.tcl`) that may have no arguments or a single argument with the path to the serial port device, in which case the graphical interface will be used, or 2 or more arguments, in which case the command-line mode will be entered. An exception to this is the call ▌↓*new*

```
gpsman [OPTIONS] start travel [INTERVAL]
```

that is parsed as if in command-linde mode, but then launches the graphical interface (this will only work with Garmin receivers). ▌*new*↑

2.6.4 MacOS X systems

GPSMan can be run under MacOS X systems by installing a suitable Tcl/Tk package and by using a USB to serial adapter. The following configuration is known to work (information kindly supplied by Mathias Herberts, Mathias.Herberts@iroise.net):

- Keyspan USB to Serial Adapter (USA19HS), (`http://www.keyspan.com/products/usb/usa19hs`),
- Tcl/Tk Aqua, (`http://tcltkaqua.sourceforge.net`),
- device to use: `/dev/cu.USA19...`.

To install GPSMan, unpack the `gpsman-6.1.2.tgz` archive, and edit the GPSMan main file, `gpsman.tcl` for configuration (see below (3)) and put where it can be executed. The other GPSMan files should go into the directory whose path is given at the beginning of `gpsman.tcl`.

2.6.5 Other systems

After unpacking the `gpsman-6.1.2.zip` archive the GPSMan main file, `gpsman.tcl`, should be edited for configuration (see below (3)) and put where it can be executed. The other GPSMan files should go into the directory whose path is given at the beginning of `gpsman.tcl`.

Chapter 3

Configuration

A lot of parameters may be configured in the GPSMan main file `gpsman.tcl`. Most of them are also defined in the preferences file, and the values there will overwrite those in `gpsman.tcl`.

GPSMan needs a user directory to keep both the preferences file and other files for user definitions (like user-defined projections). The path to this directory and the name of the preferences file are given at the beginning of `gpsman.tcl`. When GPSMan is launched and does not find the user directory, it either attempts to create it, or (in non-Unix systems) asks for it to be created and leaves. If the preferences file does not exist, it forces it to be created.

Users wanting to load their own Tcl/Tk code (at their own risk!) can do so by putting it in a file named `patch.tcl` in the GPSMan user directory. This file will be loaded immediately after all the GPSMan source files.

If GPSMan has been installed from the Debian or other Linux/Unix distribution packages no changes are mandatory. Otherwise, on Unix systems the information on the program source files directory, user directory and default preferences file must be correctly set. A default serial port device can be defined and will be used if no argument is passed to the main program.

On other systems the same applies to the information on the serial port.

The following list gives a description of all the options that can be configured:

- for non-Unix systems: serial device to which the receiver will be connected; users of GPSMan must have read/write permission.
- path to directory containing the program source files.
- path to user GPSMan directory that will contain the preferences file and other files for user definitions (like user-defined projections); this directory is normally not to be used explicitly by the user.
- name of the preferences file; the user directory is searched for it only if there is not a file under the same name in the current directory.
- print command or any command for further processing of Postscript files generated by GPSMan; if no such command is wanted or available this option should be set to the empty string; GPSMan will use the file `print.tmp` under the user GPSMan directory as a temporary file.

- the language to be used by GPSMan; new languages can be included by translating the `lang*.tcl` files that contain the text and messages in Dutch, English, French, German, Indonesian, Italian, Spanish and Portuguese (help here will be acknowledged) and inserting new abbreviations for month names in the `ALLMONTH` array.
- use of character composition (accents, cedilla) using Western European (isolatin1) mode, and of `Delete` key to delete last character.
- choice of main window: there are three permanent windows for the map, lists, and receiver connection; either the map or the lists window can be selected as being the main window.
- GPS receiver dependent values: GPS brand, baud rate of serial communication, default receiver protocol (only for Garmin receivers), whether all characters should be accepted in names and comments, length of names, comments, maximum numbers of waypoints, routes, waypoints in routes, and track points, use of creation dates and of lowercase letters in strings. In the distribution, the values are set for use with a Garmin GPS II. ↓*new* *new*↑
- (for Lowrance receivers only) sampling interval, in seconds, when acquiring tracks.
- default symbol and default display option to use with waypoints; correct names for symbols and display options can be found in file `symbols.tcl`.
- default line widths for representing routes, tracks and polylines on the map. ↓*new*
- when displaying a track, count of track points before showing point number; 0 means no numbers, 1 means all points numbered, 2 every other point numbered, and so on. *new*↑
- whether polylines on the map should react to mouse events; they should not if they are considered as background information. ↓*new* *new*↑
- behaviour when reading a data item with the same name as another item of the same type in the data-base: either overwrite the existing one, or create under a new name.
- behaviour when a data item with hidden information is changed: remove the hidden information, keep it, or ask the user.
- distance unit to be used. ↓*new*
- altitude unit to be used in data items; this option has no effect in altitude values displayed in real-time log or navigation windows of the Garmin variant. *new*↑
- format of positions, default datum and time offset, date format. ↓*new*
- default map projection and cursor position format when starting with an empty map. *new*↑
- accurate formulae (slower than the normal ones) for computing distances and bearings.
- whether to ask for confirmation of projection parameters.
- whether to use a window to control slow operations, and help balloons. ↓*new*
- MapGuide text format default version. *new*↑
- map dimensions, length of line for displaying a scale, and initial map scale given as the distance corresponding to the given line length. The possible values for this distance depend on the choice of unit made before. ↓*new* *new*↑
- map font and travel window font sizes: either in points or as `Tcl/Tk` for the Tcl/Tk default value. The map font will be used in map labels and can be changed from the map menu.
- interface appearance: number of maximum elements per menu, initial positions of windows, dimensions, colours.

- saving the program state on exit and deleting the saved state files after restoring.
- permission of created files (in Unix numeric notation).
- default paper size and usable paper dimensions.
- abbreviated names for months in all known languages.
- paper sizes and dimensions, used when saving plots or maps as Postscript files. The dimensions are floating-point numbers followed by `c` for centimetres, `i` for inches, `m` for millimetres, or `p` or nothing for printer's points (1/72 inch).

Chapter 4

Using GPSMan in graphical mode

4.1 Launching GPSMan

If GPSMan was installed from a Debian or RPM package, just call `gpsman` from a shell or from the applications menu of the window manager (if it was set up by the package installation program).

In other Unix or Linux systems call `gpsman.tcl` from the shell or use the shell script`gpsman.sh` (it must be configured first).

In these systems there could be a single argument with the path to the serial port device, or the call has the form ▌↓*new*
`gpsman [OPTIONS] start travel [INTERVAL]`
that will be parsed as if in command-linde mode (5), but then launches the graphical interface (this will only work with Garmin receivers). ▌*new*↑

In other systems, execute `gpsman.tcl` with the Tcl/Tk `wish` program.

4.2 Terminology

Here is a list of a few terms that will be used below.

Waypoints, routes, and tracks are examples of data used in GPS receivers. A waypoint (sometimes abbreviated to WP) describes a precise location through its geographic coordinates. A sequence of waypoints is called a route (RT) and is defined by the user. A track (TR, also called a trail in Lowrance receivers) is a sequence of track points (TPs) recorded by the GPS receiver over a time period and giving the positions, each with a time-stamp, of the receiver during that period, and including if possible the altitude and the depth (both in either metres or feet, depending on the existing option for this). ▌↓*new* ▌*new*↑

Route stages are the parts of a route between each two consecutive waypoints. Route stages are called *edges* in Graph Theory, *legs* in aviation, and *links* by Garmin. At present, GPSMan

deals with three data fields for each stage: a comment, a label (that will appear in the map), and hidden information. ▌↓*new*

Polylines (LN), also called polygonal lines, are similar to tracks but have points (LPs) without time-stamps. Polylines are mainly used as background in the map window.

Track and polyline segments are subdivisions that, in the case of tracks, normally indicate that there were time gaps in which the receiver got no position information. The representation in the map of a track or polyline having different segments is a polygonal line that is interrupted between each segment (segments with a single point will hardly be visible). Segments are defined by their starting points and in the track or polyline edit window it is possible to mark or unmark each point (except the first) as being a segment starter. ▌*new*↑

Data items refer to the elements stored in the GPSMan data-base. Apart from the data items the GPS receivers use and polylines, groups (GRs) of such items can be defined and used. ▌↓*new*
▌*new*↑

Forgetting a data item means deleting it permanently from the data-base.

Input/Output operations in GPSMan have the following names (see below for the definitions of the GPSMan file formats):

- *loading* from and *saving* to files in GPSMan format;
- *getting* from and *putting* into the GPS receiver (this corresponds to the terms *downloading* and *uploading*, respectively, used in other software);
- *importing* from and *exporting* to files in a foreign format. Currently recognized formats are given below (4.18).

  Allowed characters in waypoint names of depend on the brand of the receiver. Garmin names can only have uppercase letters and digits, even if Garmin receivers may use others (see the Garmin specification...). GPSMan also accepts lowercase letters if the existing option on this was set by the user, and hyphens. Lowrance names can have uppercase letters, digits, hyphen, single quote, period, parentheses, slash and also space. And there are no constraints on Magellan names. However, if data files are to be shared among users with receivers of different brands, the more strict rules (those of Garmin, at present) should be followed.

  When a name with characters not allowed is read from a file or from the receiver, the user is asked for a new name (alternatively GPSMan can generate it, or all names in the current input operation). A name of a waypoint already in the data-base will not be accepted. Cancelling the renaming makes the waypont to be ignored, what may cause inconsistency if it belongs to a route.

  Allowed characters in route names also depend on the brand of the receiver. Although until recently route names used to be numbers, there are now receivers accepting letters and other characters as well.

GPSMan does not check the characters in the route name, but will refuse to output a route with a non-numeric name to a receiver or file if the receiver protocol or the file format disallow it.

A unique name is used for each item of each type. When a new item is read in and it has the name of an item of the same type in the data-base the latter is forgotten and overwritten, unless they are waypoints with different positions and the renaming option was previously selected by the user.

It should be noted that

1. all input operations with the exception just mentioned are *destructive*: new items will replace data-base items having the same name. This is the behaviour of most GPS receivers, and avoids having obsolete information in the data-base.
2. the test for the equality of waypoint positions may fail because of rounding errors, at least when the comparison implies a change of position format or a change of datum.

Renaming raises the problem of generating a new suitable name for the item. Currently, GPSMan will try to keep the first part of the old name following it by digits. If the constraints on name length and uniqueness cannot be met, the new name will be a two-letter code for the item type and a hyphen followed by a number.

When an item is renamed, its previous name is kept in the remark field.

When generating a name for a new item or for replacing names with unacceptable characters, GPSMan will use a name with a two-letter code for the item type and a hyphen followed by a number, except in the case of routes for which a number will be used.

Renaming waypoints can lead to ambiguities in what are the actual waypoints of routes or groups. This will only happen when reading from a file in GPSMan format having routes with waypoints referred to by name only, or having groups, and in which there are different waypoints under the same name.

To minimise the problems with these situations, GPSMan creates a group containing the items that were renamed and those for which there may be ambiguities, after any input operation in which they occur.

Comments and remarks (NB) can be specified for some items. The difference is that comments can be got from and put into the GPS receiver, while remarks are only kept by the interface and may be saved to and loaded from GPSMan files. The syntax for comments depends on what the receiver accepts. The syntax for remarks is free: any ASCII character, any length, although no blank lines are allowed.

4.3 Data

The contents of the GPSMan data-base are shown in lists, one per each item type. Item names, which are unique, are presented in alphabetical order.

List menus contain the actions allowed on the list: creating a new item, clearing the list, reading/writing items, and counting the number of items in the list. The menu for the groups list is a little different and is described below (4.8).

Loading operations read all the data in a GPSMan file irrespective of from what menu the operation was launched.

To open an item for editing (only possible when no other item of the same kind is being edited) or viewing its data, use double-click on the item name with the mouse left-button. Double-clicking also works with the same meaning on other lists of item names, as well as on graphical representations of items (except polylines) in the map window. ▌↓*new*

An edit/show window for an item will be closed and re-opened in case there is a read operation redefining it. ▌*new*↑

To display/clear an item on/from the map click on the item name (in the corresponding list) with the mouse right-button, or use the `Clear` menu-button on the map window.

A read operation redefining items that are currently on the map will cause the map to be updated in order to keep it consistent with the new definitions.

Pressing a key on a list will scroll it to make visible the first element whose initial character is the same or higher in ASCII order than the key character. Note that this is case sensitive (i.e., `a` is not `A`). This also works on lists presented for choosing items. Lists can be scrolled by moving the wheel of a wheelmouse.

Hidden information is kept (in the data-base and in files) associated to a data item that has been read in (from a file, or from the receiver) when that information cannot be edited using GPSMan. This is done mainly with data fields that are not of general use, and provides a means of restoring the data item back to any receiver that works with the same communication protocols, without loosing information. When such a data item is modified its hidden information is either deleted, or kept, or acted upon as the user sees fit, according to an option. Keeping the hidden information may cause incoherent items to be created and therefore should be used with care.

Hidden information in a waypoint or a track can be displayed from their edit/show windows (a button for this is created only when such information exists).

Saving the program state when exiting is controlled by an option that may inhibit this feature, do the saving if the user confirms it, or do it without asking. When saving the state, files in the user data directory are created that contain the current data-base, the map state and information on which edit/show windows are currently being used. No information is kept on

1. the state of the communication with the receiver
2. the changes in pending edit operations or items partially defined
3. the state of computation, elevation, and real-time log windows.

The saved state is automatically loaded when GPSMan is launched and finds a saved state file in the user directory. There is an option to control whether to delete saved state files after the state being restored; it can be set in the same way as the save state option. In any case GPSMan automatically overwrites saved state files when saving a new state, and it is therefore a good idea to save important data to a file before quitting the program. ▌↓*new*

Note that in command-line mode the saved state is not restored and the state is not saved. ▌*new*↑

4.4 Waypoints

A position format and a datum for presenting the position of each waypoint are chosen by the user. Changing the format or the datum may be made at will, but too many conversions may degrade the accuracy of the data.

The position formats and datums of all the waypoints in a group can be changed in a single operation as described below (4.8).

Some information that may be relevant for choosing a datum is given when describing how to define new datums (4.12).

The following position formats can be used:

- `DMS` for degrees followed by minutes, both as integers, followed by seconds as a floating point number; the degrees value can be preceeded by a minus sign or one of the letters `N` or `S`;
- `DMM` for degrees as integer followed by minutes as a floating point number; the degrees value can be preceeded by a minus sign or one of the letters `N` or `S`;
- `DDD` for degrees as a floating point number; this value can be preceeded by a minus sign or one of the letters `N` or `S`; ↓*new*
- `GRA` for centesimal degrees as a signed floating point number; *new*↑
- `UTM/UPS` for easting zone number, northing zone letter, easting and northing of Universal Transverse Mercator or Universal Polar Stereographic coordinates.
- one of the available grid systems, either predefined or user-defined, with a zone identifier (void for some grids), a easting and a northing (both in metres by default; user-defined ones may be in feet). The predefined grids are: ↓*new*
  - `BMN`: Austrian "Bundesmeldenetz"; *new*↑
  - `BNG`: British National Grid;
  - `BWI`: British West Indies grid;
  - `CMP`: Portuguese military maps grid; ↓*new*
  - `CTR`: Italian Carta Tecnica Regionale; *new*↑
  - `GKK`: German grid ("Gauss-Krueger-Koordinatensystem")
  - `IcG`: Iceland grid;
  - `ITM`: Irish Transverse Mercator;
  - `KKJP`: basic Finnish grid;
  - `KKJY`: uniform Finnish grid; ↓*new*
  - `Lamb93`: French Lambert 93 grid;
  - `LambNTF`: French Lambert NTF ("Nouvelle Triangulation de France") grid;
  - `LambNTFe`: French Lambert NTF ("Nouvelle Triangulation de France") zone II *étendue* grid; *new*↑
  - `LV03`: Swiss grid in the LV03 frame;
  - `RDG`: The Netherlands grid;
  - `SEG`: Swedish grid; ↓*new*
  - `TAlbers`: Teale Albers grid (used in California, USA); *new*↑
  - `TWG`: Taiwan grid.

  More details on these grids can be found below (4.13).
- `MH` Maidenhead locator coordinates: this is a special kind of grid used mainly for specifying the position of radio stations, each smallest subdivision being 5 minutes in longitude and 2.5 minutes in latitude. This means that conversions from other position formats to this one will most probably loose accuracy.

The altitude for a waypoint is given as a (possibly signed) floating point number in either metres or feet, depending on the existing option for this. ↓*new* *new*↑

A symbol and a display option are also chosen for each waypoint. GPSMan symbols and display options may not all be supported by the receiver. When GPSMan is aware of this a tilde ~ will appear before the symbol name in the symbols menu. Symbols and display options not supported will be transmitted to the receiver as the default values; if these are also not supported, the symbol will be transmitted as a waypoint dot, and the display option as "Symbol & name". ▌↓*new*

Symbols of all the waypoints in a group can be changed in a single operation as described below (4.8). ▌*new*↑

Creating a waypoint can be made as for other items from menus and also in the following ways:

▌↓*new*

- at a given distance and bearing in relation to an existing waypoint, from its window; the maximum allowed distance is about 20000km (more precisely, π times the ellipsoid semi-major axis); if the distance is greater the new point will coincide with the old one; ▌*new*↑
- from the map, if the map has been geo-referenced: see below (4.10);
- from a track, either as one of its points, or by taking an average of the coordinates: see below (4.6);
- from the waypoints in a group by taking an average of the coordinates: see below (4.8); ▌↓*new*
- (Garmin variant only) from the travel menu, by taking the coordinates of the last point in the real-time log if there is one: see below (4.15.2). ▌*new*↑

The position of a waypoint displayed on the map can be changed there through a menu, as described below (4.10).

Clusters of waypoints can be created by giving a group of waypoints to be used as centres: see below (4.8).

4.5 Routes

Routes may happen to have waypoints that were permanently deleted by the user (`Forget` button in waypoint window). In this case the values of distances and bearings for such points and the total distance will not be shown. Saving, exporting or displaying a route with undefined waypoints will be prevented with a warning.

Routes have a colour and a width (in pixels) used in displaying it in the map. ▌↓*new*

▌*new*↑

A route can be created from a track : see below (4.6).

A route can be changed or created by drawing on the map as explained below (4.10).

To change a route stage a double-click with the mouse left-button should be made on one of the stage fields in the route edit window. An edit window will pop up that must be used and closed before going on.

Changes in a waypoint belonging to a route being edited/inspected will be reflected in the route window.

When modifying a route the coherence of its waypoints and its stages cannot be checked by GPSMan. For instance, when adding a new waypoint after another one the stage starting from the latter is not affected, and when replacing a waypoint by another one the stages ending on and starting from it are not affected.

The edit window for routes allows some operations on routes that may be useful. They are:

- "Invert": take the route from the last to the first waypoint;
- "Chop head": all waypoints from the first to and including the one selected are deleted; if there is no selection the first waypoint is deleted;
- "Chop tail": all waypoints from and including the one selected to the last are deleted; if there is no selection the last waypoint is deleted;
- "Include before", or "Include after": include the route (whose number was selected in the sub-menu) before or after the selected waypoint; if there is no selection, the inclusion will be done before the first or after the last waypoint, respectively.
- "Clear": delete all waypoints.

A route can be converted into a track from the route window.

An elevation graph for a route can be plotted, as a side view or in perspective, from the route window if there are at least 3 waypoints with a valid altitude field.

Clicking with the mouse left-button in a side-view graph will draw a line with the numbers of the waypoints at that horizontal coordinate. The `Shift` key and mouse left-button will clear all these lines.

The perspective graph is shown from South but different viewing directions can be obtained by using the N-E-S-W buttons, or the "Show" button after the "+15", "-15" (degrees) buttons. The bearing scale shows the viewing direction in degrees that is the current one only when the "Show" button is disabled (this button updates the graph). The scale can also be used to change the viewing direction if the "animate" button is checked. As the animation or a change of viewing direction may lead to a long computation, there is an "Abort" button to stop it. The menu from the "View" button allows for changes in the vertical or horizontal scales, and to hiding/displaying labels in the graph. A N-E-S-W cross is displayed in the graph and can be moved by using the mouse left-button. Clicking with the left-button on a point of the route and then using the mouse middle-button (or left- and right-buttons in a two-button mouse) will move the whole graph.

Elevation graphs can be saved as a Postscript file or further processed (e.g., printed) in Postscript — cf. the "print command" option (3)), or saved in other graphics formats if the Img Tcl/Tk library is available. This library has two problems when saving an image:

- any window or icon over it will also be part of the saved image; and
- depending on the format, errors can occur if the image has too many colours.

The area enclosed by a route can be computed under the following conditions — badly wrong values will result if they are not met! The route stages are taken as sides of a polygon and if the last waypoint is not the same as the first, a "virtual" side from the first to the last waypoint is considered. The polygon must be non-intersecting: there can be no multiple occurrences of waypoints (apart from the first one being also the last) and no intersections of the polygon sides. GPSMan will only check for multiple occurrences of waypoints. The method for computing areas is an approximate method that is not reliable when there are

sides of the polygon too small when compared to others or there are very small angles between the sides. Results of area computations should be used with care and if possible checked against results of other forms of area measurement.

The details of the area computation are as follows. An algorithm for computing the area of a (non-self intersecting) polygon on the sphere is first tried out. If there are very small intermediate values that may indicate approximation errors, the area is computed by first projecting the polygon onto the plane (using the Transverse Mercator projection centred at the first point of the polygon) and applying then an algorithm for computing the area of a (non-self intersecting) polygon on the plane. A warning message is issued if this happens.

4.6 Tracks

Tracks have a datum for all its points, a colour and a width (in pixels) used in displaying it in the map. The colour will be sent to the GPS receivers supporting it, in which process a colour matching algorithm will be applied if the original colour is not in the set of colours accepted by the receiver. Any colour matching algorithm may give unexpected results and the one used in GPSMan is no exception. ▌↓*new* ▌*new*↑

Each track point has the following information: time stamp, position (always shown in the DMS format), altitude and depth in either metres or feet, depending on the existing option for this. In the computation results there are six fields for each point: distance to next point, cumulative distance to next point, altitude in the unit selected by the user (a negative number means depth instead), time to next point, speed in the line to the next point and bearing to the next point. ▌↓*new* ▌*new*↑ ▌↓*new* ▌*new*↑

A track can be subdivided into segments by some receivers at points in which the GPS fix was lost. The first track point always starts a segment, and any other track point can be marked as a segment starter from the track edit window by using the mouse right-button on the last column of the points list. When displaying a track with segments on the map, the segments will not be connected. ▌↓*new* ▌*new*↑

The edit window for tracks allows some operations on tracks that may be useful to clean uninteresting start or end parts of a track, or to compose a single track from several others. They are:

- "Chop head": all track points from the first to and including the one selected are deleted; if there is no selection the first track point is deleted;
- "Chop tail": all track points from and including the one selected to the last are deleted; if there is no selection the last track point is deleted;
- "Include before", or "Append": the track points of another track are put before the first, or after the last track point. To ensure sensible values for speed between track points, their dates may have to be changed. GPSMan will show the distance between the last point in the first track to the first point in the second and will propose a new date for this one. This date is computed assuming a constant speed and may be changed by the user. All dates in the second track will be adjusted according to the selected date, keeping the original differences.

An elevation graph for a track can be plotted, as a side-view or a perspective, from the track computation window if there are at least 3 track points with a valid altitude field. They are similar to the elevation graphs for routes (4.5), although the side-view graph can be plotted against time instead of total distance if there is valid time information. ▌↓*new* ▌*new*↑

A speed graph for a track can be plotted from the track computation window if there are at least 3 track points with valid time information. It is similar to the side-view elevation graph and can also be plotted against time instead of total distance. ↓*new*

A climb rate graph for a track can be plotted from the track computation window if there are at least 3 track points with valid time information. The graph is plotted against time, the vertical units being (user selected) altitude unit per second. If there is enough points a noise-reducing filter, kindly provided by Paul Scorer, is applied to the data. *new*↑

Creating a waypoint from a track point can be done by double clicking with the mouse left-button on a track point listed in a track window. This will open, for edition, a new waypoint having the same coordinates unless there is already a waypoint being edited. If the track is currently on the map the number of each track point together with the track name will appear in the help balloon when the cursor is over the point.

A waypoint with average coordinates can be created from a track window. Its latitude, longitude and altitude will be computed as the averages of the latitudes, longitudes and altitudes of the track points. This will be useful for obtaining more precise coordinates for a waypoint by recording a track with the receiver standing still.

A track can be converted into other line items by a simplifying algorithm that keeps a certain number of the track points as points of the new line, which can be a route, another track, or a polyline. When converting to a polyline information on segments is only used if all the points are kept. ↓*new* *new*↑

The algorithm that was developed for this may be seen as a variant of the Douglas-Peucker algorithm for finding critical points in polylines (see, e.g., [Heckbert and Garland, 1997] or [Li, 1995]). It starts from a straight line between the first and the last track points; if the number of points to keep is greater than 2, any point that stands furthest from the line will be retained, and the line is replaced by two new lines, those from the first to the new point and from it to the last one. This procedure is repeated always replacing one of the lines for which the distance to an intermediate point is maximum. The review of [Heckbert and Garland, 1997] describes an algorithm by Ballard and Brown (published in 1982) that seems to be very close to this one.

Although GPSMan lets the user fix the number of points to keep between 2 and the number of track points, there is a maximum number of points per route depending on the GPS receiver. It should also be noted that the time needed to find the simplified line will increase significantly with the number of points (although keeping all the track points will take only the time to create the new item).

So that a choice may be made between different numbers of points, GPSMan may be asked to display the simplified line and also the original track on the map on the fly. When the user clicks the `Ok` button, the map will be restored, the simplified line is used in forming a new item (in the case of a route, with new waypoints having names of the form `ZT`n, with n a 4-digit integer), and an edit window will be opened for editing the new item. If the edit window for the item type was already in use, then the item is created under an automatically generated name. For a route GPSMan will create a new group with all the new waypoints for easier access.

An animation of the movement corresponding to a track can be viewed in the map window (`Animation` button in the track edit/show window). A control window will appear that allows for (re-)starting, pausing, or aborting the animation, for skipping to the next track point, for setting the speed (the scale changes are exponential), and for choosing whether the last point shown will be centred on the map window. The default speed is that in the track: the delay between the presentation of two consecutive points is the difference between their time stamps. If a time stamp is not defined the default delay is 30 seconds. The state of the animation and the total (real) time since the beginning (if defined) are displayed.

A track can be created from a route or a polyline from the route window or the polyline window. ▌↓*new* ▌*new*↑

4.7 Polylines

▌↓*new*

Polylines have a datum and a position format for all its points, a colour and a width (in pixels) used in displaying it in the map.

Changing the position format to grid coordinates may produce the following effects:

- the polyline datum is changed to the datum required by the grid
- some/all positions are shown as undefined (either `-- 0 0` or `-- 0`) if they are out of the scope of the grid; this however will not affect the internal representation of the position and a change to a suitable format will yield correct values. When reading a polyline from a file, positions may also be shown as undefined in which case the position format should be changed.

Each polyline point has a position and, possibly, its altitude in either metres or feet, depending on the existing option for this. A point can be changed by double-clicking on the corresponding line in the list of points of the polyline edit window.

A polyline can be subdivided into segments and the segments will be drawn unconnected when the polyline is displayed in the map. The first point always starts a segment, and any other point can be marked as a segment starter from the polyline edit window by using the mouse right-button on the last column of the points list.

A polyline can be created from a track : see above (4.6).

A polyline can be created by drawing on the map as explained below (4.10).

The edit window for polylines allows some operations on polylines that may be useful to clean uninteresting start or end parts, or to compose a single polyline from others. They are:

- "Chop head": all polyline points from the first to and including the one selected are deleted; if there is no selection the first point is deleted;
- "Chop tail": all points from and including the one selected to the last are deleted; if there is no selection the last point is deleted;
- "Include before", or "Append": the points of a polyline are put before the first, or after the last point.

– "Loop": a copy of the first point is added to the end of the polyline.
– "Delete": deletes the selected point, if any.
– "Clear": deletes all points.

A polyline can be converted into a track from the polyline window, keeping the information on segments. *new*↑

4.8 Groups

Groups are very useful in cataloguing the available data and in operating on sets of items. The possible operations are

– to forget items,
– to display/clear items on/from the map,
– to transfer items to/from the receiver or to/from a file,
– for a set of waypoints:
  * to define an average waypoint (4.8),
  * to change their position formats (4.8),
  * to compute clusters (4.8).

As groups are also used by GPSMan to present the results of a search as described below (4.9), searching for items is an effective way of creating a group on whose items an operation is then performed.

A group contains a certain number of data items and is represented internally as set of item names (together with their types). Operations on a group may fail or only partially succeed if one of its elements is not currently in the data-base.

Groups can have other groups as elements but one group cannot be an element of itself even if indirectly (in technical terms: groups are well-founded sets).

Clearing from the map an item that belongs to a group that has been displayed will not affect the display-state of the group. To be sure that all the elements of a group are actually displayed, the user should clear the group from the map and then display it again.

Deleting from or adding items to a group will not affect their display-state.

Forgetting a group will delete permanently the group from the data-base but not its elements. This operation is not prevented by the fact that any of its elements cannot be cleared from the map.

Forgetting a group and all its elements will delete permanently not only the group but also all its elements (recursively, i.e., including the elements of groups in the group). The group is deleted even if some of its elements cannot be cleared from the map and is therefore not deleted.

Saving a group (to a GPSMan file) will save all the information on the group and on its elements.

Creating an average waypoint from the waypoints in a group can be made from the group window. The coordinates of the new waypoint will be the averages of the latitudes, longitudes and altitudes of waypoints in the group and its sub-groups (recursively).

Changing the data of waypoints in a group can also be made from the group window in what concerns:

↓*new*
new↑

- the symbol,
- the position format, or
- the datum.

All the waypoints in the group and its sub-groups (recursively) will change to the (same) selected value. If one of the waypoints is being edited, the edit window will also be changed. In the case of the position format or the datum the position will revert to its initial value (when the edit window was created). In the case of the symbol, the change will be reflected on the map if necessary.

↓*new*
new↑

Clusters of waypoints can be created from a group by taking the waypoints in it (and its sub-groups, recursively) as centres of the clusters and searching the data-base for waypoints that fulfil a selected condition for each centre. The conditions that can be tested are: that the waypoint is within a given distance range of the centre, or that the waypoint belongs to the quadrangle of given latitude and longitude ranges whose middle-point is the centre. It is obvious that the first condition will be much slower to evaluate than the second, and therefore making clusters based on quadrangles should be preferred when the number of waypoints currently defined is large. Each cluster will be created as a group: its name is of the form `Cluster` n, and its remark has the name of the centre and either the quadrangle dimensions, or the distance range.

Input/output operations on the elements of a group allow for selecting which items of which types to read or write. In general the user will choose the groups and the item types for the operation. Then GPSMan collects in a list the names of the items of the given types that belong to the selected groups. This list of names is used to perform the I/O operation.

Selecting the "Group" type means that the search for items will be done in the groups that are elements of the selected groups, recursively. In more technical terms, the resulting list may be seen as a flattening of the group structure. In no case the list of names will contain names of groups.

Details of each specific operation are as follows:

- in output operations, the "All" menu entry means that all groups will be considered. When saving or putting elements this also means that all types should be considered.
- in input operations, there is the option of reading either the items whose names are not in the list of names GPSMan builds, or the items having the names in that list. The former is useful for preserving data in the selected groups; items that are not in the data-base will also be read in. The latter is useful for updating or restoring the information in the selected groups without affecting the other data; items that are not in the data-base will not be read in. All items of non-selected types will be discarded except waypoints belonging to routes if the route type was selected.
- when exporting or importing information to/in any foreign format with a single type per file, a single type (apart from "Group") must be chosen.
- when getting information from the receiver the "Track" type cannot be used as there is no point in updating or changing previously recorded tracks.

4.9 Searching data items

In order to search data items the user specifies a set of constraints. An item will be included in the search results only if it verifies all the constraints in the set that are applicable to its type.

The types of items to be searched for can be more than one, to each type being applied only the constraints that make sense with it.

The search domain is either the entire data-base, or a set of groups. In the latter case, the search will be recursive, i.e., will also explore the groups that are elements of the given groups, and so on. Furthermore, if the search includes the type "Group", the given groups will be included in the search results.

The patterns for searching by names, comments and/or remarks follow the Tcl/Tk `glob` command conventions. In brief:

1. `?` stands for any single character
2. `*` stands for zero or more characters
3. `[`*xyz*`]` stands for any of the characters within the brackets
4. `[`*a*`-`*z*`]` stands for any character in the range *a* to *z*, inclusive
5. `\`*c* stands for the character *c*.

The distance to a waypoint or to a location given by its coordinates can be used to search for waypoints (a related operation is making clusters of waypoints (4.8)) and/or tracks. With tracks all track points of each track may have to be checked what may take a long time.

The search is based on either an allowable maximum distance, or a distance interval. A bearing for the search can also be given, together with an angle that will be centred along it.

Results , if any, are presented as elements of a new group with a name of the form `FOUND n` where `n` is a number. The remark of the group gives a succint description of the constraints used in the search.

A dialog window will be presented giving the choice between ending the search while keeping (`Ok` button) or forgetting (`Cancel`) the group with the results, and making a new search while keeping (`Another`) or forgetting (`Change`) the group.

4.10 Map

The map window will contain a graphical representation of data. Several operations on the map can be made using the mouse or the keyboard and a summary of these can be found Appendix C.

Its contents can be saved as a Postscript file or further processed (e.g., printed) in Postscript — cf. the "print command" option (3).

It is assumed that the user has chosen the relevant datum and projection before asking for some data to be displayed.

Some information that may be relevant for choosing a datum is given when describing how to define new datums (4.12).

The available projections and the way new projections can be defined are described below (4.13). Projections may have parameters, in which case they are computed either when a data item is displayed and the map is void, or when a map background image is loaded. According to an option the user is asked to accept or change them.

When a map background image is loaded it will be geo-referenced and a transformation of coordinates may be selected for that purpose. There are three such transformations: affine, which covers rotation and non-conformality, affine conformal, and affine conformal with no rotation, that corresponds to applying only a scale factor and that is used when there is no background image. Obviously there will be deformation when either the projection or the transformation is not suitable for the image.

More detailed explanations of how to use background images and projections and coordinate grids are given in the next sections (4.11, 4.13).

Measuring distances and azimuths on a non-empty large-scale map can be done by using the mouse right-button when pressing the `Shift` key to select a sequence of positions (at a distance greater than 1 metre). Arrows between each position to the next will be displayed and a dialog will show the total distance and the azimuth of the last position from the first. The arrows will be deleted when the dialog is closed. The dialog allows for the line formed by the arrows to be closed (linking the last position to the first, unless they stand at less than 1 metre) and to be used in creating a polyline item (LN). Note that distance values computed in this way can be wrong on small-scale maps if two consecutive points are too far away from each other. ↓*new* *new*↑

Items can be displayed on the map by using the `Display on map` button in the map window. Other methods include:

- using the `Display on map` option when reading new data from files or the receiver,
- right-clicking on a name in an items list,
- using the `Display on map` option of the edit window for an item,
- using the display entry in the menu that pops up with `Control`-key left-click on a waypoint in the map.

To clear an item from the map there is the `Clear` menu-button in the map window. Other ways of achieving the same effect:

- right-clicking on a name in an items list,
- using the `Display on map` option of the edit window for an item,
- using the clear entry in the menu that pops up with `Control`-key left-click on a waypoint in the map.

A waypoint can be created on the map , if the map has been geo-referenced, by clicking with the mouse left-button on an empty place, or by using the `Return` (or `Enter`) key. This can only be done when no waypoint is being edited. When a route is being edited on the map the `Return` key has no effect, and the left-button on an empty place creates a waypoint that is added to the route (see below (4.10)). The position format for the new waypoint will be the one in use for the map cursor coordinates. UTM/UPS will be used instead when the position is out of the range of the selected grid. To finely position the cursor, the arrow keys for scrolling the map and the `Return` key should be used instead of the mouse.

A menu-button for a waypoint on the map will be created by pressing the `Control` key and clicking on the waypoint with the mouse left-button in Unix/Linux systems, or only the mouse right-button in other systems. It will allow for moving the waypoint (i.e., changing its position), starting the definition of a route (see next paragraph), or for displaying or clearing:

- all waypoints within a certain distance;
- all waypoints in the rectangle having as opposite corners this waypoint and a waypoint chosen from the menu;
- all routes containing this waypoint;
- all routes containing waypoints on the map.

A waypoint that is being moved is placed in its new position by using the mouse left-button. The right-button cancels the operation. A balloon will show the possible actions.

A route can be changed or created on the map by using

- the `Edit on map` button from the route edit window; there must be at least one waypoint in the route; or,
- the `Start RT` entry of the menu corresponding to a mapped waypoint when no route is being edited; the route edit window will be opened and the waypoint becomes the route starting point.

In either case, changes made on the route on the map will appear in the route window. The cursor will show the current insertion point that at first is the end point of the route, but that can be moved to in between any two waypoints of the route (if there are as many) — this may be seen as changing the corresponding route stage.

Edit operations are performed by using the mouse buttons and the `Shift` and `Control` keys, and/or by using a menu that will appear by pressing the `Control` key and clicking the mouse left-button (not on a waypoint!).

Clicking with the mouse

- left-button on a waypoint adds it to the route; a waypoint cannot follow itself in a route;
- left-button where there is no waypoint, creates a new waypoint and adds it to the route; if the operation is cancelled the waypoints created this way will be discarded;
- left-button together with the `Shift` key removes the previous waypoint from the route unless there is only one;
- right-button in Unix/Linux systems or `Control` key and the mouse left-button in other systems, stops the route definition from the map; if there is a waypoint under the cursor it will be added to the route, otherwise a new waypoint is created and added to the route; the route defined so far can now be further edited in its window;
- right-button with the `Control` key changes the insertion point to the previous stage, if there is one;
- right-button with both the `Control` and the `Shift` keys change the insertion point to the next stage, if there is one, or, when changing the last stage, to the end of the route;
- middle-button (or left- and right-buttons) with the `Shift` key cancels the definition; the same can be achieved by using the `Cancel` button of the route edition window.

The operations available from the menu (`Control` key and the mouse left-button in Unix and Linux systems, or only the mouse right-button in other systems, not on a waypoint) are the following (only those that are meaningful will be shown at any given moment; the corresponding shortcut using the mouse/keyboard is shown if there is any):

- stop editing on the map, and either include the current point under the cursor (right click), or do not add any more points;
- cancel the whole edit operation (`Shift` and middle click);
- delete either the previous waypoint (the one before the current insertion point; `Shift` and left click), or the first waypoint of the route;
- edit previous stage, i.e., change the insertion point to the previous stage (`Control` and right click);
- edit next stage, i.e., change the insertion point to the next stage, if there is one, otherwise to the end of the route (`Control` and right click);
- add to end, i.e., change the insertion point to the end of the route (when editing the last stage, this can be done with `Control` and right click);
- close menu, destroying the menu-button.

When using `Control`-right click and `Control-Shift`-right click to go from one stage to another the lines in the map are only redrawn when the cursor moves.

During the edition of the route, waypoints can be moved to other positions as described above (4.10).

Scrolling and panning the map can be done by using the `Locate` menu-button, the mouse, the keyboard, or a wheelmouse.

Selecting an item with the `Locate` menu-button (only items on the map are listed) scrolls the map so that the selected item becomes centred. In case of a route, track or polyline this applies to its first point. ▌↓*new* ▌*new*↑

Dragging the mouse with the middle button down or moving it with the `Control` key pressed will pan the map.

The keyboard arrow keys and the `Space` and `Delete` keys scroll the map in the expected way, while the arrow keys with the `Shift` key scroll the map in the SE-NW and NE-SW directions.

Users of a wheelmouse can use the wheel in it for the same purpose: with no modifier key for vertical motion, with the `Shift` key for vertical motion by one page, with the `Alt` key for horizontal motion, and with the `Control` key for horizontal motion by one page.

As the cursor coordinates are updated when the cursor moves, the use of the keyboard for scrolling is also a means for finely positioning the cursor.

Reading items that are on the map will update the map, so that the items are shown according to their newly read definitions.

4.11 Map background images

A map background image can be loaded either from a file containing an image in a recognized graphics format, in which case it must be geo-referenced, or from an *image information file* (see 4.17.2 for its format) that contains geo-referencing information together with the path of the files containing images. GPSMan will automatically detect the kind of file it has to load from when one selects either the entry `Map->Background->Load` from the `Map` menu-button (if the main window is the map window), or the entry `Load` of the `Background` menu-button (if the main window is the lists window). If the file is an image information file, loading the image and setting up the map window is done without user intervention.

It only makes sense to have as background images maps in one of the projections (4.13) that GPSMan implements.

Tcl/Tk accepts both the GIF and PNM graphics formats. GPSMan tries to load the Img Tcl/Tk library that provides support also for JPEG, TIFF and other formats. This library has two problems when saving an image:

- any window or icon over it will also be part of the saved image; and
- depending on the format, errors can occur if the image has too many colours.

4.11.1 Geo-referencing an image

Geo-referencing a background image consists in selecting a projection and a coordinates transformation and in placing control waypoints in the image in order to fix the values of the parameters of the projection and the transformation. An alternative is to indicate that there is a TFW file or a OziExplorer map file associated to the image. ↓*new* *new*↑

Only then is it possible to compute map coordinates from geodetic coordinates or the inverse. In fact, to convert from geodetic coordinates (i.e., coordinates in the spheroid, typically as latitude and longitude) into pixel coordinates in the screen there are two operations:

1. a projection, that from geodetic coordinates computes Cartesian plane coordinates, and
2. a coordinates transformation, that from Cartesian plane coordinates computes pixel coordinates. This is needed because the image can be rotated or distorted.

When geo-referencing an image to be loaded from a file in one of the accepted graphics formats, the following information must be known:

- the projection used in the image,
- the datum,
- the coordinates transformation to use, unless there is a TFW or OziEplorer map file defining it; see below (4.11.2). ↓*new* *new*↑

If the image is that of a map, the projection and the datum will hopefully be described in it. The corresponding options should be selected from the two menu-buttons at the left on the bottom of the map window, if the map window is the main window, or the `Datum` and `Projection` menu-buttons of the map window, if the lists window is the main window. If a TFW or OziExplorer map file is used, the projection and the datum are selected from a dialog after the file is read. ↓*new* *new*↑

As to the datum, it must be emphasised that some maps have a cartographic datum (the one used for projecting the map elements) and then one or more sets of grid lines projected using other different datums. The datum to be selected in GPSMan is the cartographic one, not any of those for the grids. On the other hand, if the intersection points of a grid are to be used as control points for geo-referencing the image, the datum for the grid should be used when creating the corresponding waypoints. More information on datums can be found below (4.12).

After having selected the projection and datum as described above, geo-referencing proceeds by choosing the coordinates transformation and by giving information from which the parameters for the transformation and possibly for the projection can be computed.

4.11.2 Coordinates transformations

The transformations presently available in GPSman are

– affine conformal with no rotation. It is the one to choose if the central vertical line of the map image is oriented North-South (geographic, not magnetic) and there is no distortion in different directions (i.e., in each point the map scale is the same irrespective of the direction that is considered).
– affine conformal. To be used when the vertical of the image is not oriented N-S, and there is no distortion in different directions.
– affine. This is the more general case: the vertical of the image is not oriented N-S and there may be distortion in different directions. It is used if geo-referencing is done with a TFW or a OziExplorer map file. ↓*new* *new*↑

Geo-referencing using control points needs 2 or 3 control waypoints with known coordinates. They are placed over the image so that both their geodetic coordinates and their pixel coordinates are known. These waypoints can either be selected from the existing ones, or be defined when geo-referencing the image; in the latter case only the name and position will be asked for, and if no name is given one is automatically generated. If they are defined beforehand it is a good idea to use either no symbol, or to use the `Mark, x` symbol so that later on they can be placed exactly where they should be in the image. If the transformation is the "affine conformal" one 2 waypoints will be needed, otherwise 3. In the latter case, the waypoints should be chosen so as to form an almost equilateral triangle, in order to minimise positioning errors.

After defining any control waypoints, either the entry `Map->Background->Load` from the `Map` menu-button (if the main window is the map window), or the entry `Load` of the `Background` menu-button (if the main window is the lists window) should be selected. The image file to be loaded is then chosen.

Dialog windows will then allow for the selection of the transformation to be used and the 2 or 3 waypoints that will serve as control points. The image is then presented in the map window.

In the case of the "affine" and "affine conformal" transformations each waypoint is then placed by the user over the image where it should be and its name and position entered if it did not exist.

If the "affine conformal no rotation" transformation was selected, the user is asked first for the names and positions of the control waypoints to be defined if there are any. One of the waypoints is placed first and 2 lines will de drawn. Each of the other 2 waypoints must be over each of these lines. When the mouse is moved, the 2 wayponts will move over these lines until the user clicks the left-button to place them both at the same time. The operation is finished by clicking on the `Ok` or `Cancel` buttons of the dialog window.

It is a fact that for the "affine conformal no rotation" transformation, 2 waypoints would be sufficient. However GPSMan asks for 3 to be placed so that the user may place 2 of them at the same time, in this way having more control on positioning errors. The 3 waypoints should form an almost equilateral triangle that can be shown in the map window. The order of the 3 waypoints is important, as the first one cannot be moved after being placed. This waypoint, then, should be such that there are no doubts on where it should go. It will be shown together with lines that will contain the other two, and will be placed by clicking the left-button. The other two will be placed as a pair in the same way, scale changes being displayed. ↓*new*

Using a TFW or OziExplorer map file is an alternative way of fixing the parameters of the affine transformation. Unfortunately, there is no publicly available descriptions of the formats of these files and the implementation is based on a popular guess of the meaning of the values in TFW files and on observing sample files.

With a TFW file (Tiff World file) the only thing to be done is to select either the entry `Map->Background->Load` from the `Map` menu-button (if the main window is the map window), or the entry `Load` of the `Background` menu-button (if the main window is the lists window), select the `TFW file` method and choose the file, unless there is a file with the same base name as the image file and the extension `.TFW` or `.tfw`. After this, the projection and the datum are selected/changed and a dialog window may appear if there are projection parameters that can be changed.

With a OziExplorer map file the method is selected in a similar way and a file must be selected if there is not a file with the same base name as the image file and the extension `.MAP` or `.map`. Only part of the information in the file is used, namely: the datum, and the geodetic and pixel coordinates of control points. Three control points are chosen by finding the triangle whose side with minimum length is maximum. They are used in computing the affine transformation parameters. Information on the projection as found in the file is displayed in a window to help in selecting the projection and in setting the projection parameters. *new↑*

4.11.3 After geo-referencing an image

The map scale cannot be changed if there is a background image, and an image can only be loaded to an empty map.

After geo-referencing an image, the information on it can (should) be saved through the entry `Save geo-ref info` (under either `Map->Background`, or `Background` menu-buttons) so that the next time it can be loaded with no need for geo-referencing.

Other background images can be loaded after having one image geo-referenced by using the `Change` option of the map `Background` menu.

All images must have the same datum, projection and coordinates transformation as the first image. Each image will be described by the path of its file.

In order to load different sheets of a map to the background, images assumed to have exactly the same size as the first image can be loaded to a slot in a grid. This is done by selecting the grid slot in the diagram that is shown and using the `Load` button. Selecting a non-empty slot will show the file path of the corresponding image.

For loading images that cannot be taken as being in a grid (overlapping other images or having different sizes) the `Load` button in the right panel of the dialog should be used. A single control waypoint is needed to be placed over the new image and can be either selected from the data-base or defined before being placed. The list of file paths for the images loaded in this way is shown in a list. Selecting one of them will create a representation of the image in the diagram of the left panel. It will appear only when the cursor is on the right panel, and it may be too far away to be seen.

The `Clear` buttons in the dialog will remove the selected images in either panel. The first image cannot be removed.

4.12 Datums and ellipsoids

A horizontal (or geodetic) datum defines the form and the position relative to the Earth axis of the geometric reference surface of the Earth used for locating points and in projections. The form is an ellipsoid which is usually defined by giving its semi-major axis `a`, and its flattening `f` (or its inverse), i.e., the quotient of the difference between its semi-major and semi-minor axes by its semi-major axis. Its relative position is described by the shift in

Cartesian coordinates (`dx`, `dy`, `dz`) with respect to a reference datum, usually the "WGS 84".

GPSMan contains comprehensive sets of datums and ellipsoids. Their definitions have a remark field used for documenting them whenever possible, as well as fields for the error estimate in metres (`ex`, `ey`, `ez`; the value `-1` stands for unknown), the number of satellite measurement stations, and the zone of validity given by S-N latitudes and W-E longitudes. All these fields are for information only and may be empty. The definitions can be inspected (but not changed) from the corresponding entries under the `Definitions` menu-button. It should be noted that some datums have variants for different regions. For instance, the "European Datum 1950" has at least 15 such variants and it has been observed that Garmin receivers do use the local variant for Portugal/Spain when this datum is selected and a waypoint in Portugal is created. This means that using the average "European Datum 1950" in such a situation may lead to large position errors. Probably the same will happen with other datums having variants.

Users may define their own datums and ellipsoids from the entries under the `Definitions` menu-button. These definitions, that cannot override those in GPSMan, are automatically saved in a file in the GPSMan user directory, and will be loaded when GPSMan is started. Currently GPSMan does not prevent changing or forgetting a datum or ellipsoid that is in use: it is the user's responsability to avoid inconsistencies due to such operations. When sharing files having data depending on user-defined datums with other users, the definitions of the relevant datums and ellipsoids should also be shared.

4.13 Projections and coordinate grids

4.13.1 Selecting and defining projections

Selecting the map projection is done by using the second (from the left) menu-buttons on the bottom of the map window if the map window is the main window, or the `Projection` menu-button if the lists window is the main window.

If a background image is to be loaded the projection and the datum should be set to the projection and datum used in the image (see above (4.11.1) for the details on this). If there is no image, the map projection should be selected according to the map scale and the geometry of the region to be covered. ▌↓*new*

World maps and small-scale maps need suitable projections, such as Mercator. Using projections that were developed for large-scale maps, such as the Transverse Mercator will give strange results or even errors. With small-scale maps the following should also be noted:

- a map cannot have overlapping parts; this implies that for each point on the Earth the projection procedure will give a single projected point;
- for projections having a central longitude or false longitude parameter, GPSMan converts any longitude value to the range from -180 to 180 (inclusive) centred on that central/false longitude; this means that it may happen that a point created too far to the West (or East) of this longitude will be mapped to the East (or West) of it, causing lines (routes, tracks or polylines) to be displayed in a wrong way; it is therefore very important that the central or false longitude be given an appropriate value;
- when initializing a map projection the default value for the central/false longitude in GPSMan is the average of longitudes of the points that are to be displayed then; if the option on asking for confirmation of projection parameters is set, the user will have the opportunity to change it;

– measuring distances on a small-scale map as explained above (4.10) can yield wrong values for points too distant from each other.

new↑

Projections can be either predefined or user-defined. There are a small set of predefined projections. Some of them admit particular cases, in the sense that they have parameters whose values can be fixed. The user may define such particular cases along with a coordinates grid associated to it.

Each projection has an associated coordinates grid that will be used as default position format for displaying the map cursor coordinates and when a waypoint is created from the map. This position format can be changed from the map window (menu-button near the cursor coordinates).

When defining a projection, the user may also define a new coordinates grid User-defined grids cannot have more than one zone.

User-defined projections and grids are automatically saved in a file in the GPSMan user directory, and will be loaded when GPSMan is started.

To define/change a projection there are the appropriate entries under the `Definitions` menu-button.

When defining a new projection, which is necessarily a particular case of a general projection, the user must select first the general projection to use, along with a name and short name. The short name is for internal use and will also serve as the coordinates grid name, if the user associates one to the new projection. The values of the projection parameters must be then given. The user may either associate to the new projection an existing grid, or create a new grid by selecting a distance unit (currently either metres or feet), by giving the values for the false easting and northing (for some projections these parameters are in fact the easting/northing of the false origin or of the projection centre), sensible bounds to the coordinates, and by choosing whether or not a fixed datum must be used with the grid. The bounds given will be used to check that the grid is not used outside its intended scope. All values of latitudes and longitudes must be given either in the datum of the grid if there is a fixed one, or in the datum being used for the map.

An user-defined grid cannot be forgot if it is currently associated to another projection or in use for displaying the map coordinates. Changing the definition of a user-defined grid may cause inconsistencies in previously projected data.

4.13.2 Predefined projections and grids

With the UTM/UPS (Universal Transverse Mercator/Universal Polar Stereographic) projection a single UTM zone is used, that of the first point displayed. Points in different zones will be projected into the same zone what may produce some deformation. There are no parameters that can be changed by the user.

The Transverse Mercator projection, also known as Gauss or Gauss-Kruegger projection is used with large scale maps and is not suitable for longitude ranges larger than about 6 degrees. It has 3 parameters: the latitude and longitude of the centre and the scale factor at the central meridian. The first two are computed as the averages of the latitudes/longitudes of the first points being mapped, while the third one has the default value of 0.9996 (used for UTM).

↓new
new↑

Particular cases of the Transverse Mercator projection are used in several maps usually for a certain country or region. Besides UTM, GPSMan predefines the following ones:

- the Austrian "Bundesmeldenetz" (BMN) projection. Parameters: central latitude `0`, central longitude in three zones of 3 degrees named `M28`, `M31` and `M34` and centred at `10.3333333`, `13.3333333` and `16.3333333E`, scale factor `1`. The datum to be used is called "Austrian (MGI)". Coordinates in the BMN grid have a false northing of 0 and a false easting that depends on the zone: 150km in zone `M28`, 460km in `M31`, and 750km in `M34`. ↓*new* *new*↑
- the British National Grid (BNG) projection. Parameters: central latitude `49`, central longitude `-2`, scale factor `0.9996012717`. The datum to be used is called "Ordnance Survey Great Britain". Coordinates in this grid correspond to a false easting of 400km and a false northing of 100km.
- the British West Indies projection. Parameters: central latitude `0`, central longitude `-62`, scale factor `0.9995`. The datum to be used should be based on the "Clarke 1880" ellipsoid. Coordinates in this grid correspond to a false easting of 400km. ↓*new*
- the projection used in the Italian Carta Tecnica Regionale (CTR). Parameters: central latitude `0`, central longitude in two zones of 6 degrees centred at `9` and `15E`, scale factor `0.9996`. Coordinates in the CTR grid have a false northing of 0 and a false easting that depends on the zone: 1500km in the first zone, and 2520km in the second one. *new*↑
- the German Grid projection (GKK: Gauss-Krueger-Koordinatensystem). Parameters: central latitude `0`, central longitude in zones of 6 degrees centred at `0`, `3`, `6`, `9`, `12`, and `15E`, scale factor `1`. Coordinates in the GKK grid have a false easting of $z \times 1000 + 500$ km, where z is the zone number.
- the Irish Transverse Mercator Grid (ITM) projection. Parameters: central latitude `53.5`, central longitude `-8`, scale factor `1.000035`. The datum to be used is called "Ireland 1965". Coordinates in this grid correspond to a false easting of 200km and a false northing of 250km.
- the Portuguese Military Maps projection, used in 1:25000 maps published by the Portuguese Army Geographic Institute. Parameters: central latitude `39.66666666666667`, central longitude `-8.13190611111111`, scale factor `1`. The datum to be used is called "Lisboa". Military coordinates in these maps correspond to a false easting of 200km and a false northing of 300km.
- the Swedish Grid (SEG) projection. Parameters: central latitude `0`, central longitude `15.808277777778`, scale factor `1`. Coordinates in this grid correspond to a false easting of 1500km.
- the Taiwan Grid projection (TWG). Parameters: central latitude `0`, central longitude in 6 zones of 2 degrees centred at `115`, `117`, ..., `125`, and scale factor `0.9999`. Coordinates in the TWG grid have a false easting of 250km. This grid is usually employed with either the "Hu-Tzu-Shan" datum (also known as "TWD67"), or the "TWD97" datum (whose definition could not be found for inclusion in GPSMan).
- the Uniform Finnish Grid (KKJY) projection. Parameters: central latitude `0`, central longitude `27`, scale factor `1`. Coordinates in this grid correspond to a false northing of 500km. There is a single zone named `27E`.
- the Basic Finnish Grid (KKJP) projection. Parameters: central latitude `0`, central longitude in zones of 6 degrees centred at `21`, `24`, `27`, and `30E`, scale factor `1`. Coordinates in the KKJP grid have a false easting of $z \times 1000 + 500$ km, where z is the zone number.

The Swiss Oblique Mercator projection is a particular case of an Oblique Mercator projection, which in turn differs from the Mercator and Transverse Mercator projections in that the central line with true scale is neither the equator (as in the Mercator), nor a meridian (as in the Transverse Mercator), and is chosen to suit the region to be mapped. In the Swiss Oblique Mercator this line has an azimuth of 90 degrees and contains the centre of the projection. There are three parameters: the latitude and longitude of the centre, and the scale factor, the default values being the averages of latitudes and of longitudes of the first points to be projected and 1, respectively.

The Swiss LV03 Grid projection is a particular case of the Swiss Oblique Mercator projection with centre at Bern, N46.9524055556, E7.43958333333 degrees in the "CH-1903" datum, and a scale factor of 1.

The Swiss LV03 grid has false easting and northing of 600km and 200km, and use the "CH-1903" datum.

The Lambert Conic Conformal projection has two variants: single standard parallel (under the name `Lambert Conic Conf 1` in GPSMan), and two standard parallels (called in GPSMan `Lambert Conic Conf 2`).

The former has 3 parameters: the latitude and longitude of the centre and the scale factor at the natural origin. The first two are computed as the averages of the latitudes/longitudes of the first points being mapped, while the third one has the default value of 1 (corresponding to a tangent cone; a value of less than 1 stands for a secant cone).

The latter has 4 parameters: latitudes of the two standard parallels (along which the cone intersects the geoid) and of the false origin, and longitude of the false origin. The first two default to the extremes of latitudes of the first points being mapped, and the position of the false origin defaults to the average of the positions of these points.

The Iceland Grid projection is a particular case of the Lambert Conic Conformal projection with 2 standard parallels at N64.75 and N64.25 degrees, a false origin at N65, W19 degrees and the "WGS 84" datum.

The Iceland grid has a false easting and a false northing of 500km. ↓*new*

The Lambert 93 grid projection is a particular case of the Lambert Conic Conformal projection with 2 standard parallels at N44 and N49 degrees, a false origin at N46.5 E3 degrees and a datum based on the "GRS 80" ellipsoid. The "WGS 84" datum can be used for applications with handheld GPS receivers.

The Lambert 93 grid has a false easting of 700km and a false northing of 6600km.

The Lambert NTF grid projection has 4 zones, named `I`, `II`, `III` and `IV`, each corresponding to a particular case of the Lambert Conic Conformal projection with 2 standard parallels with the following parameters (in degrees):

zone	**I**	**II**	**III**	**IV**
1st parallel	50.3959116667	45.8989188889	43.1992913889	41.5603877778
2nd parallel	48.5985227778	47.6960144444	44.9960938889	42.0000593542
lat f origin	49.5	46.8	44.1	42.165

the longitude of the false origin being that of the Paris meridian at E2.3372083333 degrees, and the "NTF (Nouvelle Triangulation de France)" datum.

The Lambert NTF grid has for zones `I` to `III` a false easting of 600km and a false northing of 200km, and for zone `IV` a false easting of 234.358m and false northing of 185861.369m.

The Lambert NTF *étendue* grid projection is the same as the projection for zone `II` of the Lambert NTF grid projection.

The Lambert NTF *étendue* grid has a false easting of 600km and a false northing of 2200km. *new*↑ *new*↓

Lambert Equal Area Conic projection is a conic, equal-area projection. It has four parameters: the latitude of a standard parallel, the polar aspect (either `north` or `south`), and the latitude and longitude of the centre.

The Albers Equal Area projection is a conic, equal-area projection. It has four parameters: the latitudes of the two standard parallels, and the latitude and longitude of the centre.

The Teale Albers grid projection is a particular case of the Albers Equal Area projection with standard parallels at 34 and 40.5 degrees North, and centre at the equator, 120 degrees West.

The Teale Albers grid uses this projection with the `NAD27 CONUS` datum, false easting of 0 and false northing of -4000km. Note that coordinate values may be negative. *new*↑

The Mercator projection can be defined as a Lambert Conic Conformal projection either with the equator as its single standard parallel, or with the two standard parallels at equal North and South latitudes (i.e., symmetrical with respect to the equator).

This leads to two variants: single standard parallel (named in GPSMan `Mercator 1`), and two standard parallels (named in GPSMan `Mercator 2`).

The former has 2 parameters: the longitude of the centre and the scale factor at the natural origin. They are taken as the average of the longitudes of the first points being mapped, and as 1, respectively.

The latter has 3 parameters: the latitudes of one of the two standard parallels and of the false origin, and longitude of the false origin. The first default to the maximum of the absolute values of the latitudes of the first points being mapped, the position of the false origin defaults to the average of the positions of these points.

The Stereographic projection is an azimuthal conformal projection used both for large scale and small scale mapping. There are 3 possible aspects: polar, oblique and equatorial, which are dealt with automatically by GPSMan. A particular case of this projection is the Universal Polar Stereographic (4.13.2) that is used in the UTM/UPS.

The Stereographic projection has three parameters: the latitude and the longitude of the centre (tangent point) and a scale factor. By default the scale factor is 1 and the coordinates of the centre are taken as the average of the latitudes of the first points to be mapped.

The Schreiber double projection is a variant of the Stereographic projection, in which each point in the ellipsoid is first projected in a sphere and the resulting point projected in a plane that intersects the sphere. This projection must be used with a datum based on the "Bessel 1841" ellipsoid, usually the "Rijks Driehoeksmeting" datum. If the given datum is for a different ellipsoid, GPSMan will change the datum to "Rijks Driehoeksmeting". All parameters for this projection are fixed: the centre is at N52 09 22.178 E5 23 15.5 in the "Rijks Driehoeksmeting" datum (coordinates of the Amersfoot OLV church), the scale factor is 0.9999079, and the constants for the conversions between the isometric latitudes in the ellipsoid and in the sphere are $n = 1.00047585668$ and $m = 0.003773953832$.

The Netherlands grid uses the Schreiber dual projection with a false easting of 155km and a false northing of 463km, and the "Rijks Driehoeksmeting" datum. Acceptable ranges of values are: 0–290000 for x, 290000–630000 for y, 50.3–53.45 for latitude, and 3–7.45 for longitude.

The Cassini-Soldner projection is a neither conformal nor equal-area projection used in the 19th century. It is still used for mapping areas with a small E-W extent. Scale is true along a central meridian and distortion increases significantly with distance from it. It has two parameters: the latitude and the longitude of the natural origin. These parameters are taken as the averages of the latitudes and longitudes of the first points being mapped.

The American Polyconic projection is also a neither conformal nor equal-area projection used before the computer era. It has a single parameter: the standard latitude, whose default value is taken as the average of the latitudes of the first points to be mapped.

4.14 Distances and bearings

There are two sets of formulae for computing distances and bearings that the user may choose

1. the so-called Law of Cosines for Spherical Trigonometry, that is not very accurate but is quite fast, and
2. the modified Rainsford's Method with Helmert's elliptical terms with a high degree of accuracy but slower; this method cannot be applied if one of the points is a geographical pole, in which case GPSMan applies the Law of Cosines.

Experiments with them seem to indicate that the differences to be expected are in the range of less than 100 metres in distances of more than 10 kms, and no differences in bearings when they are presented in degrees as integers.

Bearings in GPSMan are always geographic (True North).

4.15 Real-time logging

At present there are two variants of the implementation, that will probably be merged in the future. Any receiver sending data in the NMEA 0183 v2.0 standard format can be used with GPSMan and can use any of the two variants. This also applies to Garmin and Lowrance receivers. For the time being there is no support for real-time logging if the receiver is declared to be a Magellan.

Users of Lowrance receivers will want to use the variant for Lowrance. Users of Garmin-defined protocols, either the so-called Garmin protocol , or the Simple Text Output Protocol, should use the variant for Garmin.

Both variants implement some sort of simulator that can be helpful for tests and in getting acquainted with the interface before going to real-time usage.

GPSMan will work with the variant corresponding to the receiver brand selected in the options menu. After changing this option the program must be restarted because different code has to be loaded.

4.15.1 Variant for the Lowrance

This variant was designed and implemented by Brian Baulch (baulchb@hotkey.net.au) who has prepared a description of it that can be found in Appendix B.

The file `exerciser.tcl` used for simulation by this variant can be found in the `util` directory. It must be edited for configuration before use.

4.15.2 Variant for the Garmin

This variant implements:

- Garmin and NMEA 0183 protocols,
- recording of logging data that can be saved to file (at a certain moment, or continuously) and converted to a track,
- real-time animation in the map window (moving map),
- a travel/navigation interface, and
- a driving simulator and a random generator that produce fake logging data, to be used in training/testing.

Protocols supported by this variant are the following:

- Garmin PVT (position, velocity and time) Data Protocol, which is a part of the Garmin (`GRMN/GRMN` or `Garmin`) protocol,
- Garmin Simple Text Output Protocol (`Text Out`), and
- part of NMEA 0183 v2.0.

The receiver must be configured to use one of these protocols. On Garmin receivers this is done in the receiver's `Interface` display under `Setup`. Not all Garmin receivers support the first two, in which case NMEA 0183 should be selected and the variant for the Lowrance may also be used. If the selected protocol is not supported, either GPSMan knows about that and issues a warning, or there will be no information captured by GPSMan.

Facts that may help in choosing among the available protocols:

- Garmin PVT can be used along the rest of the Garmin protocol, meaning that getting and putting other information from/into the receiver can be done while real time logging is on; the receiver will temporarily stop sending logging information while these operation take place — this may be DANGEROUS if the logging information is crucially needed for navigation; it is the user's responsability not to initiate such operations in these conditions;
- both Simple Text and the implemented part of NMEA 0183 are one way protocols: information is only sent from the receiver to GPSMan; this means that it is not possible for GPSMan to check the connection with the receiver: GPSMan will be passively waiting for information to appear on the serial port;
- Simple Text carries less information than Garmin PVT (neither EPE, expected position error, nor EPV, expected position vertical error), while NMEA 0183 may carry more information than Garmin PVT; it is difficult to say more than this because there is no complete information on which NMEA sentences (commands) are sent by receivers.

Selecting the protocol in GPSMan is done in the GPSMan's receiver window using the `Protocol` menu, or the entry with this name of the `GPS Receiver` menu. Note that there is an option that defines the default protocol to be used. ▌↓*new*

If GPSMan was invoked by using the `start travel` command (5.8) the protocol will be either the default in the preferences file or the protocol given by the `-prot` command option. ▌*new*↑

Two other entries in these menus start simulators of logging data that will be helpful in getting acquainted with the interface before real-time usage:

- `simulator` generates random (and somewhat inconsistent) data;
- `driving simulator` provides an interface for the user to "drive" (starting from a waypoint) and in this way generate logging data.

A simulator will be switched off only when a different protocol is selected.

Controlling the real-time logging is done with the three buttons `Get Log` (or `Stop`), `Record` and `Animation` in the receiver window, or the corresponding entries of the menu under `GPS Receiver->Real-time track log`. The first starts and stops the input of logging information, the second launches a window that records that information, and the third starts the animation on the map. These buttons/entries can be actuated independently of each other, but it is obvious that the recording or the animation cannot start or go on if the input has not started or has been stopped. In this way the user may select when to record or when to have the animation. The recording window or the animation control window must be used to stop recording or the animation, respectively. ▌↓*new*

If GPSMan was invoked by using the `start travel` command (5.8) the animation is started unless the connection to the receiver could not be established. ▌*new*↑

The control buttons in the recording window and in the animation control window affect only the recording and the animation, respectively, except in what concerns the logging time interval which is the same for the recording and the animation. The minimum value for the time interval depends on the rate at which the receiver sends information. The initial value for it is 2 seconds. The recording window and the animation control window will appear only after the first valid logging information is received, and this means at least 2 seconds from the clicking on the buttons/entries. ▌↓*new*

If GPSMan is invoked by using the `start travel` command (5.8) the time interval can be given as the last argument to the command. ▌*new*↑

The recording window shows several columns with the logging information. These are, from left to right:

- number of the fix,
- local date and time,
- the latitude and the longitude (datum: "WGS 84"),
- altitude in metres (the existing option on user unit for altitude is *not* considered), ▌↓*new*
- quality of the position fix, ▌*new*↑
- EPE (estimated position error), EPH (expected position horizontal error), EPV (expected position vertical error) in metres,
- the 3 coordinates of the velocity vector in metres/second,
- TRK, the true track or bearing (CMG, course made good, track made good), true North.

Columns for which there is no information for the very first fix will be hidden automatically. The title of a column is a button that hides the column. This will be wanted for columns that are not being used or needed. The information in a hidden column is not lost and is updated. At any time a hidden column can be shown again by selecting its name from the `Show` menu.

The `Restart` button will destroy all the recorded information and restart recording.

The entries in the `Save` menu save as text the contents of the columns either in a one-shot fashion (`Existing log`), or writing the current contents and writing each new information when it comes up (`continuously`). In the latter case, the file will be closed when this menu entry is deselected or when real-time logging is stopped.

The text obtained with this menu cannot be re-loaded by GPSMan. To save the information in a format readable by GPSMan the `Make TR` button should be clicked to create a track, which can then be saved and loaded in the normal way.

The moving map works as the animation for a track described above (4.6). The main difference is in the scale that instead of setting the animation speed sets the logging interval.

For the time being there is no automatic loading of background images, a feature present in the variant for the Lowrance.

The travel/navigation interface provides

- two alternative displays, configurable by the user, to show real-time logging data and navigation information,
- navigation information (including warnings) to
  - record the current position and return to it (MOB, man over board), ▌↓*new*
  - create a waypoint whose coordinates are those of the last point in the real-time log; this fails if the waypoint edit window already exists, ▌*new*↑
  - go to a selected waypoint,
  - follow a selected route or track,
  - go back, following in reverse the real-time track since travelling was started;
- a `Change` menu (referred to as *the menu* in the remainder of this subsection) to control travelling/navigation and giving access to the receiver menu.

WARNING! use of this interface is at the sole risk of the user. In particular, the use of a laptop computer while driving alone is extremely dangerous. The user may also want to read the safety alert from the United States Coast Guard regarding the use of GPS equipment in boats.

A distinction is made here between *travelling* and *navigating* in the sense that the latter implies that there is a destination (when going to a waypoint or following a route) while the former does not. Starting a travel changes the map window top replacing the buttons by the travel/navigation displays. Stopping travelling restores the map window and will suspend (but not forget) current navigation objectives which will be resumed if travelling is again selected. ▌↓*new*

If GPSMan was invoked by using the `start travel` command (5.8) travelling is started unless the connection to the receiver could not be established. ▌*new*↑

The following information from the real-time log can be shown when travelling:

- local time,
- position (latitude and longitude, "WGS 84" datum),

– altitude in metres (the existing option on user unit for altitude is *not* considered), ↓*new*
– `TRK`, true track, current bearing (COG, course over ground, CMG, course made good, track made good), true North, *new*↑
– quality of the position fix,
– speed in user units,
– vertical speed in metres/second and up/down arrow indicators,
– type of receiver fix.

Other information only available when navigating:

– `CTS`, course to steer,
– `TRK/CTS` indicator, with an external compass ring (current track pointing up) and a CTS arrow,
– `TRN`, turn (difference between course to steer and true track),
– `TRN` arrow indicator,
– `To`, current destination (when following a route or track, next point in it),
– `From`, previous point in route or track,
– distance to current destination,
– `ETA`, expected time of arrival at current destination,
– `ETE`, expected time in route to current destination,
– `VMG`, velocity made good (velocity along line from starting/previous point to current destination), in user units,
– `XTK`, cross-track error (distance from current position to line from starting/previous point to current destination), in user units.

Navigation is started from the menu by selecting an objective which can be forgotten (navigation stops), suspended (navigation displays are frozen) and resumed.

Navigation to a point is made along a line of constant bearing with no drift (wind or stream effects) taken into account. This means that the transport velocity is assumed to be null and that CTS (the course to steer) is taken to be the bearing to the point. Computations use the so-called *plane-sailing* formulas that are commonly accepted as valid for distances up to 600 nautical miles [Gardner and Creelman, 1965], [Moore, 1964], [Macfarlane, 1963]. A point is arrived at when the distance to it is less than a user defined arrival distance.

Navigation along a route or track can be configured to be made:

– either exactly, meaning that each point must be arrived at before the next is selected as current goal; or approximately, in the sense that the current goal is abandoned in favour of the next when a certain criterion is met even if the point is not yet arrived at (see below);
– either starting from one end of the route or track, or from its nearest point (relative to the current position);
– either in the normal direction, or in reverse.

At any moment the user can force a change to the next goal from the travel menu.

Following a route or track in an approximate way is implemented by considering only the current position and the next two points (the current destination and the point after it if there is one) because of efficiency issues. This means that navigation will be blind to all other points in the route or track even when the current position is closer to any of them than to the next two points. The criterion for changing the goal has a looseness parameter

that can be changed from the menu. When a change to the next goal is about to be made a second CTS arrow (in a different colour) will be shown in the `TRK/CTS` indicator giving the bearing to it.

Tracks will be simplified to the maximum number of waypoints allowed in a route before being followed in an approximate way. In this case, numbers identifying track points will be preceded by an asterisk ("`*`") indicating that the numbers are different from those in the track.

Going back is implemented as following exactly the travelling log in reverse. This log is started when travelling begins and cleared only when restarting travelling, upon user confirmation. As a copy of it is used in going back, such a navigation will not be affected by clearing it.

Warnings will be issued as messages at the left upper corner of the map. Different colours are used for different priorities. User warnings can also be issued after configuration from the menu, and warnings (except those standing for danger) can be disabled without loosing the configuration. In case a warning is issued when another one is still being displayed it will either replace the previous one or be discarded depending on their relative priorities. There are the following classes of priority, in decreasing order:

	owner	usage
`important`	GPSMan	possibly dangerous situations
`high`	user	
`medium`	user	
`normal`	GPSMan	arrivals, errors
`low`	user	
`info`	GPSMan	other information

User warnings may be any of:

	when	needs
proximity to WP	approaching	WP, distance (user units)
	leaving (anchor)	WP, distance (user units)
speed	in excess	limit (user units)
vertical speed	out of range	minimun (may be <0), maximum (m/s)
turn	in excess	limit (degrees in -180 to 180)
cross-track error	in excess	limit (user units)

The configuration of the travel/navigation displays, warnings, arrival distance, and parameters for changing goal are kept on the saved state.

4.16 Miscellaneous

- changes in option values in some cases do not take effect immediately but only after GPSMan is restarted. Some care should be taken to avoid inconsistencies due to this. In particular, changes in the distance unit affect the possible values for the initial map scale, so that a change in the latter is normally needed in the next session after changing the former.
- anything that looks like a button normally *is* a button.
- closing a window from the window manager may cause data to be lost, and GPSMan may be unable to create it again.
- at any time only one waypoint, one route, one track, one polyline and one group may be open for editing; other such items may be viewed but not edited.

– when exiting from the program (`GPS Manager` button, or `ctrl-c` in the GPSMan windows), unsaved data will be lost unless the interface state is to be saved (see above (4.3)). As saved state files will be overwritten automatically by GPSMan it is a good idea to save important data to a file before quitting the program. ↓*new*

– operations that can take a long time are either subject to confirmation (if they can not be interrupted), or can be controlled from a slow operation dialog window if the option on this is selected. This window has an "Abort" button for cancelling the operation and a text box where all the error messages during the operation are collected. When the operation ends the window is closed automatically if there are no messages, otherwise it must be closed manually. *new*↑

4.17 GPSMan Files

GPSMan uses text files to store data. The `Load/Save` options in the menus deal with files in GPSMan format. The `Import/Export` options deal with files in foreign formats. In the GPStrans format (described in the documentation of GPStrans), all positions are exported in DDD format, although any available position format is accepted in imported files.

Files in GPSMan format can be either item information files (with data of different types: waypoints, routes, tracks, polylines and/or groups), or image information files (for saving information on background images for the map). ↓*new* *new*↑

These file formats are independent of the language used. That is, there will not be commands in Tobagonian even if a `lang*.tcl` file was provided for it and GPSMan was set to use that language.

4.17.1 Item information files

Item information files in GPSMan format (based on the GPStrans format) are as follows:

– lines whose first character is a `!` character are commands:

Format definition commands used to describe the format used thereafter; before the definition of waypoints, routes, polylines and tracks a position format and a datum must be given.

* `!Format: P T D`, where `P` is the position format (`DMS`, `DMM`, `DDD`, `GRA`,`UTM/UPS` or a coordinates grid name), `T` is time offset relative to UTC (a floating-point number between -12 and 12), and `D` is the datum name (to end of line). ↓*new* *new*↑
* `!Position: P`, where `P` is the position format (`DMS`, `DMM`, `DDD`, `GRA`,`UTM/UPS` or a coordinates grid name). ↓*new* *new*↑
* `!Datum: D`, where `D` is the datum name (to end of line).
* `!Creation: B`, where `B` (one of `yes` or `no`) states whether creation date fields are used.

Data commands used to start a data section:

* `!W:`, next lines (up to another data command or end of file) describe waypoints.
* `!R: N`, definition of route number `N`. After the number and a tabulation character, a comment will appear. After the comment a new tabulation may appear followed by fields (separated by tabulations) giving attribute-value pairs under the form `Attr=Val`. There are two possible attributes: `width` and `colour`, for the width in pixels and the colour of lines on the map window. After such a line there may appear a remark (see `!NB:` command below). Next lines (up to another data command or end of file) describe the route waypoints and the route stages if any. ↓*new* *new*↑

* `!T: N`, definition of track named `N`. After the name and a tabulation character, fields (separated by tabulations) may occur that have attribute-value pairs under the form `Attr=Val`. There are two possible attributes: `width` and `colour`, for the width in pixels and the colour of lines on the map window. After such a line there may appear a remark (see `!NB:` command below). Next lines (up to another data command, format definition command or end of file) describe the track points and the beginnings of segments (`!TS:` command). ↓*new* *new*↑
* `!L: N`, definition of polyline named `N`. After the name and a tabulation character, fields (separated by tabulations) may occur that have attribute-value pairs under the form `Attr=Val`. There are two possible attributes: `width` and `colour`, for the width in pixels and the colour of lines on the map window. After such a line there may appear a remark (see `!NB:` command below). Next lines (up to another data command, format definition command or end of file) describe the polyline points and the beginnings of segments (`!LS:` command). ↓*new* *new*↑
* `!G: M`, definition of a group named `M`. After such a line there may appear a remark (see `!NB:` command below). Next lines (up to another data command, format definition command or end of file) describe the group elements.
* `!NB: T`, text remark `T` for waypoint, route, track, line or group; must appear after a `!R:`, `!T:` or `!LN:` command, or after a line describing a waypoint. The text is terminated by a blank line. ↓*new* *new*↑

Ancillary commands used

* to describe route stages: `!RS:`.
* to define the type of group elements (see below): `!GW:`, `!GR:`, `!GT:`, `!GL:` and `!GG:`. ↓*new* *new*↑
* to begin a different segment of a track (except the first one): `!TS:`. ↓*new*
* to begin a different segment of a polyline (except the first one): `!LS:`. *new*↑

– lines describing waypoints (a `!W:` or `!R:` command appeared before) have a name, a comment, a creation date (but see the `!Creation:` command) and a position; all these fields are separated by tabulation characters. After these fields, in the same line and also separated by tabulations, there may be pairs under the form `Attr=Val`, where `Attr` is an attribute and `Val` the corresponding value; attributes currently in use, apart from those for hidden information: `alt` for altitude (either as a value in metres, or as a list with value in metres, value in external unit and external unit), `symbol` (possible values: GPSMan symbol names, see file `symbols.tcl`), and `dispopt` (possible values: GPSMan display option names, see file `symbols.tcl`). After such a line there may appear a remark (see `!NB:` command above). A route waypoint must be given in full, and not solely by its name as before version 6.1; there is a separate script in the distribution (directory `util`, file `wpsinfull.tcl`) to convert old files. ↓*new* *new*↑

– lines describing route stages (only one between two consecutive route waypoints) start by `!RS:` followed by a tabulation, a field with the comment, a tabulation, and a field with the label. Attribute-value pairs for hidden information may appear after a new tabulation and separated by tabulations. Empty stages should not appear.

– lines describing points in a track (a `!T:` command appeared before) have a tabulation character followed by a date, the position, the altitude and the depth (each one either as a value in metres, or as a list with value in metres, value in external unit and external unit), all fields being separated by tabulation characters. If the altitude and the depth are undefined both fields are omitted; if only the depth is undefined its field is omitted; otherwise the altitude field must be present and should be void if the altitude is undefined. GPSMan accepts track point positions in any available format, but will convert them to DMS. ↓*new* *new*↑

– lines describing points in a polyline (a `!L:` command appeared before) have a tabulation character followed by the position and the altitude (each one either as a value in metres,

or as a list with value in metres, value in external unit and external unit), all fields being separated by tabulation characters. The positions must all be given in the same datum and format (there can be no commands to change the format, position format or datum between the `!L:` command and the last point). *new↑*

- lines describing elements of a group (a `!G:` command appeared before) have a first field followed by a tabulation character followed by a name (up to end of line). The first field is either empty or of the form `!GW:`, `!GR:`, `!GT:`, `!GL:` or `!GG:` that stand for group waypoint, route, track, polyline and group, respectively, and describes the type of the element. If this field is empty the type is the same as that of the previous element. A group is assumed to be well-founded: it cannot be an element of itself even in an indirect way. *↓new* *new↑*
- positions given by latitude/longitude are given as two fields (each as a DMS, DMM, DDD or `GRA` coordinate); positions in UTM/UPS have four fields: East zone number, North zone letter, x- and y-coordinates; positions in other coordinates grids have three fields: zone (possibly empty), easting and northing. All fields are separated by a tabulation character. *↓new* *new↑*
- blank lines are ignored, except as terminators of remarks (see `!NB:` command above).
- file comments (ignored by GPSMan) start by a `%` character that can be preceded only by spaces and extend to the end of line.
- attribute-value pairs that describe hidden information are written as follows:
  - the attribute name starts with a capital letter that uniquely identifies the brand of the receiver (`G` for Garmin, `L` for Lowrance, `M` for Magellan); the rest of the name depends on the implementation but normally will describe the protocol and the data field;
  - the value is a string containing standard ASCII characters excluding all control characters (i.e., all codes must be ≥ 32 and < 127); the codification of the value is also implementation dependent (for an example, see the comments in `proc HiddenCode` in file `garmin.tcl`). *↓new*
  - for compatibility with older versions some attribute-value pairs no longer used for coding hidden information are still accepted when reading. *new↑*

4.17.2 Image information files

These are files containing the following information:

1. a `!Image: P` command, with `P` the absolute path of the file containing the image in an accepted graphics format;
2. a `!Datum: D` command, with `D` the datum name for the coordinates;
3. a `!Projection: NP As` command, with `NP` the name of the projection to use and `As` a sequence of attribute-value pairs under the form `Atr=Val` describing projection parameters; the tabulation is used as separator for `NP` and each pair;
4. a `!Transf: NT As` command, with `NT` the name of the coordinate transformation to use and `As` a sequence of attribute-value pairs as in the previous command;
5. a `!Scale: S` command, with `S` the floating-point value of the map scale in pixel/metre.

After this there may be one or more lines with a `!Image at: XG,YG P` command, where `P` is the absolute path of the file containing the image in an accepted graphics format, and `XG,YG` are the grid coordinates of the image. The grid coordinates of the first-loaded image are `0,0`. `GX` changes by 1 (-1) for each image to the right (left), and `GY` changes by 1 (-1) for each image down (up).

No newlines are allowed within these commands, and arguments are separated by spaces or tabulation characters unless otherwise stated. Paths must use the slash (`/`) as separator.

4.17.3 Map information files

These are files currently used only for saving the state of the map when there is no background image. They follow the same conventions as the image information files and contain the following commands:

1. a `!Map:` command;
2. a `!Datum: D` command, with `D` the datum name for the coordinates;
3. a `!Projection: NP As` command, with `NP` the name of the projection to use and `As` a sequence of attribute-value pairs under the form `Atr=Val` describing projection parameters; the tabulation is used as separator for `NP` and each pair;
4. a `!Transf: NT As` command, with `NT` the name of the coordinate transformation to use and `As` a sequence of attribute-value pairs as in the previous command;
5. `!Position: P`, where `P` is the position format (one of `DMS`, `DMM`, `DDD`, `GRA`, `UTM/UPS` or a coordinates grid name). ↓*new* *new*↑
6. a `!Scale: S` command, with `S` the floating-point value of the map scale in pixel/metre.

4.18 Files in Other Formats

GPSMan can import and/or export data in the following formats:

- BGA for importation of British Glider Association DOS turnpoints files as waypoints; there are three optional parameters that filter which turnpoints to import: feature (also called place) that must be given exactly as it occurs in the file, findability as one of the letters A to D or G, and air activity as a 1 or 2 characters string. There is also a separate utility (2.4) for converting BGA DOS files to GPSMan format. ↓*new* *new*↑
- EasyGPS export format; only for importation of waypoints;
- Fugawi export format; only for importation of waypoints;
- GD2; for importation only;
- GPStrans; ↓*new*
- GPX; for importation and exportation; *new*↑
- GTrackMaker; for importation only; all waypoints, routes and tracks will be read from the file, the other information being discarded; ↓*new*
- IGC; for importation of tracks only; each file corresponds to a single track; there is a parameter for selecting either GPS altitude or barometric (pressure) altitude; positions in 2D fixes are discarded if the latitude and longitude are 0 and there was no previously accepted fix, or if there is a change of more than 5 degrees either in latitude or longitude from the previous fix; a valid fix after a discarded one will be marked as a segment starter in the track; the track name will be that of the file without any extension unless the name is in use; the remark field indicates that barometric altitudes was used if that is the case, and the file name if it is not the track name; a datum number of 999 is taken to mean "WGS 84"; *new*↑
- MapGuide text (not XML!) export formats from 2002 or 03/04 versions. **Warning**: undetected errors may occur if the version given when opening the file does not correspond to that of the file. This format is only for importation of routes; each file corresponds to a single route that will be split in several ones according to the maximum number of waypoints in a route; a number or identifier for the (first) route can be given when opening the file, as well as, a comment and a remark; if the original route yields more than one route, a suitable remark will be added to each route after the first one; ↓*new* *new*↑

- MapSend;
- Meridian, only in the Magellan variant; ▌↓*new*
- NMEA real-time log; only in the Garmin variant, and only for for importation of tracks; each file is taken as a track; ▌*new*↑
- OziExplorer waypoints and tracks files; only for exportation of waypoints (not more than 1000 per file) and tracks (one per file); ▌↓*new* ▌*new*↑
- Shapefile if the `gpsmanshp` Tcl/Tk library is available; version 1.1 or later is required; a single data type is kept in a file as described in the `gpsmanshp` documentation; data can be stored in 2 or 3 dimensions and GPSMan needs the following parameters when reading/writing a file in this format: dimension, position format (that must consist of a single number for each coordinate), zone (for grids), datum, distance unit and altitude unit. **Warning**: use of a UTM zone with points belonging to another zone may produce conversion errors that are not reported. In GPSMan versions before 6.1 there were two different names for 2 and 3 dimensions, the position format, the datum and altitude unit were assumed to be decimal degrees (DDD), "WGS 84", and metre. The information on track or polyline segments is not saved to Shapefile files. With `gpsmanshp` versions 1.1 or later, routes, tracks and polylines can be read from Shapefile polylines or polygons. With version 1.2, items read in from a file not written by gpsmanshp will have in their remarks any fields of the .dbf file. There is also a separate utility (2.4) for splitting the polylines in a Shapefile into different GPSMan files according to their coordinates. ▌↓*new* ▌*new*↑ ▌↓*new* ▌*new*↑

4.19 GPSMan Symbols

GPSMan defines a set of symbols for waypoints that is described below under four categories (not mutually exclusive): general use, land, water, and aviation. This set is based on the symbols described in the "Garmin GPS Interface Specification" (Revision 3), but extends it, including, for instance, the symbols used by Lowrance receivers (contributed by Brian Baulch). The GIF files for these symbols provided in the distribution were produced expressly for use with GPSMan, with some by Brian Baulch and Robert Joop. It is recognized that both these images and the set of symbols can be improved and any help will be appreciated.

4.19.1 Category: General use

WP			
Danger	Skull	Bell	
Flag	Flag pin, blue	Flag pin, green	Flag pin, red
Trace-back	Dollar		
(transparent)	(void)		
Ball	Dot	Mark, x	Circled X
Diamond, green	Diamond, red		
Square, green	Square, red		
Box, blue	Box, green	Box, red	
Pin, blue	Pin, green	Pin, red	

4.19.2 Category: Land

First aid	Info		
City, small	City, medium	City, large	City, star
Car	Rent-a-car	Car repair	Tow truck
Biker			
WC	House	Building	
Pharmacy	Phone	Post-office	Police
Tunnel	Bridge	Dam	Levee
Mountains	Elevation	Summit	
Ladder	Trail head	Tracks	Many tracks
Deer	Duck	Fish	Fish bank
Tree	Parking	Lodging	Park
Castle	Monument	Church	Chapel
Cemetery	Museum	Theater	Casino
Zoo	Scenic	Airport	Mine
Oil field			
Food	Fast food	Mug	Pizza
Movie	School	Shopping	Store
Stadium	Amusement park	Beach	Swimming
Showers	Skiing	Golf	Bowling
Snow skiing	Ice skating		
Fitness	Picnic	Camp site	Drinking water
Geocache	Geocache found		
Recreational Vehicle park			
Fuel	Fuel & store	Horn	
Exit	Exit, no services	Exit no serv large	
Mile marker	Border	Toll	
Freeway	National highway	Highway	State highway
US highway			
Street intersection	Ramp intersection	Ramp int. large	
Truck stop	Weight station		
Parachute	Glider	Ultralight	
Tower, tall	Tower, short	Take-off	Landing
Geo name, land	Geo name, man-made	Geo name, water	
Civil location	Military location		

4.19.3 Category: Water

Anchor	Fuel		
Boat	Boat ramp	Fish	Fish bank
Light	Man over board	Beach	Swimming
Wreck	Dam	Mile marker	Radio beacon
Buoy, white	Buoy, amber	Buoy, black	Buoy, blue
Buoy, green	Buoy, green red	Buoy, green white	Buoy, orange
Buoy, red	Buoy, red green	Buoy, red white	Buoy, violet
Buoy, white	Buoy, white green	Buoy, white red	
Diver down 1	Diver down 2		

4.19.4 Category: Aviation

Airport	Heliport	Private field	
Seaplane base	Soft field	Landing	Take-off
Radio beacon	Danger (avn)		
1st approach fix	Localizer outer marker		
Missed approach point	ND beacon		
TACAN	VHF omni-range	VOR-DME	VOR/TACAN
Controlled Area	Restricted Area	Intersection	
Parachute	Glider	Ultralight	
Tower, tall	Tower, short		

Chapter 5

Using GPSMan in Command-line Mode

The command-line mode can only be used in operating systems supporting command line arguments, namely Unix/Linux systems. For the time being only Garmin receivers are supported in commands requiring a receiver.

In graphical mode GPSMan is launched by invoking `gpsman` or `gpsman.tcl` with either no arguments or a single argument standing for the serial device, while in command-line mode there will always be 2 or more arguments. This is used to distinguish between the two situations, although the graphical mode can be entered from command-line mode. ▌↓*new* ▌*new*↑

The general form of invocation is (in the usual Unix notation)

```
gpsman [OPTIONS] COMMAND [COMMAND_ARGS]
```
▌↓*new*

Some commands now accept parameters in the form `NAME=VALUE`. The term parameter when used here referring to arguments of a command has always this meaning. Care should be taken if the value part is a string with spaces (or other characters with special meaning for the shell or system command interpreter) in which case quotes or some other form of quoting should be used. ▌*new*↑

The possible options are:

`-dev DEVICE` gives the serial device path

`-log` creates a log file of the serial communication named `logfile` in the GPSMan user directory

`-rec Garmin | Lowrance | Magellan` changes the brand of the receiver (NOTE: at present will only work with `Garmin`); this option can be useful to override the options in the preferences file; cannot be used with the `readput` command, nor with the `-prefs` option, and cannot occur after the `-prot` option;

`-prefs PREFSFILE` gives the path for an alternative preferences file (read after the user preferences file); cannot be used with the `-rec` option

`-prot PROTOCOL` selects the protocol to be used overriding the default protocol given in the preferences file; cannot occur before the `-rec` option; the `show protocols` (5.3) command can be used to find out which protocols are available ▌↓*new* ▌*new*↑

Options not related with `COMMAND` will be silently ignored.

GPSMan exits with either a 0 if the command was successfully executed, or a 1 if not. Some commands will write information to the standard output channel. Error messages can possibly be written to the standard error channel, but most of the errors will not be explained. In case of doubt the graphical mode should be used to see if there are problems with files or the receiver. ▌↓*new*

Note that in command-line mode the saved state is not restored and the state is not saved unless the graphical mode is entered. ▌*new*↑

Available commands are:

`is available` exits with 0 (5.1)

`is connected` exits with 0 if a connection check with the receiver succeeds (5.2) ▌↓*new*

`show WHAT` writes help information to the standard output (5.3) ▌*new*↑

`haslib LIBRARY` checks availability of library (5.4) ▌↓*new*

`getwrite IN-TYPES FORMAT [OUT-PARAMS] [OUT-TYPES] PATH` transfers data from the receiver to a file (5.5)

`readput FORMAT [IN-PARAMS] [IN-TYPES] PATH [OUT-TYPES]` transfers data from a file to the receiver (5.6) ▌*new*↑

`getrtimelog PATH [INTERVAL]` gets the real-time log from the receiver and saves it to a file (5.7) ▌↓*new*

`start travel [INTERVAL]` launches the graphical mode and starts real-time logging, animation and travelling (5.8), unless the connection to the receiver could not be established ▌*new*↑

`translate IN-FORMAT [IN-PARAMS] [IN-TYPES] IN-PATH OUT-FORMAT [OUT-PARAMS] [OUT-TYPES] OUT-PATH` translates from a file format to another (5.9) ▌↓*new*

`project LATD LONGD DATUM PROJECTION [PARAMS]` writes to the standard output the projection of a point (5.10)

`georef IMGPATH TRANSF LATD LONGD LATD LONGD [LATD LONGD] DATUM X Y X Y [X Y] PROJECTION [PARAMS]` writes to the standard output a map information file (5.11)

`source PATH` executes a Tcl/Tk script (5.12)

`exec PATH` executes a GPSMan script (5.13) ▌*new*↑

5.1 The `is available` Command

This commands exits with 0 and is intended for a quick check on the availability of GPSMan in command line mode.

5.2 The `is connected` Command

The connection with the receiver is checked and the command exits with 0 if it succeeds. Only works with Garmin receivers. ▌↓*new*

5.3 The `show WHAT` Command

This command sends help information to the standard output according to its argument, one of: `datums`, `formats`, `help`, `projections`, `protocols`, `transfs`.

`show datums` prints the names of, and existing comments on the available datums.

`show formats` prints a table of the data file formats known to GPSMan and that can be used in other commands. An example is

```
EasyGPS:        Read, WP
Fugawi: Read, WP
GD2:    Read, WP RT TR
GPSMan: Read/Write, WP RT TR LN, Data
GPStrans:       Read/Write, WP RT TR
GPX:    Read/Write, WP RT TR, Data
GTrackMaker:    Read, WP RT TR, Data
IGC:    Read, TR
 Parameters
        Read:
                alt     gps
MapGuide:       Read, RT
 Parameters
        Read:
                version 2002
                idno
                commt
                rem
MapSend:        Read/Write, WP RT TR
Meridian:       File type invalid
NMEA:   Read, TR
Shapefile:      Read/Write, WP RT TR LN
 Parameters
        Read:
                dim     3
                pformat DDD
                zone
                datum   WGS 84
                dunit   m
                aunit   m
        Write:
                dim     3
                pformat DDD
                datum   WGS 84
                dunit   m
                aunit   m
```

For each known format either there is an indication that the format is invalid (because, for instance, a library as missing or there is incompatibility with the receiver brand), or there is information on the possible read/write operations and item types which will be followed by "Data"

if a file may contain items of different types. Parameters may appear followed by their default values if any.

`show help` prints the accepted commands and options.

`show projections` prints information on the available projections. For each projection there is the name to be used as argument to commands, a long name, and the list of accepted parameters. For each parameter the name to be used in a command, a type and a short explanation are given. An example is

```
Stereogr:       Stereographic
  Parameters
        ?datum
        lat0    (lat)   Lat of centre
        long0   (long)  Long of centre
        k0      (float>0)       Scale factor
```

meaning that the stereographic projection should be spelt `Stereogr` in commands, may have a `datum=NAME` parameter, and must have 3 other parameters under the names `lat0`, `long0` and `k0`, respectively a latitude, a longitude and a positive floating-point number, standing for the coordinates of the centre of the projection and for the scale factor.

`show protocols` prints a list of known protocols that can be used with the `-prot` option. A possible list is

```
Garmin (Garmin)
NMEA (Garmin)
simul (Garmin)
SText (Garmin)
```

showing at the end the receiver brand with which they must be used.

`show transfs` prints the list of available coordinates transformations with their names to be used in commands and longer names. *new*↑

5.4 The `haslib` Command

The `haslib` command checks whether a certain library is available:

```
haslib LIBRARY
```

where `LIBRARY` is either `gpsmanshp` or `Img`. The command exits with 0 if the library could be loaded.

5.5 The `getwrite` Command

The `getwrite` command gets information from the receiver and writes it to a file: ▌↓*new*

```
getwrite IN-TYPES FORMAT [OUT-PARAMS] [OUT-TYPES] PATH
```

where

- `IN-TYPES` are the data types to get; the possible types are `WP`, `RT` and `TR`; `all` can be used as an abbreviation of `WP RT TR`;
- `FORMAT [OUT-PARAMS]` is the output file format and its parameters (if any); use `show formats` (5.3) for a list of the currently accepted formats; ▌*new*↑
- `OUT-TYPES` may be absent in which case they are taken to be `IN-TYPES`; if `RT` occurs in `IN-TYPES` then `WP` can appear in `OUT-TYPES` but no other type "conversions" are allowed; some file formats impose that there is a single data type per file; `all` can be used as an abbreviation of `WP RT TR`;
- `PATH` is the path to the output file, or, in the case of the `Shapefile` format, the path to the Shapefiles basename (file extensions will be discarded). If the `PATH` is `stdout` output will be to the standard output unless the format is the `Shapefile` format (in which case the command fails). Existing files will be silently overwritten.

This command only works with Garmin receivers.

5.6 The `readput` Command

The `readput` command reads information from a file and sends it to the receiver: ▌↓*new*

```
readput FORMAT [IN-PARAMS] [IN-TYPES] PATH [OUT-TYPES]
```

where:

- `FORMAT [IN-PARAMS]` is the input file format and its parameters (if any); use `show formats` (5.3) for a list of the currently accepted formats; ▌*new*↑
- `IN-TYPES` are the data types to read; the possible types depend on the format and some formats impose that there is a single data type per file; they may be absent if the format requires a unique data type or if the files can have items of different types; `all` can be used as an abbreviation of `WP RT TR`;
- `PATH` is the path to the file to read from, or, in the case of the `Shapefile` format, the path to the Shapefiles basename (file extensions will be discarded). If the `PATH` is `stdin` input will be from the standard input unless the format is the `Shapefile` format (in which case the command fails);
- `OUT-TYPES` are the data types to put; the possible types are `WP`, `RT` and `TR` and may be absent in which case they are taken to be `IN-TYPES`; if `RT` occurs in `IN-TYPES` then `WP` can appear in `OUT-TYPES` but no other type "conversions" are allowed; `all` can be used as an abbreviation of `WP RT TR`.

As usual, putting information into the receiver can cause data stored in it to be overwritten.

This command only works with Garmin receivers.

5.7 The `getrtimelog` Command

The `getrtimelog` command gets the real-time log from the receiver and saves it to a file, until the process is killed:

```
getrtimelog PATH [INTERVAL]
```

where:

- `PATH` is the path to the output file, unless it is `stdout` in which case output will be to the standard output. Existing files will be silently overwritten.
- `INTERVAL` is the number of seconds between two consecutive entries in the log and defaults to 2.

This command only works with Garmin receivers.

5.8 The `start travel` Command

▌↓*new*

The `start travel` command launches the graphical interface and starts real-time logging, animation and travelling, unless the connection to the receiver could not be established

```
start travel [INTERVAL]
```

where:

- `INTERVAL` is the number of seconds between two consecutive entries in the log and defaults to 2.

This command only works with Garmin receivers. ▌*new*↑

5.9 The `translate` Command

The `translate` command reads information from a file in a given format and writes it to another file in another given format: ▌↓*new*

```
translate IN-FORMAT [IN-PARAMS] [IN-TYPES] IN-PATH \
          OUT-FORMAT [OUT-PARAMS] [OUT-TYPES] OUT-PATH
```

where:

- `IN-FORMAT [IN-PARAMS]` and `OUT-FORMAT [OUT-PARAMS]` are the input file format and the output file format and their parameters (if any); use `show formats` (5.3) for a list of the currently accepted formats; *new↑*
- `IN-TYPES` are the data types to read; the possible types depend on the input format and some formats impose that there is a single data type per file; they may be absent if the format requires a unique data type or if the files can have items of different types; `all` can be used as an abbreviation of `WP RT TR LN`; *↓new*
- `IN-PATH` is the path to the file to read from, or, in the case of the `Shapefile` format, the path to the Shapefiles basename (file extensions will be discarded). If the `IN-PATH` is `stdin` input will be from the standard input unless the format is the `Shapefile` format (in which case the command fails); *new↑*
- `OUT-TYPES` may be absent in which case they are taken to be `IN-TYPES`; if `RT` occurs in `IN-TYPES` then `WP` can appear in `OUT-TYPES` but no other type "conversions" are allowed; some file formats impose that there is a single data type per file; `all` can be used as an abbreviation of `WP RT TR LN`; *↓new*
- `OUT-PATH` is the path to the output file, or, in the case of the `Shapefile` format, the path to the Shapefiles basename (file extensions will be discarded). If the `OUT-PATH` is `stdout` output will be to the standard output unless the format is the `Shapefile` format (in which case the command fails). Existing files will be silently overwritten. *new↑* *↓new*

5.10 The `project` Command

The `project` command, that can have no options, writes to the standard output the projection of a point:

```
project LATD LONGD DATUM PROJECTION [PARAMS]
```

where:

- `LATD LONGD DATUM` are the latitude and longitude in signed decimal degrees and the datum of the point to project
- `PROJECTION [PARAMS]` defines the projection to use and its parameters; if no `datum=X` parameter is given the datum of the point is used; use `show projections` (5.3) for a list of the available projections and their parameters.

The result consists in two numbers corresponding to the x- and y-coordinates of the projected point. In the case of projections related to grids, except for UTM, no false easting or northing is added. Information on zones is not given.

5.11 The `georef` Command

The `georef` command, that can have no options, writes to the standard output a GPSMan map information file (4.17.3) containing geo-referencing information for an image:

```
georef IMGPATH TRANSF LATD LONGD LATD LONGD [LATD LONGD] DATUM \
       X Y X Y [X Y] PROJECTION [PARAMS]
```

where:

- `IMGPATH` is the path to the image file;
- `TRANSF` is the coordinates transformation to use; the command `show transfs` (5.3) produces a list of the available transformations;
- the `LATD LONGD` pairs and `DATUM` are the latitude and longitude in signed decimal degrees and the datum of the 2 or 3 control points whose pixel coordinates are the `X Y` pairs in the same order; the origin of the pixel coordinates is the top left corner of the image and the y-coordinates increase downwards;
- `PROJECTION [PARAMS]` defines the projection and its parameters; if no `datum=X` parameter is given the datum of the points is used; the command `show projections` (5.3) gives a list of the available projections and their parameters.

5.12 The `source` Command

The `source` command executes a `source` Tcl-command on the file whose path is given as argument:

```
source PATH
```

5.13 The `exec` Command

The `exec` command executes the GPSMan commands in the file whose path is given as argument:

```
exec PATH
```

Blank lines and lines whose first non-blank is "`#`" are discarded without further analysis. A "\" before the end of a line is taken as continuation line marker, but this may cause an error if there is an unmatched quote "`"`".

The following commands are available for use in GPSMan scripts:

`read FORMAT [PARAMS] [TYPES] PATH` read data from a file;

`write FORMAT [PARAMS] [TYPES] PATH` write data to a file;

`get TYPES` read data from receiver;

`put TYPES` write data to receiver.

An example of a GPSMan script file could be

```
  # WPs from the receiver to file in GPSMan format
getwrite WP GPSMan myWPs.gpsman
  # save them also in Shapefile format
write Shapefile dim=3 pformat=UTM datum="Datum 73" \
      shape/myWPs
```

new↑

Chapter 6

Support for Lowrance, Magellan and Garmin Receivers

6.1 Support for Lowrance receivers

Support for Lowrance receivers was developed by Brian Baulch (baulchb@hotkey.net.au) who has written a draft supplement for the present document that can be found in Appendix A.

6.2 Support for Magellan receivers

The support for Magellan receivers is a contribution of Matt Martin (matt.martin@ieee.org).

6.3 Support for Garmin receivers

Any Garmin receiver should (theoretically) connect with no problems to GPSMan. It must be set to use the Garmin protocol: in the receiver's `Interface` display, under `Setup`, the `Garmin/Garmin` or `Garmin` option must be selected. Alternatively, for real-time logging only, it can be set to use the NMEA 0183 protocol, by selecting the `NMEA` option. For the use of this protocol see the description of real-time support (4.15).

When using the Garmin protocol GPSMan may need to convert between bytes and floating point numbers. Tcl/Tk has no machine-independent way to do these conversions and GPSMan only implements them for little- or big-endian architectures that follow the IEEE floating point standard (this will cover most personal computers and workstations). Some Garmin receivers do not use protocols having floating point numbers and are not affected by this. In any case when connecting to the receiver GPSMan tests whether there are problems with the conversions, in which case the user is asked to confirm or cancel the operation.

GPSMan follows closely the "Garmin GPS Interface Specification", December 6 1999, 001-00063-00 Revision 3, available from the Garmin WWW site. This document is known to be outdated and not completely correct, as tests with recent models show.

Some data fields are not directly accessible to the user but are nevertheless kept by GPSMan as hidden information as described above. This is the case with the data on proximity distance, facility name, city, state, country code, and class.

GPSMan identifies the receiver model when connecting to it for the first time in a session. If the receiver implements the Protocol Capabilities protocol the list of protocols it uses is also obtained. This will probably be the case with the more recent models. Otherwise a table of protocols is looked up. At present there are entries in it for the receiver models in the list below.

When GPSMan gets a list of protocols for a receiver not yet listed in the table a file is created in the GPSMan user directory and the user is asked to send it to the author of GPSMan. This file should not be removed until the table is updated so that GPSMan knows these steps were already taken.

A problem was detected with a Garmin GPS 12Map that sends some packets twice. This caused havoc in the count of packets and made GPSMan to reset the connection. The solution has been to implement a test for repeated packets that are discarded if a flag is set.

eMap	
eTrex	_, Euro, Legend, Mariner, Summit, Venture, Vista
ForeTrex	
Geko	201, 301
GPS	5, 48, 65, 72, 75, 76, 89, 90, 125 Sounder, 126, 128
GPS 12	_, XL (Chinese, Japanese), CX, Map
GPS 38	_ (Chinese, Japanese)
GPS 40	_ (Chinese, Japanese)
GPS 45	_ (Chinese), XL
GPS 55	_, AVD
GPS 96	_, AVD, XL
GPS 120	_ (Chinese), XL
GPS II	_, Plus
GPS III	_, Pilot, Plus
GPSMAP	60C, 76, 76S, 135 Sounder, 175, 180, 195, 196, 205, 210, 220, 235 Sounder, 295
GPSMAP 130	_ (Chinese)
GPSMAP 230	_ (Chinese)
Rino	110, 120
StreetPilot	3
GPSCOM	170, 190

Acknowledgements

Thanks are due to:

- Alberto Morales (amd77@gulic.org) for contributing the support for Spanish and for his suggestions.
- Alessandro Palmas (alpalmas@tin.it) for his implementation of elevation graphs for routes and tracks, for providing continued support for Italian, for contributing the exportation of data in OziExplorer format, and for his help in debugging and his suggestions for improvements.
- Ashutosh Dutta (dutta@cs.columbia.edu) for testing the communication with the Garmin GPS 72 receiver and for his suggestions.
- Anders Lennartsson (anders.lennartsson@sto.foa.se) for permission to translate code in GPSTrans for the Swedish grid coordinates.
- Andreas Lange (Andreas.C.Lange@GMX.de) for providing continued support for German and the description of the GKK projection.
- Brian Baulch (baulchb@hotkey.net.au) for his work on adapting GPSMan for Lowrance receivers, for his ideas on improvements to GPSMan, for the wheelmouse support, real-time logging (variant for the Lowrance), and help on finding and destroying bugs.
- Brice-Olivier Demory (brice-olivier.demory@epfl.ch) who did the tests leading to the explicit support of the Garmin GPS 76 receiver.
- Carsten Tschach (tschach@zedat.fu-berlin.de) for making available GPStrans.
- Dan Jacobson (jidanni@yahoo.com.tw) for his help in debugging, his suggestions and the information on the Taiwan grid.
- David Kaplan (dmkaplan@ucdavis.edu) for contributing the RPM packages of GPSMan.
- David Wolfskill (david@catwhisker.org) who prepared a FreeBSD package of GPSMan, and helped with the use of the Img package.
- Edouard Lafargue who tested the communication with the Garmin eMap.
- Eduardo Veloso who kindly provided links to information on aviation symbols.
- Eric Spierings (ericsp@odont.uio.no) who sent screenshots of GPSMan at work.
- Frank Jordan (Universität Duisburg) for his help and patience in testing the communication with the Garmin eTrex receiver, for his suggestions on improvements to GPSMan and for his help in the process of making the GPSMan Debian package a part of the official Debian distribution.

- Frank Kujawski (Frank@Kujawski.org), for the tests with the eTrex Legend and the contribution of code for converting route information in MapsOnUS HTML format into GPSMan format.
- Harald Koenig (Universität Tübingen) for the tests of the communication over a USB serial port under Linux and his suggestions.
- Harald Stauss (harald.stauss@web.de) for his help in debugging, his suggestions, and for testing the communication with the Garmin eTrex Euro.
- Heiko Thede (Heiko.Thede@gmx.de) that contributed a shell and Tcl scripts to convert export files from Map&Guide 2002 and 2003/2004 text formats to GPSMan format.
- Herbert Tammer (H.E.Tammer@DNB.NL) for the tests with the GPSMAP 76.
- Jan Max Krueger (University of Konstanz) for making available sample IGC files and information on their use.
- Janne Sinkkonen (janne@iki.fi) for permission to translate code in GPStrans for the Finnish KKJY grid coordinates.
- Jean H. Theoret (ve2za@rac.ca) who contributed the code for changing the symbol of each waypoint in a group.
- Jim Wang who tested the communication with the Garmin GPSMAP 295.
- João Pedro Pedroso (LIACC, Universidade do Porto) for his help with the criteria to change goals when navigating a route in an approximate way.
- John Hay (jhay@icomtek.csir.co.za) for his suggestions and tests.
- John M. Quinn (U. S. Geological Survey) for making available the GEOMAG algorithm for estimating the magnetic declination.
- Jonathan Pennington (john@coastalgeology.org) who tested the communication with the Garmin GPS III+.
- José Paulo Leal and Luís Damas (LIACC, Universidade do Porto) who keep solving my problems with Tcl/Tk.
- Klaus Ethgen (Klaus@Ethgen.de) for his great help in testing the communication with the Garmin GPSMAP 76S, and in debugging the low-level communication with Garmin receivers, and for sending information on the Garmin GPSMAP 60C.
- Kyle Grieser (yuf@phoenixdsl.com) for all the work in testing the communication with the Garmin 12Map.
- Laurent Bonnaud (bonnaud@lis.inpg.fr) for sending a list of protocols and corrections on language support files.
- Luísa Bastos (Universidade do Porto), Gil Gonçalves (Universidade de Coimbra), José Alberto Gonçalves (Universidade do Porto) and Sérgio Cunha (Universidade do Porto) for their help with geodetic formulae and information.
- Marko Hyvärinen (mth@sun3.oulu.fi) for testing the communication with the Garmin 12CX.
- Martin Buck (m@rtin-buck.de) who contributed the resizing of 2D graphs, and a change to track edit window.
- Martin Ostermann (Aachen University of Technology) who contributed code for converting waypoint information in MapBlast HTML format into GPSMan format.

- Mathias Herberts (Mathias.Herberts@iroise.net) for the information on how to use GPSMan under MacOS X systems.
- Matt Martin (matt.martin@ieee.org) who implemented the support for Magellan receivers.
- Max Spring (mspring@cisco.com) for testing the communication with the Garmin eTrex Mariner.
- Niki Hammler (http://www.nobaq.net) who wrote a Perl script for reading waypoints data in Fugawi export format.
- Odilon Ferreira Jr (odilonf@estaminas.com.br), the author of GPS-TrackMaker, who kindly provided information on the file formats and datums used by his program.
- Paul Scorer (P.Scorer@leedsmet.ac.uk) who suggested the importation of FAI IGC data files and the drawing of climb rate graphs, kindly providing a noise-reduction filter for them, and contributed code for importing BGA DOS turnpoint files.
- Peter H. Dana (University of Texas) who provided help in correcting bugs in the conversion of UTM/UPS coordinates and information on map projections.
- Povl H. Pedersen (pope@my.terminal.dk) who corrected a bug in the implementation of the Garmin protocol.
- Rob Buitenhuis (rob@buitenhs.demon.nl) who contributed the support for Dutch, the definition of the Schreiber projection and of the Netherlands grid and helped in debugging.
- Robert Joop (rj@rainbow.in-berlin.de) who contributed new symbols, provided valuable suggestions and helped in debugging.
- Rogério Reis (LIACC, Universidade do Porto) for an algorithm for simplifying tracks, for his ideas on the interface functionality, help in debugging and his work on creating and maintaining a Linux Debian GPSMan package.
- Rolf Hatt (rolf@hatt.com) who tested the communication with the Garmin GPSMAP 180.
- Sabine Broda (LIACC, Universidade do Porto) who contributed the support for German since version 6.0.
- Stefan Hauser (etienne@imp.ch) who made the tests of the communication with the Garmin eMap 2.71 and for his suggestions.
- Stefan Heinen (Stefan.Heinen@synopsys.com) who tested the communication with the Garmin eTrex Summit, contributed new data structures for datums, the procedure to access them and changes to improve the focus policy and bindings under MS-Windos, as well as for his worked out suggestions.
- Thomas Trauber for testing the communication with the Garmin eTrex receiver.
- Tri Agus Prayitno (acuss@bk.or.id) who provided the support for Indonesian.
- Valère Robin (valere.robin@wanadoo.fr) who contributed the support for French and the importation of the EasyGPS export and GPX formats, the exportation in GPX format and for his suggestions.
- those who sent lists of protocols for Garmin receivers: John Matthews, Christoph Dworzak, Sabine Sagner-Weigl, Matthias M. Weber, David Klotz, Gerrit Huizenga, Hugo Trippaers, Eric D. Christensen, Gracjan Ziolek, Louis Mandelstam, Dan Hobner, Peter Van Loock, Aapo Rista, Chuck Cox, Alexander Damyanov, Joel Staker.

- Wolfgang Rupprecht, Tony Mollica, Andrey Semiuglov, Ron Thomas, John Madore, Kevin Geiss, Russell Senior, Urs Forster, Scot E. Wilcoxon, David Fletcher, Dragan Milicic, Anto Veldre, Slaven Rezic, Ronaldo Reis Jr., Tomasz R. Surmacz, Andreas Hünnebeck, Kenneth Ingham, Gianluca Interbartolo, Victor Yip, Siegfried Leisen, Christian Hubbauer, David Klotz, Paulo Quaresma, Christian Potthoff, Ron Schmars, Paul Makepeace, Louis Mandelstam, Dan Hobner, Chuck Cox, Paolo Cavallini, Meinolf Braeutigam, Pascal Brisset, Stefan Nickl, Joakim Majander, Graham Meadows, Frank Mohr, John Francis Lee, Jens C. Rasmussen, Balazs Lengyak for their help in detecting and tracking down bugs and/or for their suggestions.

The work presented here has been partially supported by funds granted to LIACC through the Programa de Financiamento Plurianual, Fundação para a Ciência e Tecnologia and Programa POSI.

Bibliography

[Gardner and Creelman, 1965] A. C. Gardner and W. G. Creelman, *Navigation*. Pergamon Press, 1965.

[Heckbert and Garland, 1997] Paul S. Heckbert and Michael Garland, Survey of Polygonal Surface Simplification Algorithms. Technical report (draft), School of Computer Science, Carnegie Mellon University, 1997.

[Li, 1995] Zhilin Li, An examination of algorithms for the detection of critical points on digital cartographic lines. *The Cartographic Journal*, 32, 121–125, 1995.

[Macfarlane, 1963] William Macfarlane, *Home Trade Navigation Guide*. Brown, Son and Ferguson, 1963.

[Moore, 1964] D. A. Moore, *Basic Principles of Navigation*. Kandy Publications, 1964.

Appendix A

Lowrance supplement to the GPSMan Documentation

```
Lowrance supplement to the GPSMan Documentation.
************************************************

1)  GPS receiver setup.

Follow the instructions given in your Lowrance manual. Set the
Com Port to 19200 bps, 8 data bits and no parity. Use the
correct Lowrance accessory data cable for your particular unit.

2)  Getting Waypoints from the GPS unit.

The Lowrance GlobalNav 212 receiver stores up to 999 Waypoints
internally. GPSMan downloads all 999 whether valid or not. The
indices of invalid (Unallocated) Waypoints are listed by
GPSMan and unused index numbers allocated when new Routes are
made or new Waypoints are created by GPSMan. For this reason all
Waypoints and Routes are read into buffers on initialisation of
the serial interface. This read operation can take nearly two
minutes at 19200 baud, please be patient.

The buffer mentioned above is not read into GPSMan memory until
the "Get WPoint" and "Get Route" buttons in the GPSMan "GPS
Receiver" window are clicked. This should be done before creating
any Waypoints or Routes with GPSMan, all Waypoints and/or Routes
should then be saved to the receiver before exiting GPSMan.

3)  Waypoint Names.

GPSMan is now able to handle Waypoint names containing spaces, so
spaces are no longer automatically deleted.

Note that the ASCII characters ".", "'", "(", "/", ")" and "-" can
also occur in Lowrance along with the ASCII space character (" ").
```

4) Lowrance Trails.

The terms "trail" and "track" are used interchangeably by GPSMan.

5) Time Offset.

GPSMan for Lowrance does not use the "Time Offset" setting under
the options menu. However it is recommended that this variable
be correctly set, for compatibility reasons. All times are displayed
in local time, not UTC.

This program uses the Lowrance LSI 100 interface protocol rev 1.1.
Copies of this protocol are available from www.lowrance.com.

Feedback welcomed.

Appendix B

Support for real-time logging (variant for the Lowrance)

```
*** WARNING! ***

Use of a laptop computer by the driver of a moving vehicle incurs risk!

Use this program at your own peril!

Do not attempt keyboard or mouse input whilst moving!

**************************
Welcome to GPSMan-autoMapic.
**************************

GPSMan-autoMapic is beta software designed to give moving-map real-time
plotting. It is not receiver-specific, and should work with any GPS receiver
that has the ability to output a standard NMEA 0183 v2.0 "GGA" sentence.
It has been developed on a ThinkPad 380 (150Mhz Pentium) using a Lowrance
GlobalNav 212 receiver.

GPSMan-autoMapic has been tested with both the Auslig RASTER250K map series
(150 dpi, original margins cropped, map sliced into three sheets 1 degree
latitude by 0.5 degree longitude by the author) and with a4 scans (150 & 120
dpi) such as city street-maps.

It also has operated sessions in excess of 12 hours and 980 km travelled
without any operator input required.

* Track log files. *

The software writes a log file to disk (in the current directory)for each
map image loaded. These files have the name "mapname.trk" derived from the
"mapname.img" of the current image loaded, and are in GPSMan standard
format. If a track crosses a specific mapsheet more than once, such tracks
will be appended to the original file. Each such track will have a unique
```

name "mapname-n" wher n is a number 0 to 999. With the "Manual plot" function the track name is user selectable, defaulting to "NMEA1".

*** Warning ***

The performance of this software is dependent on computer speed!
GPS Receivers that output a full string of NMEA sentences, without the ability to turn off those not required, may cause buffer-overflow when using slower computers. This is a Tcl feature and beyond my control at the moment. This bug may limit the size of map images that can be loaded. For example, an a3 image appears to be the limit with all sentences turned on in the Lowrance receiver & using a ThinkPad 380. A more powerful machine will handle larger images, test with "exerciser", details are given below. Feedback to baulchb@hotkey.net.au will help us in deciding on future development.

Use of the software is straightforward. The Map images required should have been prepared and georeferenced in the normal GPSMan fashion. All NMEA sentences are to the WGS 84 datum, so set GPSMan to WGS 84 in the options menu for georeferencing. However the georeferencing waypoints should of course be to the same datum as the mapsheet being referenced.

When this has been done an ".aut" file has to be prepared to show the bounding box for each georeferenced sheet needed. This file has to be manually prepared and has five or six tab-separated fields in each record, one record per line. The fields required are -

Image-file path. (The full path of the .img file created during
    georeferencing)
Latitude of the bottom of the image quadrangle, signed DDD format.
Latitide of the top of the image quad, likewise.
Longitude of the left side.
Longitude of the right side.
Optional image name or number. Can include any ASCII character including space.

There is no header required or permitted.

A simple example file (example.aut) -

```
~/Images/SE5401.img     -17     -16     138     139.5   CHARTERS TOWERS
~/Images/SE5402.img     -17     -16     139.5   141     MOUNT ISA
```

For portability of recorded logs, the image filename may need to be upper-case only. Check your GPS receiver specifications. However I have not yet ever needed to re-load a recorded track into the GPS. GPSMan capability is fully adequate, and can handle lower-case.

The images must be specified in degrees. If working with UTM or national grids, convert the co-ordinates to DDD positions with GPSMan.

Adjacent images can overlap, in fact this is preferable. If a point falls into a space between adjacent images a warning will be posted by GPSMan, the same warning will be posted if no .aut file is loaded or the position

"falls off the edge of the world". As soon as the position falls within an image's bounds again then that image will be loaded. Tracks recorded with no image currently loaded will be saved in "Blank.trk" as "Blank-n" (n 0 to 999 as above).

The plotting function is started from the "GPS receiver/Real-time track log/ Get Log" menu of GPSMan. Use "Lowrance" protocol not NMEA. From the "Start log" button a window will appear from which the logging interval can be set and the *.aut file loaded. The plotting interval cannot be changed, all points received will be plotted. To stop plotting/logging select "GPS Receiver/Real-time track log/STOP".

"Interval" defaults to 10 seconds. At this setting the 12 hour log mentioned earlier would take approx 220 kb of disk space. If disk space is at a premium increase the value of "Interval" accordingly.

If NMEA logging is all that is required, use the "GPSReceiver/Real-time track log/Start/Manual plot" menu to start the software. A map image can be preloaded but need not be. If no image is loaded the map scale can be changed whilst plotting.

Manual loading is a useful way of using slow machines, but introduces safety and convenience issues for single-manned vehicles. The track is automatically saved to disk (default name NMEA1.trk) when logging is stopped. This trackname can be changed from the setup window before starting the log.

The "exerciser.tcl" test sentence generator.
****

(Note - some Lowrance/Eagle models include a "Simulator" which is preferable to the "exerciser". Follow the instructions in the Lowrance manual.)

This program can be used for stationary testing of the autoMapic function. It requires the use of another computer and a null-modem cable or adaptor (e.g. a breakout box). Exerciser.tcl will send a series of NMEA sentences at preset intervals. Initial settings are controlled by the "set" statements at the top of the program, change with a text editor. An explanation is given below, but make the changes in the program, not here.

```
set SRLPORT /dev/ttyS0  # set serial port correctly.

set BaudRate 4800       # NMEA Standard.

set Hours "0"           # Do not change
set Minutes "0"         # ditto
set Seconds "0"         # ditto

# Set Interval to 1000 (1 sec.) for Lowrance (2000 for Garmin?)
set Interval 1000

# Latitude of the desired starting point.
set LatDeg 27
set LatMin 54.30
set LatSign S           # N or S as applicable.
```

```
# Longitude of the desired starting point.
set LongDeg 153
set LongMin 19.334
set LongSign E          # E or W as applicable.

# Size and direction of steps. The units are minutes.
set LatIncr 0.03
set LongIncr -0.05

# Change to 1 (true) to send the entire (Lowrance) series of sentences.
# 0 (false) sends only the required GGA sentence.
set SendDummies 0

To use the exerciser, first copy exerciser.tcl to the "dummy" computer which
must have Tcl/tk loaded. Start exerciser.tcl then start GPSMan on the
"Primary" computer. The two computers should have had serial ports already
connected with the null-modem cable.

Brian Baulch (baulchb@hotkey.net.au) 3 Apr 2002.
copyright (c) Brian Baulch 2000, 2002.
```

Appendix C

Map: Mouse and Keyboard Shortcuts

The tables of the following two sections summarise the availabe shortcuts for map operations. The last section explains the notation for events.

C.1 By Operation

scroll up (move map down) slowly	`<Key-Up>`
scroll up (move map down)	`<Button-4>`
scroll up (move map down) fast	`<Key-Delete>`, `<Shift-4>`
scroll down (move map up) slowly	`<Key-Down>`
scroll down (move map up)	`<Button-5>`
scroll down (move map up) fast	`<Key-space>`, `<Shift-5>`
scroll left (move map right) slowly	`<Key-Left>`
scroll left (move map right)	`<Alt-4>`
scroll left (move map right) fast	`<Control-4>`
scroll right (move map left) slowly	`<Key-Right>`
scroll right (move map left)	`<Alt-5>`
scroll right (move map left) fast	`<Control-5>`
scroll NE (move map SW) slowly	`<Shift-Up>`
scroll SE (move map NW) slowly	`<Shift-Right>`
scroll SW (move map NE) slowly	`<Shift-Down>`
scroll NW (move map SE) slowly	`<Shift-Left>`
panning slowly	`<Control-Motion>`
panning fast	`<B2-Motion>`
create waypoint	`<Button-1>`, `<Return>`
stop motion of waypoint (if one moving)	`<Button-3>`
open item (if over item)	`<Double-1>`
measure distance and azimuth	`<Shift-3>` (in two positions)
open waypoint menu (if over waypoint)	Unix: `<Control-1>` non-Unix: `<Button-3>`

add waypoint to route being edited on map (if any)	<Button-1>
delete waypoint from route being edited on map (if any)	<Shift-1>
edit previous stage of route being edited on map (if any)	<Control-3>
edit next stage of route being edited on map (if any)	<Control-Shift-3>
open route menu if editing it on the map	Unix: <Control-1> non-Unix: <Button-3>
finish edition of route on map	Unix: <Button-3> non-Unix: <Control-1>
cancel edition of route on map	<Shift-2>

C.2 By Event

<Key-Up>	scroll up (move map down) slowly
<Shift-Up>	scroll NE (move map SW) slowly
<Key-Down>	scroll down (move map up) slowly
<Shift-Down>	scroll SW (move map NE) slowly
<Key-Left>	scroll left (move map right) slowly
<Shift-Left>	scroll NW (move map SE) slowly
<Key-Right>	scroll right (move map left) slowly
<Shift-Right>	scroll SE (move map NW) slowly
<Key-Delete>	scroll up (move map down) fast
<Key-space>	scroll down (move map up) fast
<Return>	create waypoint
<Control-Motion>	panning slowly
<Button-1>	create waypoint, or add waypoint to route being edited on map (if any)
<Double-1>	open item (if over item)
<Control-1>	open waypoint menu (if over waypoint); otherwise Unix: open route menu if editing it on the map non-Unix: finish edition of route on map
<Shift-1>	delete waypoint from route being edited on map (if any)
<B2-Motion>	panning fast
<Shift-2>	cancel edition of route on map
<Button-3>	stop motion of waypoint (if one moving) Unix: finish edition of route on map non-Unix: open waypoint menu (if over waypoint); otherwise non-Unix: open route menu if editing it on the map
<Shift-3>	mark positions to measure distance and compute azimuth (not when loading an image or editing a route on map)
<Control-3>	edit previous stage of route being edited on map (if any)
<Control-Shift-3>	edit next stage of route being edited on map (if any)
<Button-4>	scroll up (move map down)
<Shift-4>	scroll up (move map down) fast
<Control-4>	scroll left (move map right) fast
<Alt-4>	scroll left (move map right)
<Button-5>	scroll down (move map up)
<Shift-5>	scroll down (move map up) fast
<Control-5>	scroll right (move map left) fast
<Alt-5>	scroll right (move map left)

C.3 Notation for Events

Mouse	
`<Button-`n`>`	mouse button: 1=left, 2=middle, 3=right $n > 3$ used for wheel-mouses
`<B`n`-Motion>`	mouse button n and mouse motion
`<Double-`n`>`	double click with mouse button n
Mouse and keyboard	
`<Shift-`n`>`	mouse button and shift key
`<Control-`n`>`	mouse button and control key
`<Control-Shift-`n`>`	mouse button, control and shift keys
`<Alt-`n`>`	mouse button and alt key
`<Control-Motion>`	control key and mouse motion
Keyboard	
`<Key-Up>`	up-arrow key
`<Shift-Up>`	shift and up-arrow keys
`<Key-Down>`	down-arrow key
`<Shift-Down>`	shift and down-arrow keys
`<Key-Left>`	left-arrow key
`<Shift-Left>`	shift and left-arrow keys
`<Key-Right>`	right-arrow key
`<Shift-Right>`	shift and right-arrow keys
`<Key-Delete>`	delete (or backspace) key
`<Key-space>`	space key
`<Return>`	return (or enter) key

Appendix D

Recent Changes

The following is a summary list of the more important changes made recently, no mention being made of bug corrections.

D.1 Changes After Version 6.1

D.1.1 Version 6.1.2

- the user manual explains how GPSMan can be run under the MacOS X system using a USB port; thanks to Mathias Herberts (Mathias.Herberts@iroise.net).
- new options:
  - on using window to control slow operations; in answer to a comment by Paul Scorer (p.scorer@leedsmet.ac.uk).
  - on whether polylines on the map should react to mouse events; in answer to a question from John Hay (jhay@icomtek.csir.co.za).
  - default receiver protocol (Garmin variant only); in answer to questions from users of the NMEA protocol.
- new command `start travel` that launches GPSMan in graphical mode turning on real-time logging, travelling and animation; after a question from John Hay.
- in travel mode (Garmin variant only), creating a waypoint whose coordinates are those of the last point in the real-time log; also in answer to a question from John Hay.
- new importation format: BGA (British Gliding Association) turnpoint (DOS) files, based on the `dos2gpsman` script by Paul Scorer (p.scorer@leedsmet.ac.uk); the script is also part of the GPSMan distribution.
- with `gpsmanshp` version 1.2, items read in from a Shapefile not written by `gpsmanshp` will have in their remarks any fields of the .dbf file; asked by John Hay.
- new separate utility (`shape2quadr.tcl`) for reading a lines DDD Shapefile and from a list of quadrangles produce GPSMan files one for the set of lines that belong to each quadrangle.

D.1.2 Version 6.1.1

- explicit support for the Garmin ForeTrex receiver, thanks to Aapo Rista (Aapo@Rista.net).

D.2 Version 6.1

- support for Spanish contributed by Alberto Morales (amd77@gulic.org).
- new options:
  - user unit for altitude (metres, feet); suggested by Chuck Cox (chuck413@sbcglobal.net).
  - map default projection, map default position format.
  - default line width for representing routes, tracks and polylines.
  - MapGuide text format default version.
- route waypoints in GPSMan data files must now be given in full, using names only is no longer supported; there is a new utility `wpsinfull.tcl` in the `util` directory to convert old files.
- polylines as a new data type, to be used as background for the map; suggested by David Kaplan (dmkaplan@ucdavis.edu).
- creation of a new waypoint at a given distance and bearing from an existing one; asked by Alessandro Palmas (alpalmas@tin.it).
- routes and tracks now have colours that can be changed.
- speed graph for tracks; suggested by Paulo Quaresma (pq@di.uevora.pt).
- climb rate graph for tracks; suggested by Paul Scorer (P.Scorer@leedsmet.ac.uk) who kindly provided a noise-reduction filter for the data.
- 2D plots can now be resized; code contributed by Martin Buck (m@rtin-buck.de).
- 2D elevation and speed graphs can now be plotted against time instead of total distance.
- the symbol of each waypoint in a group can be changed from the group window; code contributed by Jean H. Theoret (ve2za@rac.ca).
- computation of distance along line on the map, extending previous measurement between two points; suggested by Victor Yip (vgyip@ucdavis.edu).
- control window or confirmation for slow operations; suggested by David Kaplan.
- support for GPX export format, both for exportation and importation (including files produced by GPSBabel), contributed by Valère Robin (valere.robin@wanadoo.fr).
- exportation of waypoints and tracks in OziExplorer format; contributed by Alessandro Palmas.
- importation of the 2003/2004 version of MapGuide; from information sent by Heiko Theide (Heiko.Thede@gmx.de). There will be conversion errors if the version of the file is different from that given by the user.
- importation/exportation of MapGuide files no longer depends on the receiver brand; asked by Alessandro Palmas.

- importing NMEA log from file as a track (Garmin variant only); asked by Siegfried Leisen (lei@cray.com).
- importing/exporting Shapefile files in position formats other than decimal degrees for latitude/longitude and metres for altitude, and datums other than "WGS 84". With `gpsmanshp` version 1.1 it is now possible to read Shapefile polygons.
- importation of IGC (FAI GNSS) data files; asked by Paul Scorer.
- support for TFW (Tiff World File metadata format) files for geo-referencing images; asked by Paolo Cavallini (cavallini@faunalia.it).
- partial support for OziExplorer map files for geo-referencing images; asked by Eric Spierings (ericsp@odont.uio.no).
- support for grades (centesimal degrees) as signed floats.
- using the central/false longitude of a projection to fix the range of possible longitude values; this corrects problems with longitude values near E180 and have implications on how points and lines are displayed in small-scale maps.
- new projections:
  - CRT Italian projection and grid, from information sent by Alessandro Palmas.
  - Austrian projection and grid (Bundesmeldenetz); suggested by Alessandro Palmas.
  - Teale Albers projection and grid; asked by David Kaplan.
  - Albers Equal Area and Lambert Equal Area Conic projections.
  - French projections and grids; asked by Alessandro Palmas.
- new syntax for some old commands in command-line mode, and several new commands, including execution of GPSMan command scripts and Tcl/Tk scripts; partly in answer to questions from Joakim Majander (Joakim.Majander@enprima.com) and Pascal Brisset (pascal.brisset@free.fr).
- explicit support for Garmin receivers (with thanks to): Geko 201 (Christoph Dworzak, Sabine Sagner-Weigl, David Klotz), Geko 301 (Laurent Bonnaud), GPS5 (Matthias M. Weber), GPS12 Map (Gracjan Ziolek), StreetPilot 3 (Gerrit Huizenga), Garmin Rino 120 (Chuck Cox), Garmin Rino 110 (Alexander Damyanov), GPSMap60C (Klaus Ethgen).

D.3 Version 6.0.1

- importation of EasyGPS format, contributed by Valère Robin (valere.robin@wanadoo.fr).
- explicit support for the Garmin GPSMAP 196, thanks to John Matthews (john@matthews-net.org.uk) who sent the list of protocols.

D.4 Version 6.0

- support for Magellan receivers, contributed by Matt Martin (matt.martin@ieee.org).
- Indonesian language support, contributed by Tri Agus Prayitno (acuss@bk.or.id).
- perspective elevation graphs and enhanced side-view graphs of routes and tracks, contributed by Alessandro Palmas (alpalmas@tin.it).

- changes in the focus policy and bindings in non-Unix like systems, contributed by Stefan Heinen (Stefan.Heinen@synopsys.com).
- importation of files in Map&Guide and related software (Falk, Power Route) export format; based on scripts contributed by Heiko Thede (Heiko.Thede@gmx.de).
- a few new options: default baud rate, disable checking characters in names and comments, dafault printing command, size of font used in map labels, default widths for route and track lines on the map.
- new command line mode (no graphical interface) for GPSMan to be used by other applications; suggested by Curt Mills (hacker@tc.fluke.com) for use with Xastir.
- changes in the renaming of items with the same name (only applying to waypoints now) and in the way new names are generated; based on suggestions by Robert Joop (rj@rainbow.in-berlin.de).
- routes can now have non-numeric identifiers.
- routes and tracks have now a field for the width of the lines used to represent them on the map; suggested by Klaus Ethgen (Klaus@Ethgen.de).
- a track can now be simplified yielding a new track; suggested by Klaus Ethgen.
- segments in tracks; on a demand by Tomasz R. Surmacz (ts@wroc.apk.net).
- during the animation of a track, the current track point can be skipped; the scale for animation speed is now exponential; suggested by Slaven Rezic (rezic@onlineoffice.de).
- a group can be now forgotten with all its elements, easying the task of forgetting large numbers of items.
- hidden information in waypoints and tracks can now be displayed from their edit/show windows; suggested by Robert Joop.
- distances and azimuths can now be measured on the map; suggested by Victor Yip (vgyip@ucdavis.edu).
- new projections:
  - Schreiber double projection, and more accurate RD grid; implemented from a RDNAP document (available from `http://www.rdnap.nl`) in an English translation kindly provided by Rob Buitenhuis (rob@buitenhs.demon.nl).
  - Swiss Oblique Mercator and LV03; asked for by Andreas Hünnebeck (andreas@huennebeck-online.de).
- the Iceland (IcG) and the Netherlands (RDG) grids are now predefined.
- control waypoints can now be defined during geo-referencing.
- subsidiary map background images can be now be placed out of the grid; suggested by Stefan Heinen.
- real-time logging, Garmin variant:
  - travel/navigation interface.
  - continuously saving the real-time log to a file for use by other applications; suggested by Ashutosh Dutta (dutta@cs.columbia.edu).

- support for several new graphics formats if the `Img` library is available; suggested by David Wolfskill (davidmyatcatwhisker.org) and Stefan Hauser (etienne@imp.ch).
- the font size of labels on the map can now be changed; suggested by Ronaldo Reis Jr. (chrysopa@insecta.ufv.br)
- printing map and elevation graphs to a PS printer; suggested by Anto Veldre (anto.veldre@tllapt.ee)
- importation and/or exportation of data files in
  - GTrackMaker format; with the help of Odilon Ferreira Jr (odilonf@estaminas.com.br), on a suggestion by Ronaldo Reis Jr.
  - gd2 format; suggested by Kenneth Ingham (ingham@i-pi.com).
  - ESRI Shapefile format if the `gpsmanshp` library is available.
- new symbols added for geocache and geocache found; images for them and for border kindly provided by Robert Joop.
- better support for the Garmin Protocol Capability protocol, thanks to tests by Klaus Ethgen.
- explicit support for Garmin receivers: eMap version >2.7 (tests by Stefan Hauser), GPSMAP 76 (tests by Herbert Tammer, H.E.Tammer@DNB.NL), GPS 76 (tests by Brice-Olivier Demory, brice-olivier.demory@epfl.ch), GPS 72 (tests by Ashutosh Dutta), GPSMAP 76S (tests by Klaus Ethgen).

Index