
navit

Jan 18, 2020

1	Preparation	3
1.1	Get a supported map package	3
1.2	Install TTS	3
1.3	Connect GPS	3
2	Getting started	5
2.1	Setup proper startup-position	5
2.2	Setup GPS	5
2.3	Enable Map	5
3	Maps	7
3.1	OpenStreetMap	7
3.2	Binfile	10
4	Configuration	15
5	Configurable Sections	17
5.1	General	17
5.2	Display	19
5.3	Vehicle	21
5.4	Maps	23
5.5	Layout	23
5.6	Advanced	25
6	Changelog	29
6.1	Unreleased	29
6.2	[0.5.4] - 2020-01-18	30
6.3	v0.5.3 - 2018-08-31	32
6.4	v0.5.2 - 2018-08-30	32
6.5	v0.5.1 - 2018-04-19	35
6.6	v0.5.0 - 2015-12-31	52
6.7	[v0.5.0-rc2] - 2015-09-02	52
6.8	[v0.5.0-rc1] - 2015-08-08	52
7	Programming guidelines	53
7.1	Enforced guidelines via CircleCI	53
7.2	Coding Style	53

8	Commit guidelines	57
8.1	'Core' components changes	57
8.2	Commit per feature	57
8.3	Format of the commit log	57
9	Linux Development	59
9.1	Build dependencies	59
9.2	Taking care of dependencies	61
10	Android Development	65
10.1	Developing for Android	65
10.2	Testing an alternative build	68
11	MacOS Development	69
11.1	What you will need	69
11.2	Installation instruction	69
11.3	Speech	70
11.4	Using xcode	70
11.5	Something went wrong?	70
12	WinCE Development	71
12.1	Building using arm-mingw32ce	71
12.2	Remote Debugging	72
12.3	Building using arm-cegcc	72
12.4	Building Navit	75
13	Windows Development	77
13.1	Compiling using CMake	77
13.2	Compiling / debugging using CodeBlocks & mingw compiler	77
13.3	Compiling and running using cygwin	78
13.4	Compiling a native binary using mingw	80
13.5	TroubleShooting	81
13.6	Cross-Compiling win32 exe using Linux Ubuntu 14.04.1	81
13.7	Windows Mobile/Windows CE	83

This document is the official User Manual for [Navit](#), the OpenSource vector based navigation software.

Navit is a open source (GPL) car navigation system with routing engine.

It's modular design is capable of using vector maps of various formats for routing and rendering of the displayed map. It's even possible to use multiple maps at a time.

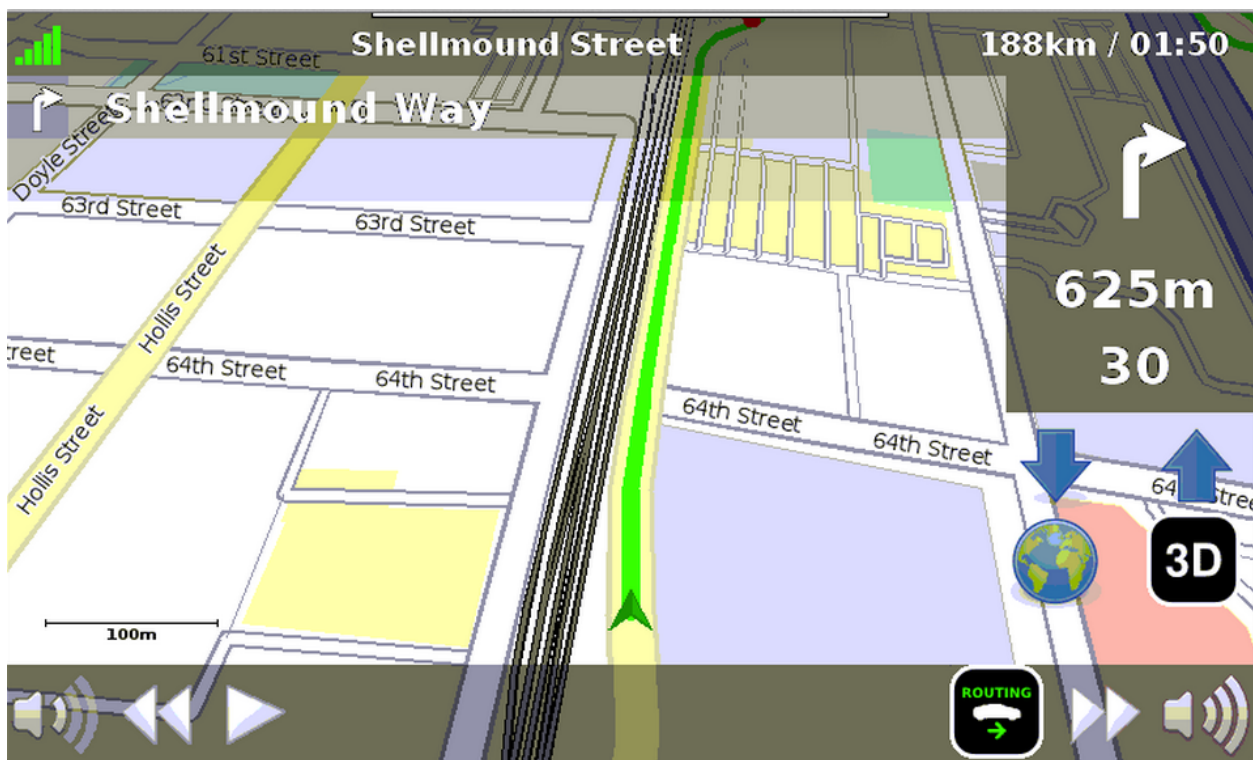
The user interfaces are designed to work well with touch screen displays. Points of Interest of various formats are displayed on the map.

The current vehicle position is either read from gpsd or directly from NMEA GPS sensors.

The routing engine not only calculates an optimal route to your destination, but also generates directions and even speaks to you.

Navit currently speaks over 70 languages!

Navit is highly customizable, from map layouts and on-screen display to the details of the routing engine.



There are much more options, see [\[\[configuration\]\]](#).

1.1 Get a supported map package

Navit can use different *Maps* formats, including free *OpenStreetMap* data. In order to use one of these maps, download a map of your desired area and store it into a local folder (such as `navit/maps` or `/usr/share/navit/maps`).

1.2 Install TTS

To get speech support, you need to install a text-to-speech tool such as **espeak**, **mbrola** or **festival**. These tools can be invoked from the command line. Test your setup by invoking the tools manually e.g.

```
espeak "This is a text!"
```

On some systems Navit comes with integrated espeak support, so you don't need to download it separately.

1.3 Connect GPS

Now connect your GPS. The exact procedure for this varies depending on the type of GPS device you are using and how you connect it to your computer. On Linux / Unix systems, your GPS should typically show up as a character device, i.e., you will find an entry in the `/dev` folder corresponding to your GPS device. Again, the file name depends on the type and connection method of your GPS receiver. See [\[\[Connecting a GPS receiver\]\]](#) for details.

Most GPS receiver will output the position in `[[NMEA]]` format, which can be used directly in Navit or using a GPS daemon program such as **gpsd**. A simple way to test whether your GPS receiver works and outputs NMEA data is to dump its output to a console. For example, if your GPS receiver can be found at `/dev/rfcomm0`, you can dump its output using

```
cat /dev/rfcomm0
```

The output should give you steady NMEA position updates. You can feed this information into **gpsd / xgps** or **QLandkarte** to see a graphical representation of your position and to check whether the output corresponds to your current position.

Currently Navit doesn't provide a graphical tool to change settings, so you have to do the changes manually using a texteditor. Please open your current **navit.xml** file and do the following steps:

2.1 Setup proper startup-position

On Navit's very first startup, it needs a **center** to look at on the map. By default this is set to Munich in Germany (at latitude 48.08 and longitude, which is conveniently covered by the sample map created on installation).

```
<navit center="4808 N 1134 E" />
```

See furthermore: *Configuration*

2.2 Setup GPS

Add the GPS connection

```
<vehicle name="My" enabled="yes" source="file:///dev/ttyS0" active="1"/>
```

See furthermore: *Configuration*

2.3 Enable Map

Just change the map entry corresponding to your local folders

```
<map type="binfile" enabled="yes" data="/var/navit/maps/uk.bin" />
```


Navit can use a variety of maps from a number of sources. This includes free and commercial material, that can be read using different map drivers which allow different sets of functionality. The visual design of maps can be adjusted with a layout.

The following matrix shows at-a-glance the features that each supported map type has.

Provider	Coverage	Age	Bin-file	Garmin	mg	poi_geodownload	textfile
OpenStreetMap	global	daily	Yes	Yes			
Garmin maps	global		Yes	Yes			
Marco Polo Grosser Reiseplaner	EU	> 4y					
Routeplaner Europa 2007	EU	> 4y					
Map + Route	EU	> 4y					

3.1 OpenStreetMap

OpenStreetMap <<http://www.openstreetmap.org/>> is a free editable map of the whole world. It is made by people like you. OpenStreetMap allows you to view, edit and use geographical data in a collaborative way from anywhere on Earth. Maps from OpenStreetMap can be used in Navit.

3.1.1 Quick Start

- Go to [Navit Planet Extractor](#)
- Zoom in and select the area that you are interested in. Use the 'Map Controls' on the right to switch between 'Navigation' and 'Select' modes. or use
- preselected areas:
 - [Germany](#)

- France
 - BeNeLux
 - Spain/Portugal
 - Italy
 - Entire planet
- Hit “Get Map!”
 - Move the downloaded map to the directory of your choice, and add it to the active the mapset (see [[Configuration]]) in navit.xml with a line similar to the following:

```
<mapset>
  <map type="binfile" enabled="yes" data="/path/to/your/map/osm_bbox.bin" />
</mapset>
```

3.1.2 Add OSM map to your mapset

Move the downloaded map to the directory of your choice, and add it to the active mapset (see [[Configuration]]) in navit.xml with a line similar to the following:

```
<mapset>
  <map type="binfile" enabled="yes" data="/path/to/your/map/my_downloaded_map.bin" />
</mapset>
```

3.1.3 Topographic Maps

Navit will display elevation/height lines but the required data is not included in most OSM derived maps.

Navit compatible maps with height lines can be created by feeding the output of *Phyghtmap* <<http://wiki.openstreetmap.org/wiki/Phyghtmap>> to Navit’s maptool. Alternatively the SRTM data can be downloaded in osm.xml format http://geoweb.hft-stuttgart.de/SRTM/srtm_as_osm/, avoiding the Phyghtmap step. The information can be either merged with OSM derived maps or used in a separate layer.

Many Garmin type maps such as http://www.wanderreiterkarte.de/garmin_de.php also have the height lines information but routing will not work with them.

3.1.4 Processing OSM Maps yourself

You can create your own Navit binfiles from OSM data very easily using [[maptool]], the conversion program which installs alongside Navit. “maptool” can process both OpenStreetMap XML Data files (*.osm files*) and *OpenStreetMap Protobuf Data files* (.pbf files) Follow these steps to process your own maps.

Download your own OSM data

OSM data can be downloaded from a variety of sources. OpenStreetMap XML Data files are regular textfiles, easily editable in any text editor. OpenStreetMap Protobuf Data files are binary files, which take up less space (so are quicker to download and process) but are not editable.

- OpenStreetMap XML Data

- Geofabrik provides pre-processed OpenStreetMap XML Data files of almost all countries, and all continents. This method is probably the easiest way of downloading OpenStreetMap XML Data for an entire country or continent. Note that the OSM files are bziped
 - planet.openstreetmap.org hosts the complete data set (the whole world). You can use [Osmosis](#) to cut it into smaller chunks.
 - [OpenStreetMap ReadOnly \(XAPI\)](#) The API allows to get the data of a specific bounding box, so that download managers can be used. For example: `wget -O map.osm "http://xapi.openstreetmap.org/api/0.6/map?bbox=11.4,48.7,11.6,48.9"`
 - [OpenStreetMap \(visual\)](#) allows you to select a small rectangular area and download the selection as OpenStreetMap XML Data.
- “OpenStreetMap Protobuf Data” * [<http://download.geofabrik.de/osm/> Geofabrik] provides pre-processed OpenStreetMap Protobuf Data files of almost all countries, and all continents.

Convert OSM data to Navit binfile

The following examples assume that you have installed Navit system-wide. If this is not the case, you will need to provide an absolute path to the “maptool” executable, which is in the `navit/maptool` folder.

Please also note, that maptool uses country multipolygon relations. So it’s a good idea to include the whole country boundary to your dataset. You can use the [josm](#) editor to download the country boundary relation and save it as osm file. Then this file can be concatenated with your sub-country level excerpt.

From .osm

```
cat my_OSM_map.osm | maptool my_Navit_map.bin
```

Or

```
maptool -i my_OSM_map.osm my_Navit_map.bin
```

Or for multiple OSM data files use the `<tt>-dedupe-ways</tt>` option to avoid duplication of way data if a way occurs multiple times in the OSM maps.

```
cat my_OSM_map1.osm my_OSM_map2.osm my_OSM_map3.osm | maptool --dedupe-ways my_Navit_
↪map.bin
```

From .bz2

```
bzcat my_OSM_map.osm.bz2 | maptool my_Navit_map.bin
```

From .pbf

```
maptool --protobuf -i my_OSM_map.osm.pbf my_Navit_map.bin
```

3.1.5 Processing the whole Planet

The OpenStreetMap wiki [<http://wiki.openstreetmap.org/index.php/Planet.osm> Planet.osm] page lists mirrors where Planet.osm can be downloaded. There are also downloads of smaller areas such as the UK and parts of Europe. These smaller excerpts are a lot quicker to download and process.

In case you want the whole planet.osm (24GB in December 2012), it is even possible to process planet.osm. It will take about 7 hours, requires > 1GB of main memory and about 30 GB disk space for result and temp files - planet.bin is currently (as of December 2012) 9.6GB:

```
bzcat planet.osm.bz2 | mapproc -6 my_Navit_map.bin
```

Please note -6 option (long name `-64bit`) used above. It should be used always if output bin file grows above 4GB, or generated file will not work at all. Using that option on smaller files slightly increases their size and makes them unreadable by some unzip versions.

3.1.6 Tips

- To enable a map you have downloaded refer to [\[\[OpenStreetMap#Adding_an_OSM_map_to_your_mapset|adding OSM map to navit.xml\]\]](#)
- If you don't see any map data in Navit (assuming your map is properly specified in navit.xml) using the Internal GUI click anywhere on the screen to bring up the menu. Click on "Actions" and then "Town". Type in the name of a town that should be within your map data. Select your town from the list that appears. This will bring up a sub-menu where you can click "View On Map". Note that if you have a GPS receiver you can also just wait till you get a satellite lock.
- To avoid changing navit.xml if you update your maps and the maps have different file names use the wildcard (*.bin) in your navit.xml file. For example:

```
<map type="binfile" enabled="yes" data="/media/mmc2/maps/*.bin"/>
```

3.2 Binfile

Navit has its own map format, called **binfile** format. It's a binary format, optimized for use with Navit (rendering, search, routing on embedded devices).

3.2.1 Dividing the world into tiles

The map coordinates of binfile are in meters, measured from equator and null meridian, with a merkatator projection. The limit of this world is about 20000 km in each direction from this null point. This doesn't cover polar regions, but it's ok for now.

So, the world gives a 40000x40000km rectangle (tile). This rectangle is divided into 4 equally-sized sub-rectangles (tiles) called a,b,c and d counter clockwise ...

- a is top right
- b top left
- c bottom right
- d bottom left

Each of the sub-rectangles (tiles) is then further subdivided, so a rectangle “aa” is top right in “a”

This is continued further up to 14 levels (but the number of levels might be variable). So in the end you get many tiles which are containing objects which are within this tile... And the tiles are also containing references to tiles which are within them.

A Navit binfile is actually a ZIP file; each tile is a member file in a zip file. So to extract an area of this file, you just read the zip file sequentially.

If you extract a small binfile using a zip tool, you will see member files adbdbbbbadcadc, adbdbbbbadcadc, Each file contains the data for one tile.

3.2.2 Projection

The coordinates are projected using a modified “Google” (i.e. EPSG900913) projection. Navit assumes an earth radius of 6371000m rather than the WGS84 value of 6378137m. Thus, for navit, the circumference of the earth is (2 x 20015087m).

3.2.3 Tile data format

A tile itself is a dynamic data structure.

As declared in `navit/map/binfile/binfile.c`, a tile is represented in memory by a struct :

```

struct tile {
    int *start;           /* Memory address of the whole data structure */
    int *end;            /* Memory address of first memory address not belonging to_
↪tile
                        * thus tile->end - tile->start represents the size of the_
↪tile
                        * in multiples of 4 Bytes */
    int *pos;           /* current position inside the tile */
    int *pos_coord_start; /* pointer to the first element inside the tile that is a
                        * coordinate. That is the first position after the header of_
↪a
                        * tile. The header holds 3 entries each 32bit wide integers.
                        * header[0] holds the size of the whole data-structure.
                        * header[1] holds the type of the item
                        * header[2] holds the size of the coordinates in the tile */
    int *pos_coord;     /* current position inside the coordinates region within the_
↪tile */
    int *pos_attr_start; /* pointer to the first attr //TODO explain attr format// data
                        * structure inside the tile's memory region */
    int *pos_attr;     /* current position inside the attr region */
    int *pos_next;     /* link to the next tile */
    int zipfile_num;
}

```

3.2.4 Content

Inside the binfile, each tile file contains a list of items. Each item is stored like this (everything is 4 bytes wide and always aligned):

```
{
  int: Length of the item (not including this length field) in integers
  int: Type of the item (from item_def.h)
  int: Length of the coordinate data which follows in integers
  {
    int: pairs of coordinates with consisting of 2 integers each
  } 0..n
  {
    int: length of the attribute (not including this length field) in integers
    int: Type of attribute (from attr_def.h)
    {
      int: Attribute data, depending on attribute type
    } 0..n
  } 0..n
}
```

3.2.5 Extracting a specific area

You can calculate the bounding box of the current tile.

Then there are two possibilities:

- The tile overlaps with the area you are interested in: Then simply copy the whole file data, including its header to the output, and add an entry to the directory which will be written later
- The tile doesn't overlap: Then don't drop that file, but instead write a file with size 0 and the same name to the output (I will explain later why this is needed), and add an entry to the directory

At some point you will have reached the end of the zip files, then you have to write the zip directory and the "end of directory" marker.

This will be very fast (you don't have to look into the zip files, which would mean decompressing and compressing them again) but has some disadvantages:

- You will have many empty files in it which are not really necessary. This is needed because the reference to sub-tiles are by number, and not by name (would be slow), and so the position of a tile within the zip file is not allowed to change
- You get some data you didn't want to have: this is because a tile which overlaps with your area of course doesn't contain only data from your wanted area, but from the area where it is located

3.2.6 How an object is placed in a tile

An object is placed inside of a tile using the following approach

- If the object can fit into one of the 4 top tiles it is placed in that tile
- The 4 sub-tiles are then checked to see if the object will fit inside of any of the four tiles that are contained inside of the sub-tile. If so, it is moved down a tile. This step is repeated until the object spans 2 or more tiles (or the lowest tile level is reached)
- If the object can't fit inside of any of the 4 top sub-tiles it is placed inside of the top-most tile

An object 'fits' inside of a tile if the coordinates of the object (min lat, min lon, max lat, max lon) lie inside of the coordinates of the tile (tile_min_lat, tile_min_lon, tile_max_lat, tile_max_lon)

Any object that cross the equator or the poles is placed in the top-most tile because it can not fit inside of any sub-tile.

Some important objects are placed into upper level tiles despite of their length to be easier reachable for routing or display purposes. This is done by specifying maximum tile name length for them in `item_order_by_type()` function in `navit/maptool/misc.c`.

BTW, “order” (zoom level) values used to query map and referred in `<itemgra>` and `route_depth` are equal to `(tile_name_length-4)`.

Navit is highly modular and customizable. This page aims to point out the most common options which a first-time user may want to change - power users may want to consult [\[\[Configuration/Full_list_of_options|the full list of options\]\]](#). It is also possible to edit the `navit.xml` file for your Android device under Windows and Linux (Debian/Ubuntu derivatives) with a third party application called [\[\[NavitConfigurator\]\]](#).

Setting up Navit is done by editing a configuration file called “**navit.xml**”. Editing XML configurations files in a text editor is simple, they are just plain text XML files, that can be edited with any editor. Just remember to “turn off ‘save UTF8 byte mark’ in Preferences” or navit may complain very much on the first byte of the file.
 The XML configuration file is splitted into sections within a hierarchy:

```
<config>
  <plugins></plugins>
  <navit>
    <osd></osd>
    <vehicle></vehicle>
    <vehicleprofile></vehicleprofile>
    <mapset></mapset>
    <layout></layout>
  </navit>
</config>
```

Navit comes **preshipped** with a default `navit.xml` together with `navit_layout_*.xml` files that are stored at various locations (depending on your system).

For Linux-like OSes:

- in `~/.navit/`: e.g. `/home/myusername/.navit/navit.xml` (This is probably to best place to customize your settings!)
- in `/usr/share/navit` or `/etc/navit`

Navit will apply settings in the following order:

- in the current directory (used on Windows)
- location supplied as first argument on the command line, e.g.: `navit /home/myusername/navittestconfig.xml` (Used mainly for development)

- in the current directory as `navit.xml.local` (Used mainly for development)

{{note|In any case, you have to **adapt settings** to your system!
 This includes especially GPS, map provider and vehicle: [[Basic configuration]]}}

Configurable Sections

5.1 General

Common options such as units, position, zoom and map orientation, ... be configured in this section.

5.1.1 General Options

Settings which affect Navit's general behaviour are located within the `<navit . . . >` tag. In a default installation, this is found on line 31 of `navit.xml`. By default, the `navit` tag is set as follows:

```
<navit center="4808 N 1134 E" zoom="256" tracking="1" orientation="-1" recent_dest="10  
↪">
```

Some common attributes are discussed below. For more advanced features, see [[Configuration/Full_list_of_options | the full list of options]].

Initial map position

On Navit's very first startup, it needs a **center** to look at on the map. By default this is set to Munich in Germany, which is conveniently covered by the sample map created on installation.

```
center="11.5666 48.1333"
```

Coordinates can be written in different formats; see [[Coordinate_format]] for the full list. To determine a specific latitude and longitude for your location you can use <http://itouchmap.com/latlong.html>. Usually, changing the "center" setting is not necessary, since it is only used during the first start. On subsequent starts, Navit will remember the last map position (stored in "center.txt") and ignore the "center" setting.

When Navit starts, it will display the map at a pre-defined **zoom**. The default zoom level is 256. The lower the value, the closer you will be zoomed in. <!--what does the zoom number express? How can I estimate it?-->

```
zoom="256"
```

For those using the [[SDL GUI]], a level of 128 is recommended.

```
zoom="128"
```

Note that once Navit has started, the zoom level can be altered using [[OSD]] or menu items.

Use the **orientation** attribute to orient the map in either the direction of travel, or oriented North. .To orient the map in the direction of travel:

```
orientation="-1"
```

or to orient North:

```
orientation="0"
```

Orienting the map North whilst in [[3D]] mode will provide visually confusing results, and is not recommended. When in 3D mode, it's best to have the map oriented in the direction of travel.

Autozoom

Navit has the ability to **autozoom** the map in or out dependent upon your speed.

```
autozoom_active="1"
```

To de-activate autozoom:

```
autozoom_active="0"
```

3D pitch

Navit has the capability to display either a 2D map (bird's eye perspective) or a [[3D]] map (some amount of tilt looking to the horizon). Navit's default configuration is to startup in the 2D perspective but it is possible to specify that Navit start with a 3D perspective. The amount of tilt is specified by setting the value of `<tt>pitch</tt>`.

The **pitch** value defines default camera tilting, with a value from 0 to 359. Note that usable values lie between 0 and 90 where 0 is bird's eye perspective looking down and 90 is human perspective looking forward. Also note that values closer to 90 will slow down map drawing, because the line of sight gets longer and longer and more objects are seen.

For example, the following added to the `<tt>navit</tt>` tag will force Navit to start with a pitch of 30 degrees:

```
pitch="30"
```

Imperial units

By default, Navit use the metric system of measurements when displaying or announcing distances, speeds etc. However, you can configure Navit to display and announce these values in imperial units. Simply add an `<tt>imperial</tt>` attribute to the Navit tag, and set its value to 1, as shown below:

```
imperial="1"
```

Speeds should now be displayed in units of miles-per-hour, whilst distances are converted to miles (large distances) and feet (small distances).

Default layout

When no specific layout has been specified by the user, navit uses a default layout to draw maps. The `<tt>default_layout</tt>` attribute of the navit tag allows to specify which layout to use as default;

```
default_layout="Car"
```

This string should match the `name` attribute of the required `layout` tag.

See [\[\[Configuration/Layout_Options| layout options\]\]](#) for more details. [\[\[Category:Customizing\]\]](#) [\[\[Category:Configuration\]\]](#)

5.2 Display

A large number of display properties can be configured, including desktop or touchscreen-optimised GUIs, on-screen display items and complete control over menu items.

5.2.1 Display Options

The Navit display is highly customisable and consists of the following components

- Graphics driver (appropriate engine for your system, to draw everything)
- GUI (enables user interaction and map display)
- OSD (shows widgets on map screen)

Graphics Driver

Different technologies can be used, to let Navit draw it's visual components. Not all might be available at your specific system

The current list of available graphics drivers:

- **android**, for the Android port
- **cocoa**, for the iPhone port
- **gtk_drawing_area**, usually most appropriate on Linux desktop systems
- **sdl**, render inside an X window, or direct to the Linux framebuffer, with min dependencies on external libraries.
- **qt5**, render using Qt5 library, either using QWidgets or QML. On top of any Qt5 supported display technology.
- **win32** - useable with `gtk` or `internal` GUIs for Windows systems only.

Experimental/less maintained drivers:

- **qt_qpainter**, render inside X window or on top of Qt Palmtop Environment.
- **opengl**, rendering via OpenGL
- **gtk_gl_ext**, rendering via OpenGL using GTK+ OpenGL extension
- **gd**, rendering using the GD Graphics Library

They can be activated and configured as following:

```
<graphics type="gtk_drawing_area" />
```

As mentioned, it's usually best to leave this as whatever the default is within your *navit.xml*, and only mess around with it if you know what you are doing, or have been told to by one of the developers.

Graphical User Interface

You can now choose which type of GUI you would like to use with Navit. Not all GUIs work with all Graphics drivers

GUI	an-droid	co-coa	gd	gtk_drawing_area	gtk_gl_ext	opengl	qt_qpainter	sdl	win32	qt5	pre-view
GTK											
Internal	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
QML2											

Generic GUI Options

There are some options available for the `gui` tag which are used by all the GUI types. These include:

- **fullscreen** - Enables Navit to start in fullscreen mode.
- **pitch** - The pitch value to pitch the map to when selecting 3D mode from the menus.
- **dimensions** - `w="1024" h="600"`

The following example uses the *internal* GUI, and starts Navit up in fullscreen mode, and will pitch the map to 35 degrees when 3D mode is selected from the menu. Note that to start Navit in 3D mode by default, `[[#Initial 3D pitch]change the pitch value in the navit tag]]`:

```
<gui type="internal" enabled="yes" fullscreen="1" pitch="35">
```

Internal GUI

The first GUI is embedded in Navit core and is primarily aimed at `[[touchscreen]]` devices, or those devices with small screens (such as netbooks). However, this GUI also works very well on desktops and laptops.

```
<gui type="internal" enabled="yes">
```

Options

A number of options specific to the *internal* GUI are available. These include:

- **font_size** - Base text size to use within the *internal* menu.
- **icon_xs** - The size that extra-small style icons should be scaled to (e.g. country flag on town search).
- **icon_s** - The size that small style icons should be scaled to (e.g. icons of *internal* GUI toolbar).
- **icon_l** - The size that large style icons should be scaled to (e.g. icons of *internal* GUI menu).
- **menu_on_map_click** - Toggles the ability to bring up the menu screen when clicking on the map. See the `[[Internal_Gui#Menu_Configuration|“internal“ GUI page]]` for more information.

An example `gui` tag using the previous options is shown below:

```
<gui type="internal" enabled="yes" font_size="250" icon_xs="48" icon_s="48" icon_l="64" ↵>
```

More options are discussed on the `[[Internal Gui]]` and the `[[Configuration/Full_list_of_options|full list of options]]`.

GTK GUI

The second GUI is called **gtk**, and is most useful for those users who wish to use a traditional windowed GUI. This is one useful to desktop use.

```
<gui type="gtk" enabled="yes" />
```

Options

A number of options specific to the `gtk` GUI are available. These include:

- `menubar` - enable/disable the menubar
- `toolbar` - enable/disable the toolbar
- `statusbar` - enable/disable the statusbar

```
<gui type="gtk" enabled="yes" menubar="1" toolbar="1" statusbar="1"/>
```

5.3 Vehicle

A number of vehicles can be defined within Navit, depending upon the device and/or operating system in use. Vehicle profiles for routing (eg: car, pedestrian, bicycle...) are also completely configurable.

5.3.1 Vehicle Options

It's important to understand the separate but linked Navit concepts of a **vehicle** and `[[Vehicleprofile|vehicleprofile]]` element. A vehicle defines the source of positional data (such as a USB GPS device), and how to present that data to the user on the map, where the vehicleprofile defines all aspects of routing.

A simple vehicle definition looks like this:

```
<vehicle name="My" enabled="yes" source="file:///dev/ttyS0" active="1"/>
```

Here some of the available options: * **active**: If set to 1, makes the vehicle the default active one. Routing, view centering and map redraw would be applied to this one by default. * **enabled**: If set to yes, Navit connects to the vehicle data source and shows the vehicle on the map. * **follow**: map follows after "n" gps updates (where n=0 means only when the vehicle leaves the map->saving CPU time) * **source** : source of GPS (required) * **profilename**: link the vehicleprofile for this vehicle.

<!-- do we need to keep this keys or note them elsewhere? Obsolete options: * **update**: This will force the map to be recentered at your main cursor's position. * **color/color2**: The color of the cursor is now specified within the cursor tag itself. * **animate**: If set to 1, the cursor will be animated as a moving dotted line, to make it easier to see. -->

GPS

Most essential detail is the `gps` source, that need to be adapted to your local setup.

Local:

Windows: * `source="serial:COM2 baud=4800 parity=N data=8 stop=1"` ** For Locosys GT31 (or BGT31 plugged by USB), settings are: `source="serial:COM3 baud=38400 parity=N data=8 stop=1"` where COM3 should be replaced by the correct COM over USB number.

Windows Mobile: * source="wince:GPD1:" - using the internal GPS driver, configured from Windows Remote GPS Settings. ** To use a BT GPS it must be configured as outgoing com port and paired, then selected as hardware port in GPS Settings. ** To start bluetooth on navit startup add bluetooth="yes". When exiting navit, the previous bluetooth state is recovered. * source="wince:COM1:" baudrate="57600"

There is useful [<http://w5.nuinternet.com/s660100031/SirfTech.htm>. SirfTech utility] that can automatically scan existing serial ports trying different baudrates to detect GPS source.

Linux: * source="file:/dev/rfcomm0" - BlueTooth GPS * source="file:/dev/ttyS0" - serial GPS connected to the first serial port (you may need to add the correct baudrate eg.: freerunner source="file:/dev/ttySAC1" baudrate="9600") * source="gpsd_dbus:" - via dbus, position reported by gpsd

Network based: If you want to connect multiple tools to your GPS, you need an multiplexer tool, as gpsd or gypsy. * source="gpsd://host[:port]" - gpsd://localhost, the default one, will try to connect to gpsd on localhost * source="socket:ipaddr:port" - socket connection (expects nmea stream) * source="socket:ipaddr:2947:r=1" - connect to gpsd in nmea mode (gpsd versions 2.39 or older) * source="socket:ipaddr:2947:?WATCH={“enable”:true,“nmea”:true};" - connect to gpsd in nmea mode (gpsd versions newer than 2.39) * source="gypsy://connectstring" - gypsy

Tricks: * source="file:/home/myhome/mynmea.log" : here, navit will replay the nmea logfile (under Windows it is currently not possible in Navit) * source="pipe:/usr/bin/gpspipe -r" - any executable that produces NMEA output - gpsbabel, gpspipe, ... * source="demo://": to use the demo vehicle. Set your Position and Destination, and vehicle will follow the calculated route. Useful if you have no nmea data source. * source="null://": no GPS at all

Logging tracks

To record your trip , you can add a sub-instance “log” to the vehicle. It is possible to add multiple logs.

```
<log type="gpx" data="track_%Y%m%d%i.gpx" flush_size="1048576" flush_time="900" />
<log type="nmea" data="track_%Y%m%d%i.nmea" flush_size="1048576" flush_time="900" />
```

This will give a log file named YearMonthDaySequencenumber.gpx/.nmea which will be kept in memory and flushed to disk when it is 1048576 bytes large or the oldest data is older than 900 seconds. <!--how to define, where the file get's stored?-->

To display your track for more than one hour, you must use [[binfile]] to create a cache file that get's display <!--do I need to add it as a map source?-->

```
<log type="binfile" data="track.bin" flush_size="0"/>
```

For customizing what is stored, see [[GPX]]

Vehicleprofile

[[Vehicleprofile | Profiles to add in the navit.xml]]
 <!-- this is tricky, here we need a step by step introduction--> Defines the behaviour of the routing and are usually linked to a vehicle section, so switching the “vehicle” (type of mobility) from within Navit, routing also will change its behaviour. This way, it is possible to include steps for pedestrian routing, but to exclude it for bike, horse or car routing. Within the vehicleprofile section, roadprofile sections are used to describe the routing behaviour of different roads. Here's a very basic example:

```
<vehicleprofile name="bike" flags="0x40000000" flags_forward_mask="0x40000000" flags_
  ↳reverse_mask="0x40000000" maxspeed_handling="1" route_mode="0">
  <roadprofile item_types="path,track_ground" speed="12" route_weight="5">
  </roadprofile>
  <roadprofile item_types="track_gravelled,track_paved,cycleway,street_service,street_
  ↳parking_lane,street_0,street_1_city,living_street,street_2_city,street_1_land,
  ↳street_2_land,street_3_city" speed="25" route_weight="15">
```

(continues on next page)

(continued from previous page)

```

</roadprofile>
<roadprofile item_types="roundabout" speed="20" route_weight="10"/>
<roadprofile item_types="ferry" speed="40" route_weight="40"/>
</vehicleprofile>

```

For details on the flags, see [\[\[Vehicle profile flags\]\]](#). The speeds are in km/h.

Only the vehicle profile names “car”, “bike” and “pedestrian” are translated in the GUI.

[\[\[Category:Customizing\]\]](#) [\[\[Category:Configuration\]\]](#)

5.4 Maps

You can use maps from a variety of sources, any number of maps can be configured and enabled at any one time.

5.5 Layout

Maps are displayed according to the rules defined in the layout. All aspects of the layout are configurable, from POI icons to colours for a particular type of highway.

For all versions shipped after nov 2018, layout XML configuration is stored in dedicated XML files called with the prefix “navit_layout_” (one file per layout definition).

5.5.1 Layout Options

Layouts

A layout defines how to **render a map**. Layouts are fully customisable, from the road colours and widths to size and type of icons to display for specific POIs. The layout is also where the **cursor** (i.e. the shape which shows where you are) is defined. A number of user-generated layouts and cursor definitions are available at [\[\[Layout\]\]](#).

Defining layers

A layout consist of one cursor and one or more **layers** which are each a set of rules on how and when to draw certain items. Those rules are called **itemgra**. The layers are rendered one by one in the order they appear in the XML config file, as are the items in each layer. If you can’t see an item make sure there is not another one hiding it. If your item is hidden, you can move your item further down in the layout section of the file.

```

<layout name="Demo layout" color="#ffefb7" font="Liberation Sans" active="1">
  <cursor w="26" h="26">
    <layer name="layer_1">
      <itemgra item_types="water_poly" order="0->
        <polygon color="#82c8ea" />
        <polyline color="#5096b8" />
      </itemgra>
    </layer>
  </layout>

```

Here the available options: * `item_types`: Comma separated list of items (see `navit/item.h` for definitions) * `order`: Range for zoom levels. * `speed_range`: Range for vehicle speed, useful for cursors. * `angle_range`: Range for pitch angle. * `sequence_range`: Useful for animated cursors.

For infos about map icons, see [\[\[Icons\]\]](#)

Overriding default (shipped) layouts

When the XML config file is parsed, layouts are taken in the order they come, and a layout with an already existing name overrides a previous definition. The default (shipped) `navit.xml` includes first system-wide `navit_layout_*.xml` files then the user-specific `navit_layout_*.xml` files, so the system-wide `navit_layout_*.xml` files can be overridden by adding a user-specific `navit-layout_*.xml` containing the same **name** attribute.

Copying the default **navit.xml** file to the user-specific location will still use the default shipped layout files, but copying one or several layout files as well to the user-specific location (and modifying them) allow to override these specific layouts.

Note on the default layout used by navit

When no layout has been specifically selected by the user (for example at first startup), navit will use the default layout specified (see [\[\[Configuration/General_Options#Default_layout|the related section to know how to configure this\]\]](#)).

Using a layer in multiple layouts

Sometimes, multiple layouts can use the same layer. For example, a reduced layout for a cleaner map may use the same layers as the regular layout, just not all of them.

To use a layer in multiple layouts, it can be referenced using the **ref** attribute. In place of the regular layer definition, use an empty tag with only the attributes “name” and “ref”:

```
<layer name="Found items" order="0-">
  <itemgra item_types="found_item">
    <circle color="#008080" radius="24" width="2" text_size="12"/>
  </itemgra>
</layer>
[...]
```

```
<layout name="Demo layout">
  [...]
  <layer name="Found items for demo layout" ref="Found items" />
  [...]
</layout>
```

```
<layout name="Demo layout reduced">
  [...]
  <layer name="Found items" ref="Found items" />
  [...]
</layout>
```

Note that the layer you want to reuse must be placed “outside” the layout. Layers defined inside a layout cannot be reused in this way.

[\[\[Category:Customizing\]\]](#) [\[\[Category:Configuration\]\]](#)

5.6 Advanced

There are many more options, including debugging, specific plugins, speech announcements, trip logging, ...

5.6.1 Advanced Options

The rest of this webpage is meant for advanced/power users who'd like to fiddle a little more under-the-hood. The average user can safely ignore this section!

Speech

Navit can announce driving directions with voice. Navit can use different mechanisms to play these announcements. Note that not all tools are available on all platforms.

Prerecorded samples

Navit is able to compose phrases if you give it a set of prerecorded **samples**. Configuration example:

```
<speech type="cmdline" data="aplay -q %s"
sample_dir="/path/to/sampledirt" sample_suffix=".wav" flags="1"
vocabulary_name="0" vocabulary_name_systematic="0" vocabulary_distances="0"/>
```

The directory “sample_dir” should contain audio files. “sample_suffix” is the common file type suffix of those files. The names of the files (without the suffix) must correspond to the text they contain. For each text it wants to speak, Navit will look for one or more sample files with corresponding names (ignoring upper/lower case). So for “turn right in 300 meters” you could use turn.wav, right.wav, in.wav, 300.wav, meters.wav. Navit will prefer files that contain multiple words: If file “turn right.wav” is present, it will be used even if you have turn.wav and right.wav.

Note that Navit internally handles all text in UTF-8 encoding. If you use a file system where file names are not encoded with UTF-8 (such as Windows), Navit will only find files for ASCII text. If you use a language that uses non-ASCII characters, the file name must be the [<http://en.wikipedia.org/wiki/Percent-encoding> percent encoding] of the UTF-8 representation of the text. For example the filename for “süd” would be “s%c3%bcd.wav” (because “ü” is encoded as C3BC in UTF-8). For this feature to work, you must set “flags” to 1.

“data” is the program that can be used to play the sample files. You should specify the program name along with any necessary parameters. The placeholder “%s” will be replaced with the file(s) to be played. All files required for a text will be passed in one go, so the program will need to support playing multiple files. Note that the %s should “not” be quoted; the text is not passed through a shell.

Note that if any file that is needed to compose the complete phrase is missing then Navit will be silent. In that case a warning will be printed. Unfortunately, there is no complete list of the samples required. However, all the navigation text is contained in the translation files (.po files), so you can get a rough list.

By default Navit is trying to announce street names. To disable this feature you can set “vocabulary_name” and “vocabulary_name_systematic” to 0 in the speech tag which will specify that the speech synthesizer isn’t capable of speaking names. Also there is “vocabulary_distances” which you can set to 0 so only the minimum set of 1,2,3,4,5,10,25,50,75,100,150,200,250,300,400,500,750 as numbers is used.

espeak

```
<speech type="cmdline" data="espeak -s 150 -v english_rp %s"/>
```

Will use espeak instead, for those who want Navit to speak to them in English, at 150 words per minute. The “%s” is filled in by Navit when sent to the speech synthesis software (with something like “Turn left” or whatever is appropriate at the time). If you need more features, you should use an external wrapper script which can contain anything supported by your shell (see [[Translations]]).

festival

flite

Mbrola

Android

```
<speech type="android" cps="15"/>
```

Start up in silent mode

To have Navit start up in silent mode, insert `<code>active="0"</code>` somewhere in your `<code>speech</code>` tag. For example on Android:

```
<speech type="android" cps="15" active="0"/>
```

In this case, you should place a `<code>toggle_announcer</code>` item in your [[OSD]] configuration, or add a menu item so you can enable speech output when you need it.

Splitting navit.xml

Navit has support for a small subset of **XInclude / XPath** for including parts of external XML files. Supported is a tag like

```
<xi:include href="some_file" xpointer="xpointer_stuff" />
```

You can leave out either href (xi:include refers to the same file it is in then) or xpointer (xi:include then refers the complete file), but not both. The “href” attribute refers to a file relative to the current directory. It is suggested to use the complete path, such as “~/home/root/.navit/navit-vehicles.xml”.

href is expanded with wordexp internally, so you can do stuff like:

```
<xi:include href="$NAVIT_SHAREDIR/maps/*.xml" />
```

Some examples on the supported syntax: .. code-block:: xml

```
<xi:include xpointer="xpointer(/config/navit/layout[@name='Car']/layer[@name='points'])" />
```

references to the XML-Tag “layer” with attribute “name” of value “points” within an XML-Tag “layout” with attribute “name” of value “Car” within an XML-Tag “navit” within an XML-Tag “config”.

```
<config xmlns:xi="http://www.w3.org/2001/XInclude">
<xi:include href="$NAVIT_SHAREDIR/navit.xml" xpointer="xpointer(/config/*[name(.)!=
↪'navit'])"/>
<navit center="4808 N 1134 E" zoom="256" tracking="1" cursor="1" orientation="0">
<xi:include href="$NAVIT_SHAREDIR/navit.xml" xpointer="xpointer(/config/navit/*[name(.
↪)!='vehicle'])"/>
```

(continues on next page)

(continued from previous page)

```
</navit>  
</config>
```

Use this as your `$HOME/.navit/navit.xml` and you will get everything under `<config>..</config>` except `<navit>..</navit>` (first `xi:include`), plus `<navit>` as specified plus everything from navit within config, except the vehicle definitions (second `xi:include`).

All notable changes to this project will be documented in this file.

Changes and documentation about Navit can be found in the wiki at: <http://wiki.navit-project.org>

A timeline of opened and closed issue tickets can be found at our trac instance: <http://trac.navit-project.org> and on our github project: <https://github.com/navit-gps/navit/issues>

Navit follows the semantic versioning:

- $x.y.z$ (patch): only bug fixes or refactoring, no changes in functionality
- $x.Y.z$ (minor): added or changed functionality but can be used as a drop-in replacement for the previous version (all data formats and interfaces are still supported); minor UI changes (such as moving individual menu items) are also allowed
- $X.Y.z$ (major): at least one of the following:
 - Major new functionality (such as Augmented Reality, inertial navigation or support for live traffic services): de-facto standard for end-user apps
 - New user interface (such as moving from the old pulldown menu UI to the Internal GUI): this is definitely the UI equivalent of a breaking API change
 - Dropped support for a data format or interface: also a breaking change and usually tends to occur along with larger changes which would warrant a new major version anyway

6.1 Unreleased

To get the list of all the unreleased commits, see: [Full Changelog](#)

6.2 [0.5.4] - 2020-01-18

6.2.1 Added

- Android: Speech: Do not pronounce hyphens during textual directions #732
- Add support to display OSM multipolygons #398
- core:feature:multipolygon support #824 (metalstrolch)
- Add/Android: Provide geo coordinates for clicked position in contextual menu #794 (lains)
- Add:Gui/Internal: Adding missing escape modes and fixing generic html escape mode #785 (lains)
- Refactor:GUI/GTK:Adding pointer and label when displaying POI on map #729 (lains)
- enhancement:maptool:slice big polygons into smaller parts #906 (metalstrolch)
- enhancement:maptool:improve processing speed and fix some memory holes #901 (metalstrolch)
- feature:graphics/android: add polygons with holes support #889 (metalstrolch)
- traffic:Improve startup performance #822 (mvglasow)
- feature:gui_internal: include post code in town serch #818 (metalstrolch)
- add:graphics:dpi scaling support #813 (metalstrolch)
- Add/Android Supporting sending & receiving geo: intents #812 (lains)
- enhancement:layout_car:use transparency to draw some map features #879 (metalstrolch)
- Add watch after reopen the gps file #872 (gefin)
- Add:graphics/windows: polygons with holes for windows CE #862 (metalstrolch)
- Add:graphics/sdl: Polygon with holes support. #861 (metalstrolch)
- Admin levels for additional countries #859 (mvglasow)
- countries:add:Kosovo #846 (metalstrolch)
- Feature:Support USERPROFILE folder for Windows target #758 (OLFDB)

6.2.2 Updated

- Update:Android:Navit logo #943 (lains)
- Update:Navit logo according to proposal in issue #436 #724 (lains)
- Update:i18n:Updated German (Switzerland) translation from launchpad #934 (pgrandin)
- Update:i18n:Updated Danish translation from launchpad #933 (pgrandin)
- Update:i18n:Updated Welsh translation from launchpad #932 (pgrandin)
- Update:i18n:Updated Kurdish (Sorani) translation from launchpad #931 (pgrandin)
- Update:i18n:Updated Swedish translation from launchpad #912 (hoehnp)
- Update:i18n:Updated English (United Kingdom) translation from launchpad #896 (hoehnp)
- Update:i18n:Updated German translation from launchpad #895 (hoehnp)
- cleanup:qt:Whitespaces and blank lines cleanup according to cpplint #930 (aerostitch)
- cleanup:search:Remove duplicate code for search_fix_spaces #917 (aerostitch)

- cleanup:search:Minor - Use fallback like the rest of the cases #886 (aerostitch)
- update: Navit logo for (sailfish) desktop #874 (metalstrolch)
- Disable maptool if protobuf-c not found #797 (metalstrolch)

6.2.3 Fixed

- Fix run when in build dir #957 (jkoan)
- fix/enhancement:graphics/layout:get default icon size from layout + draw tunnels transparent + mark oneway streets #884 (metalstrolch)
- Fix:Speech:Android: Replace hyphens with white spaces when using android speech #761 (lains)
- Android notification bar covers part of content #904
- Crash on start when reading traffic data #900
- Missing clipping for polygons with holes leads to slow rendering performance on Windows CE and SDL #869
- Bicycle routing is close to unusable outside built-up areas #844
- Wrong behaviour on Screen rotation or Windows resize #433
- fix:tomtom: Make sure the tomtom_plugin and tomtom_minimal contain navit.xml #875 (gefin)
- vehicle:fix: add compatibility for gpsd 3.20 #959 (aerostitch)
- fix:build:android:Fix android build on circleci #955 (jkoan)
- fix:various:fix compiler warnings #940 (metalstrolch)
- fix:all:fix compiler warnings #935 (metalstrolch)
- fix:core:Don't flood stack on big map elements #928 (metalstrolch)
- Fix:Android:fixes #919 (jandegr)
- fix:maptool:create valid zip64 central directory #915 (metalstrolch)
- Fix:traffic:Determine item delay correctly #903 (mvglasow)
- Fix:traffic:Use correct coord count for restored items, fixes #900 #902 (mvglasow)
- fix:core:prevent stack overflow on drawing huge polygons #897 (metalstrolch)
- Fix:Android:system bars #890 (jandegr)
- fix:graphics/qt5: Allow to draw transparent polygons #878 (metalstrolch)
- Fix:android:64 bit and cleanup2 #877 (jandegr)
- Fix:toolchain:Remove privileged operation from build script #839 (mvglasow)
- Fix:graphics/android:Fix crash on startup #838 (mvglasow)
- Fix:core: NULL pointer when calling iso8601_to_secs #837 (gefin)
- fix:core:add clipping to polygons with holes #870 (metalstrolch)
- Fix:xml:Fix bike routing #850 (mvglasow)
- Fix: Fix pkgconfig detection, and multiple subsequent build for Android #833 (lains)
- fix: vehicle: Increase layer for cursor for rotation #817 (metalstrolch)
- fix:core: Correct default layout handling. #816 (metalstrolch)

- fix: no redraw of map when stopping drag over `osd_speed_warner`, `osd_stopwatch` and `osd_odometer` #814 (hoehnp)
- Fixed typos #787 (pgrandin)
- Fix:core:Fix Warning in IDE for Items not found #778 (jkoan)
- fix:graphics:Fix issue when try to close the SDL window ion non webos #775 (jkoan)
- Fix:core:Fix `item_def.h` when used whithout defined macros (like in IDE) #771 (jkoan)
- Fix:port/Android:Create notification channel on API 26+ #756 (mvglasow)
- Refactoring:xml: Fix a typo of the OpenStreetMap name in the shipped XML #857 (aerostitch)
- qt5: Remove unnecessary moc includes. #798 (metalstrolch)

6.2.4 Other

- We started to migrate the documentation from the old wiki to the docs folder to be displayed by readthedocs.
- Add:graphics/windows: draw polygon with holes #855 (metalstrolch)
- update building of sailfish OS to version 3.1.0.12 #843 (hoehnp)
- Reverting part of `e7a70e9dc04b94b47fb7786aab4edfd25132c70f` #835 (lains)
- jandegr's fix for the windows builds issues #806 (pgrandin)
- Adding comments and const #788 (lains)
- Sync bookmarks 2 #782 (jkoan)
- Android: Do not reinitialize everything when the activity is recreated #754 (mvglasow)
- Refactoring:cmake:simplify version management #710 (bignaux)
- Refactoring:mingw:simplify toolchain #659 (bignaux)
- Refactoring:cmake:from 2.6 to 3.2 #651 (bignaux)

6.3 v0.5.3 - 2018-08-31

Release 0.5.2 had issues with the version not properly updated in the `CMakeLists.txt`. This release fixes that issue.

6.3.1 Changed

- Update Sailfish spec for release 0.5.3. [Joseph Herlant]
- Update Patch version for release 0.5.3. [Joseph Herlant]

6.4 v0.5.2 - 2018-08-30

6.4.1 Added

- Add:build:Make gradlew executable. [mvglasow]

- Add:build:Ignore gradle artifacts. [mvglasow]
- Add:doc:add codefactor.io (#599) [jkoan]
- Add:desktop:Add keywords to application desktop file (#610) [Joseph Herlant]
- Add:Android:checkstyle-javadoc (#595) [jandegr]
- Add:build:Add cmake4eclipse artifacts to .gitignore (#594) [mvglasow]
- Add Slovakia and Belgium to the maplist. [jandegr]

6.4.2 Changed

- Update:doc:Update AUTHORS file. [Joseph Herlant]
- Update in-tree espeak to espeak-1.48.04 (#624) [Stefan Wildemann]
- Update:scripts:change the sanity script to exclude fib and support. [Joseph Herlant]
- Update:scripts:simplify the variables concatenation (#620) [Joseph Herlant]
- Change parenthesis substitutuin as per CodeFactor request. [Stefan Wildemann]
- Update:scripts:change the scripts that use non-POSIX fonctionnalities to bash v2 (#618) [Joseph Herlant]
- Update:scripts:avoid iterating over ls output by using glob (#614) [Joseph Herlant]
- Update:scripts:change the scripts that use non-POSIX fonctionnalities to bash (#613) [Joseph Herlant]
- Update:style:optimize the usage of cat in shell scripts (#612) [Joseph Herlant]
- Change:gui:Add underscore to internal gui keyboard. (#607) [tengel]
- Update:i18n:Updated Čeština translation from launchpad (#602) [Pierre GRANDIN]
- Update:i18n:Updated Dutch translation from launchpad (#601) [Pierre GRANDIN]
- Update:i18n:Updated Serbian translation from launchpad (#600) [Pierre GRANDIN]
- Update:doc:Remove overused space characters from README.md (#591) [KAMiKAZOW]
- Updated header. [Pierre Grandin]
- Update:i18n:Updated German translation from launchpad (#567) [Pierre GRANDIN]
- Update:i18n:Updated Spanish translation from launchpad (#566) [Pierre GRANDIN]
- Update:gtk:use user-configured units in Points of Interest (POI) window. (#557) [Charles Curley]
- Update:xmlconfig:Use g_strdup instead of strncpy. [Joseph Herlant]
- Update:xmlconfig:Make sure we use g_malloc/g_free instead of malloc/free. [Joseph Herlant]
- Update:doc:Update the AUTHORS file. [Joseph Herlant]
- Update:i18n:Updated French (fr) translation from launchpad (#453) [Pierre GRANDIN]
- Curley distance poi (#638) [Charles Curley]
- Refactor:gui/internal: Add pointer and label when using function “view on map” (#631) [lains]
- Refactor:core:Minor documentation amendment. [mvglasow]
- Refactoring:graphics Improving quality of circles on platforms without native circle method (#628) [lains]
- Refactor:core:Improve documentation of route_value_seg() [mvglasow]
- Update:release:Add more keywords for git changelog

6.4.3 Fixed

- Fix Android apk VersionCode (#633) [jkoan]
- Fix latlon2bookmark (#627) [jkoan]
- Build:Disable map/garmin on Android. [mvglasow]
- Build:Disable vehicle/gpsd on Android. [mvglasow]
- Scripts:WinCE build script (remove sample map, add espeak-data) (#632) [lains]
- Scripts:replace s with the supported use of the space class (#617) [Joseph Herlant]
- Scripts:fix malformed shebang (#615) [Joseph Herlant]
- Cmake:Fix the way we set the LIB_DIR when LIBDIR is set for backward compatibility (#574) [Joseph Herlant]
- Debug:Fix potential unterminated C-string and refactoring debug_vprintf() (#564) [lains]
- Maptool:coverity #262887 - leaked_storage: Variable f going out of scope leaks the storage it points to. (#598) [Joseph Herlant]
- Config:Fix dtd to allow real xmlint (#583) [jkoan]
- Fixed typos. [Pierre Grandin]
- Fixed debug statement. [Pierre Grandin]
- Core:fix log replay. [Pierre Grandin]
- Vehicle_android:Doc update (#571) [jandegr]
- Graphics:Remove missing endif from if else block. [jkoan]
- Core:Fixing dbg() change introduced in 221f783ea1caaaab2f5ceadc6b0fb3e720aac3df (#560) [lains]
- CI convert with libsvg2-bin xxsmal cleanup (#554) [jandegr]
- Debug:Change line separators for dbg to work also on win* platform (#546) [Joseph Herlant]
- Android:Fix for the permissions dialog (#459) [jkoan]
- Revert Fix because the docker image was fixed. [jkoan]
- Icons:Remove links to unknown image from gui_map.svg (#542) [Joseph Herlant]
- Gui_internal:use max/min instead of fmax/fmin as absent from wince. [Joseph Herlant]
- Cleanup:android:Fix android checkstyle complains (#645) [Joseph Herlant]
- Cleanup:scripts:Fix condefactor complaints (#646) [Joseph Herlant]
- Cleanup:maptool: use g_strcmp0 instead of strcmp (#434) [Joseph Herlant]
- Cleanup:scripts:Use \$(..) instead of legacy . . to make codefactor happy (#623) [Joseph Herlant]
- Cleanup:python:cleanup python scripts (#622) [Joseph Herlant]
- Cleanup:global:force the re-attach the return type to the function declaration (#605) [Joseph Herlant]
- Cleanup:android: re-attach return type to function declaration. [Joseph Herlant]
- Cleanup:cmake:retab all CMakeLists.txt files. [Joseph Herlant]
- Cleanup:global:Use astyle to reformat everything. [Joseph Herlant]
- Android lint check (#586) [jandegr]
- Android:Fix:graphics (#580) [jandegr]

- Move the file name to the right spot. [Joseph Herlant]
- Revert #612 for sailfish build script. [Stefan Wildemann]
- Android translations (#589) [jandegr]
- Refactor:Android (#572) [jandegr]
- Rework:android:Move build system from ant to gradle (#553) [jandegr]
- Refactor:Android (#563) [jandegr]
- Refactoring:Android:cleanup (#458) [jandegr]

6.4.4 Removed

- Cleanup:osd&support_espeak:get rid of unused-variables compilation warnings (#619) [Joseph Herlant]
- Cleanup:scripts:removes deprecated usage of egrep (#616) [Joseph Herlant]
- Cleanup:win32:Remove unused variables (#611) [Joseph Herlant]
- Cleanup:cmake:remove trailing spaces. [Joseph Herlant]
- Cleanup:wince:remove unused header (#576) [Joseph Herlant]
- Cleanup:general:Declaring war to control-M. [Joseph Herlant]
- Cleanup:xmlconfig:Remove unreachable code. [Joseph Herlant]
- Cleanup:global:Cleanup some unreachable code. [Joseph Herlant]
- Android:Fix:remove unused dependency libpng (#592) [jandegr]

6.5 v0.5.1 - 2018-04-19

6.5.1 Added

- Adding question mark to icon for position pending (#423) [lains]
- Add:icon:forbidden_area (#331) [youte62]
- Add:build:Add download center (#405) [jkoan]
- Add:icon:danger_area. [youte62]
- Add:icon:carwash. [youte62]
- Add:graphics-qt5:add missing watchEvent slot (#387) [Pierre GRANDIN]
- Add drinking_water.svg designed by me - free to use. [youte62]
- Add gouvernement_building.svg designed by me - free to use. [youte62]
- Add:icon:communication (#343) [youte62]
- Add some Documentation to Plugins (#375) [jkoan]
- Add:icon:concert (#342) [youte62]
- Add:doc:added errors and usage messages to navit_svg2png script (#369) [naggety]
- Add:icon:fish (#330) [youte62]
- Add:icon:update_ghost_town icon (#332) [youte62]

- Add:icon:bridge (#339) [youte62]
- Add:icon:emergency (#326) [youte62]
- Add:gtk:add menu entry to toggle vehicle tracking (#362) [Charles Curley]
- Add:icon:hotspring (#335) [youte62]
- Add:icon:diving. [youte62]
- Add:icon:diving (#329) [youte62]
- Add:icon:drinking_water (#325) [youte62]
- Add:icon:dam (#324) [youte62]
- Add:Icon:crossing (#323) [youte62]
- Add:Icon:concert (#322) [youte62]
- Add:Icon:communication (#321) [youte62]
- Add:icon:carwash (#320) [youte62]
- Add:Icon bridge (#319) [youte62]
- Add:icon:bowling (#318) [youte62]
- Add:icon:autoservice (#317) [youte62]
- Add missing items. [jkoan]
- Add multicountry towns support in mapprool (#305) [mdankov]
- Add a script to get bbox out of geojson. [jkoan]
- Added an install target for support libs. [Pierre Grandin]
- Add some more common landuse polygons (#262) [Johan Fitié]
- Add Zoom to route button to Route section in menu (#264) [Johan Fitié]
- Add some more common landuse polygons. [Johan Fitié]
- Add:mapprool:Better error messages for wrong cmdline parameters. [Sebastian Leske]
- Add:xml:Add 43 polys to Car and Car-dark layouts. [Johan Fitié]
- Add:xpm:New SVG icon for hunting stand, to replace XPM, made by myself :-). [Sebastian Leske]
- Add:xpm:POI Icon for playground, from openstreetmap-carto. [Sebastian Leske]
- Add:gui_internal:New POI filter for bus&train. [Sebastian Leske]
- Add feature poly_reservoir. [Johan Fitié]
- Add grass, grassland, wetland and sand. [Johan Fitié]
- Add poly_basin. [Johan Fitié]
- Add: Add speech module using espeak on QMultimedia. (#233) [Stefan Wildemann]
- Add auto zoom toggle to gui_internal (#237) [Johan Fitié]
- Add Qt5 qml gui skeleton (#236) [Stefan Wildemann]
- Add desktop icons in sizes required by Sailfish OS. (#220) [Stefan Wildemann]
- Add 'Follow' and 'Active' to vehicle_demo to remove the corresponding error messages (Unsupported Attribute) (#229) [Robert Pohlink]

- Add network info menu in gui/internal for Linux (#228) [Pierre GRANDIN]
- Add rules to build sailfish package (#221) [Stefan Wildemann]
- Add strings for permission dialog. [jandegr]
- Add libqt5svg5-dev to test environment. [Stefan Wildemann]
- Add screen blanking for Sailfish OS. [Stefan Wildemann]
- Add Qt5 based vehicle. [Stefan Wildemann]
- Add QT5 graphics. Works on Sailfish OS. [Stefan Wildemann]
- Add:graphics:RaspberryPI hardware accelerated graphics support (#208) [Cedric PAILLE]
- Added automatic publishing to Playstore Beta. [Pierre Grandin]
- Added download links for PlayStore and F-Droid in the README. [Pierre GRANDIN]
- Added some usage images to the README.md. [Pierre GRANDIN]
- Add CI badge. [jandegr]
- Add gui_layers.svg. [jandegr]
- Add a feature to change zoom level in autozoom mode. [pohlinkzei]
- Add set_position to command table. [jandegr]
- Add destination forward/backward. [jandegr]
- Add:gui_internal:Support mode for native virtual keyboard. [mvglasow]
- Add:gui_internal:Rudimentary native on-screen keyboard implementation. [mvglasow]
- Add:graphics_android:Use background_color attribute for system bar tint. [mvglasow]
- Add:port_android:Add custom tint to system bars on API 19+ Still with hardcoded colors and minor rendering bugs. [mvglasow]
- Add:core:Use padding for vehicle cursor if supported by graphics driver. [mvglasow]
- Add:gui_internal:Use padding if supported by graphics driver. [mvglasow]
- Add:osd:Use padding if supported by graphics driver Still buggy (race condition) [mvglasow]
- Add:graphics_android:Implement padding for areas obstructed by system UI. [mvglasow]
- Add:port_android:Make system bars translucent on API 19+, transparent on API 21+ [mvglasow]
- Add:port_android:Apply color theme. [mvglasow]
- Add:port_android:Notification icon which complies with Android style guide for all densities. [mvglasow]
- Add configurability for speed_warner timeout. [Robert Pohlink]
- Add:port_android:Remove ActionBar as it is no longer needed. [mvglasow]
- Add:port_android:OSD button and command to show Android menu. [mvglasow]
- Adding a android/x86 build script. [Pierre GRANDIN]

6.5.2 Changed

- Update NavitVehicle.java. [jandegr]
- Update NavitVehicle.java. [jandegr]

- Change_android:Android no acra (#438) [jandegr]
- Update build_win32.sh. [jandegr]
- Update:i18n:Updated Arabic translation from launchpad (#415) [Pierre GRANDIN]
- Change:osd: compass appearance update (#413) [lains]
- Change:build:Move update of doxygen to own script (#404) [jkoan]
- Update build_win32.sh (#402) [jandegr]
- Update config.yml. [jandegr]
- Update config.h.cmake. [jandegr]
- Update osm_psql.c. [jandegr]
- Update CMakeLists.txt. [jandegr]
- Update:icon:update drinking_water icon. [youte62]
- Update:icon:update government_building icon. [youte62]
- Update:doc:Update changelog to follow semver rules (#355) [Joseph Herlant]
- Change:android:always build apk release (#364) [Pierre GRANDIN]
- Update:icon:bowling (#338) [youte62]
- Update:i18n:Updated French (fr) translation from launchpad (#353) [Pierre GRANDIN]
- Update:i18n:Updated Spanish translation from launchpad (#352) [Pierre GRANDIN]
- Update:i18n:Updated Čeština translation from launchpad (#351) [Pierre GRANDIN]
- Update:doc:Build doxygen doc and push the result to a gh-pages branch (#354) [Joseph Herlant]
- Update:icon:updated dam.svg. [youte62]
- Update:QML2:Various updates on the QML2 UI (#316) [Pierre GRANDIN]
- Updated shapefile driver (#295) [Pierre GRANDIN]
- Update Android screenshot in README.md. [Johan Fitié]
- Change default cursor in Car and Car-dark layouts (#296) [Johan Fitié]
- Update:i18n:Updated Portuguese translation from launchpad. [Pierre Grandin]
- Update size estimations in Android map downloader (#272) [Johan Fitié]
- Change Android default layout from Car-Android to Car (#224) [Johan Fitié]
- Update:i18n:Updated German translation from launchpad. [Pierre Grandin]
- Update:Allow zipcode or town seamless search. [jandegr]
- Updated: F-Droid download link (#199) [zintor]
- Update:i18n:Updated German translation from launchpad. [Pierre GRANDIN]
- Update build_tomtom_plugin.sh. [jandegr]
- Update build_tomtom_minimal.sh. [jandegr]
- Update build_tomtom_plugin.sh. [jandegr]
- Update build_tomtom_minimal.sh. [jandegr]
- Update:i18n:Updated Ukrainian translation from launchpad. [Pierre GRANDIN]

- Update:i18n:Updated Sinhalese translation from launchpad. [CircleCI]
- Update:i18n:Use rebase instead of a PR for translations updates. [Pierre Grandin]
- Update:i18n:Updated Turkish (tr) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Piemontese translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Telugu translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Serbian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Polish translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Urdu translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Simplified Chinese translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Chinese (Hong Kong) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Ukrainian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Slovak translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Tamil translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Slovenian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Thai translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Albanian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Vietnamese translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Sardinian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Swahili translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Russian (ru) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Romanian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Norwegian Nynorsk translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated German, Low translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated German, Low translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Dutch translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Norwegian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Marathi translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Bosnian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Kurdish (Sorani) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Frisian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Latvian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Esperanto translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Galician translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Hebrew translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Filipino translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Japanese translation from launchpad. [Pierre GRANDIN]

- Update:i18n:Updated French/Switzerland translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Hungarian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Hindi translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Croatian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Kazakh translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Italian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Indonesian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Kurdish translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Javanese translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Korean translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Kannada translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Luxembourgish translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Macedonian (mk) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Lithuanian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Mongolian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Malayalam translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Faroese translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Finnish translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Persian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Basque translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated French (fr) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Estonian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated English (United Kingdom) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated English (Canada) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated German translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Spanish translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Welsh translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Greek translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated English (Australia) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated German (Switzerland) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Catalan translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Danish translation for Navit translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Čeština translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Belarusian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Asturian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Arabic translation from launchpad. [Pierre GRANDIN]

- Update:i18n:Updated Afrikaans translation from launchpad. [Pierre GRANDIN]
- Update build_linux.sh. [jandegr]
- Update build_linux.sh. [jandegr]
- Update Navit.java. [jandegr]
- Update circle.yml. [jandegr]
- Update build_android_x86.sh. [jandegr]
- Update build_android.sh. [jandegr]
- Update arm-eabi.cmake. [jandegr]
- Update build_android.sh. [jandegr]
- Update tomtom. [jandegr]
- Update:i18n:Updated Norwegian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Portuguese translation from launchpad. [Pierre GRANDIN]
- Update Makefile.am. [jandegr]
- Update xml for heightlines. [jandegr]
- Update doc. [jandegr]
- Update:i18n:Updated Chinese (Traditional) translation from launchpad. [Pierre GRANDIN]
- Update gui_internal_command.c. [jandegr]
- Update:i18n:Updated German translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated French (fr) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Čeština translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Russian (ru) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Spanish translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Italian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Lithuanian translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Welsh translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Chinese (Traditional) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Chinese (Traditional) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Dutch translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Chinese (Traditional) translation from launchpad. [Pierre GRANDIN]
- Update navit.c. [jandegr]
- Update:i18n:Updated German, Low translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Polish translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated French (fr) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated English (United Kingdom) translation from launchpad. [Pierre GRANDIN]
- Update link to wiki for OSD commands. [jandegr]
- Change format_distance to more proper values. [Robert Pohlink]

- Update:i18n:Updated French (fr) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated English (United Kingdom) translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Danish translation for Navit translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Updated Polish translation from launchpad. [Pierre GRANDIN]
- Update:i18n:Fixing po import script. [Pierre GRANDIN]
- Update osd_core.c. [Robert Pohlink]
- Update attr.h. [jandegr]
- Updated Hungarian translation from launchpad. [Pierre GRANDIN]
- Updated Lithuanian translation from launchpad. [Pierre GRANDIN]
- Updated Catalan translation from launchpad. [Pierre GRANDIN]
- Update circle.yml. [jandegr]
- Update circle.yml. [jandegr]
- Update build_android_x86.sh. [jandegr]
- Update i686-android.cmake. [jandegr]
- Update build_android_x86.sh. [jandegr]
- Update i686-android.cmake. [jandegr]
- Update build_android_x86.sh. [jandegr]
- Update circle.yml. [jandegr]
- Update build_android_x86.sh. [jandegr]
- Update build_android_x86.sh. [jandegr]
- Update build_android_x86.sh. [jandegr]
- Updated COPYRIGHT for 2016. [Pierre GRANDIN]

6.5.3 Fixed

- Contrib:New version for sailfish 0.5.1-r5 (#444) [Stefan Wildemann]
- OSD:Make Image and Scale osd types obey osd_configuration setting (trac:1202) (#439) [Robert Pohlink]
- Icons:fix typo business not bussines (#430) [Joseph Herlant]
- Fix coverity builds (#432) [Pierre GRANDIN]
- Typo:Fix various misspells (#421) [Joseph Herlant]
- Maptool: prevent memory leak when fread fails (#420) [Joseph Herlant]
- Wince:Remove useless files from the output (#412) [Pierre GRANDIN]
- Fix:XML:Adding missing sea/ocean display in T@H layout theme. [lains]
- Build:Fix keyscan for wince (#410) [jkoan]
- Cmake:Fix getting the version from git to fix the wince build (#408) [jkoan]
- Build:Fix download of Cov (#400) [jkoan]
- Corrected files considering more comments by @pgrandin. [Patrick Höhn]

- Fix: added changes as suggested by @pgrandin on github. [Patrick Höhn]
- Corrected header file to change of return type of load_buffer. [Patrick Höhn]
- Wrong use of boolean values. [Patrick Höhn]
- Changes in buffer.c. [Patrick Höhn]
- Avoid NULL pointer in ch.c. [Patrick Höhn]
- Build:fixed compiler warning in maptool module. [Patrick Höhn]
- Navit:navigation.c:distance_set_last (#373) [Charles Curley]
- Build:fixed compiler warning in vehicle module (#394) [Patrick Höhn]
- Build:fixed compiler warning in graphics module. [Patrick Höhn]
- Android:Use date of commit as version code instead of system time (#376) [jkoan]
- Gtk:Initialize Follow menu entry correctly. (#390) [Charles Curley]
- Wince:fixed the wince build (#386) [Pierre GRANDIN]
- Cmake:Make copy of files more reliable (#377) [jkoan]
- Cmake:Rework of xslt convert process to fix the copy issue of the navit.dtd file in android build (see process_xslt) (#381) [jkoan]
- Gtk:Fix imperial factors. (#380) [Charles Curley]
- Android:reworked notification system for sdk 23 (#374) [jkoan]
- Core:Wrong default png size with ksvgtopng (#371) [nagety]
- Gtk:Fix imperial factors (#367) [Charles Curley]
- Improved follow vehicle toggle doc function for GTK (#363) [Charles Curley]
- Export more doxygen content. [Pierre Grandin]
- Ensure that the custom domain is set for doxygen (#361) [Pierre GRANDIN]
- Android:Fixed forced colors in layout to have android auto adaptation (#358) [jkoan]
- Android Fix dialogs and replace _ with T for Translate because _ is deprecated (#350) [jkoan]
- Core:Remove dependency on OpenSSL. [Sebastian Leske]
- Maptool:Remove option -5 (MD5 checksum) from maptool. [Sebastian Leske]
- Core:Remove binfile map encryption support. [Sebastian Leske]
- Gtk:fixed missing imperial units in the GTK ui. (#359) [Charles Curley]
- Android:Add workaround for android platform tools (#357) [jkoan]
- Android:Ensures that we capture all build errors. [Pierre Grandin]
- Core:Reduce POST_SHIFT to avoid int overflow. [Sebastian Leske]
- Build:Add -Wformat-security. [Sebastian Leske]
- Core:Fix warnings from -Wstrict-prototypes. [Sebastian Leske]
- Core:Fix warnings from -Wmissing-prototypes. [Sebastian Leske]
- Core:Fix compiler warnings. [Sebastian Leske]
- Build:Fix conditional compiler warning options. [Sebastian Leske]

- Core:Remove unused macros min/max, which conflict with C++ standard macros. [Sebastian Leske]
- Core:Use fprintf with string literal instead of variable. [trldp]
- Core:Rename README.md to README, since some external builds need a README. [Sebastian Leske]
- Fixes displaying the heightline route info (#294) [Pierre GRANDIN]
- Maptool Accept ISO3166-1:alpha2 tag. [mdankov]
- Xpm:Resize overlooked POI icons to 22x22. [Sebastian Leske]
- Fixed variables name in support lib macro. [Pierre Grandin]
- Xml:Fix XML formatting. [Johan Fitié]
- Android cursor size (#267) [Stefan Wildemann]
- Build:Fix typo which causes CMake warning. [Sebastian Leske]
- Build:Regenerate navit*.xml if XSLT files change. Fixes first part of #1333. [Sebastian Leske]
- Xml:Remove unused XSLT files android_*.xslt. [Sebastian Leske]
- Gui_internal:Add missing types to POI filter bus&train. [Sebastian Leske]
- Graphics_qt5 fix layer drawing and dragging. [Stefan Wildemann]
- Make rotating compatible to Qt 5.2 (#256) [Stefan Wildemann]
- Maptool:Update maptool help and manpage. [Sebastian Leske]
- Xml:Change colors for wetland in Car and Car-dark (#257) [Johan Fitié]
- Xpm:Remove obsolete XPM icons. [Sebastian Leske]
- Xpm:Add white outline to black SVG icons. [Sebastian Leske]
- Xpm:Add white outline to new icon hunting_stand. [Sebastian Leske]
- Fix Car layout issues with wood and water (#240) [Johan Fitié]
- Apply coding style to vehicle_qt5. [Wildemann Stefan]
- Apply coding style to graphics_qt5. [Wildemann Stefan]
- Add autozoom switch to sailfosh config. [Wildemann Stefan]
- Fix trac981: Add Align-Attr to osd type Odometer (#230) [Robert Pohlink]
- Don't crash if LANG environment not set (#232) [Stefan Wildemann]
- Fix POI toggle bug with Car-dark layout (#223) [Johan Fitié]
- Allow unusual building (#215) [Stefan Wildemann]
- Fix #1345 crash on Android. [mdankov]
- Gui_internal:Rename new option to hide_impossible_next_keys. [Sebastian Leske]
- Fix apk signing and bump sdk to 25 - nougat (#209) [jandegr]
- Remove autotools leftovers. [Wildemann Stefan]
- Qt5: only update navit geometry if root window changed size. [Stefan Wildemann]
- Qt5: fix resizing. Add config params w,h,fullscreen. [Stefan Wildemann]
- Qt5: clean up things even if there i no app object. [Stefan Wildemann]
- Qt5: allow passing Qt backend via config.xml. [Stefan Wildemann]

- Qt5: integer to pinter cast as suggested by GLib. [Stefan Wildemann]
- Fix overlay text functions. [Stefan Wildemann]
- Fixed Store client_secrets path. [Pierre Grandin]
- Fixed keyrings path. [Pierre Grandin]
- Port/android:Fix “invalid DT_NEEDED” warnings on API 23+, fixes #1348. [mvglasow]
- Tomtom:Switching to mirrored toolchain. [Pierre Grandin]
- I18n:Ensure that we pull from trunk before rebasing. [Pierre Grandin]
- I18n:Escape commit message when merging updates. [Pierre Grandin]
- I18n:Rebase i18n updates against trunk, not master. [Pierre Grandin]
- Core:Fix iPhone build broken by plugin refactoring. [Sebastian Leske]
- Fixed pot file path when building trunk. [Pierre Grandin]
- Maptool:Correct OSM tag is shop=chemist, not drogist. [Sebastian Leske]
- Map_csv:If we cannot open a CSV map file, log error and return failure. [Sebastian Leske]
- Fixed typo in build_android.sh. [Pierre Grandin]
- Fixing dbus tests dependencies. [Pierre Grandin]
- Fixed libpng version check. [Pierre Grandin]
- Fixed typo in circle.yaml. [Pierre Grandin]
- Fix old iOS jailbreak build. [jandegr]
- Maptool:properly check node reference mark. [Michael Dankov]
- Maptool:proper signedness. [Michael Dankov]
- Core:Added link to the OSD wiki page in navit_shipped.xml. [Pierre Grandin]
- Core:Prevent crash on exit if csv map was used Do not attempt to free(attr->data) of attr_item_type attribute, it holds a number, not a pointer. [Michael Dankov]
- Core:initialize value to suppress valgrind warning The warning had no actual effect, as final result of the expression was not depending on uninitialized value being tested in it. [Michael Dankov]
- Gui_internal:Properly switch keyboard to lower/uppercase Do the automatic switch in gui_internal_keypress_do instead of gui_internal_search_changed so that other text input widgets would benefit, e.g. poi search. [Michael Dankov]
- Gui_internal:Fix Cyrillic caps switch button. [Michael Dankov]
- Port_android:Hide osd buttons during map scroll Setting use_overlay attr of <osd type=”button”> now is supported and enabled in default config. [Michael Dankov]
- Port_android:No system bars tinting before KITKAT There’s no way to make navigation and status bar translucent nor transparent before KITKAT, see <https://developer.android.com/about/versions/android-4.4.html#UI> So corresponding code related to SystemBarTintColorView should be enabled since API 19, KITKAT. It actually was breaking navit on my API 17 device, drawing unrelated semitransparent bars on the top and right sides of the screen. [Michael Dankov]
- Port_android:Do not tint on pre-API-17 Tinting the navigation and status bars (introduced in a3e9e7) makes the whole screen tinted on my Gingerbread (API level 9) device. As the actual tint code in handleResize is anyway activated on API>=17, there’s no reason to create navigationTintColorView and statusTintColorView on API<17. [Michael Dankov]

- Core:Make some counters unsigned to avoid UB on overflow. [Sebastian Leske]
- Gui_internal:Repair “Set distance to”/”Get more” menu items. Fixes #1342. [Sebastian Leske]
- Gui_internal:Hide native keyboard when displaying POI search results. [mvglasow]
- Gui_internal:Make native keyboard compatible with changes after R6421. [mvglasow]
- Port_android:Restore soft keyboard when resuming activity. [mvglasow]
- Gui_internal:Further native keyboard refinements. [mvglasow]
- Port_android:Remove noisy debug code. [mvglasow]
- Gui_internal:Correctly handle fullscreen toggle on Android API 17+ [mvglasow]
- Port_android:Prevent system bar tint views from overlapping. [mvglasow]
- I18n:Fixing translations import. [Pierre GRANDIN]
- Fix wait_before_warn. [Robert Pohlink]
- Core:Fix typo in nav_status_to_text() [mvglasow]
- Osd:Draw correct icon for navigation_status OSD when resizing window. [mvglasow]
- Core:Make attr_dup_content support attr_item_type. Fixes trac #1340. [Sebastian Leske]
- Core:Exit if GUI or graphics init / connection fails. [Sebastian Leske]
- Osd: Convert zero button size to magic indicator size -1. [Stefan Wildemann]
- Osd: Make osd “Button” accept h and w props. [Stefan Wildemann]
- Core:Crash in country_search_new Properly handle coutry search by attr_country_id to fix crash in internal gui when there’s a gps fix and a map supporting attr_country_id such as Reiseplaner is active. Fixes #1337. [Michael Dankov]

6.5.4 Removed

- Removed deprecated files and fixed a configuration example. [jkoan]
- Cleanup old xpm icons (#292) [Johan Fitié]
- Removed autotools leftover (#204) [jandegr]
- Cleanup. [jandegr]

6.5.5 Other

- Show min/max elevation and distance in height profile. [Timo]
- Maptool: migrate strdup to g_strdup. [Joseph Herlant]
- Maptool use g_realloc instead of realloc. [Joseph Herlant]
- Maptool: migrate free to g_free. [Joseph Herlant]
- Maptool: migrate malloc to g_malloc. [Joseph Herlant]
- Refactor:core:Refine documentation of attr.c. [mvglasow]
- Navit/fib: cleanup unused configure* files (#422) [Joseph Herlant]
- Maptool: prevent issues when realloc fails (#419) [Joseph Herlant]

- Doc: fix misspell. [Joseph Herlant]
- Refactor:core:Some doxygen additions and corrections (#406) [mvglasow]
- Store win32 nsis installer in artifacts. [jandegr]
- Remove outdated tts interface (#397) [jkoan]
- Correct header file considering earlier changes. [Patrick Höhn]
- Further adjustments according comments. [Patrick Höhn]
- Doc: added doxygen documentation to buffer.c. [Patrick Höhn]
- Refactor:core:Add more documentation (#392) [mvglasow]
- Refactor:route:Improve documentation. [mvglasow]
- Sailfish build on newer SDKs (#388) [Stefan Wildemann]
- Refactor:route:More documentation (#385) [mvglasow]
- Modify icons designed by me - free to use. [youte62]
- Modify icons designed by me - free to use. [youte62]
- 200*200. [youte62]
- Modify icons designed by me - free to use. [youte62]
- 200*200. [youte62]
- Refactor:core:Documentation in transform.c. [mvglasow]
- Refactor:route:Improve struct route documentation. [mvglasow]
- Refine documentation (#378) [mvglasow]
- Refactor:core:Improve documentatio. [mvglasow]
- Refactor:core:Improve documentation. [mvglasow]
- Core:only GGA or RMC message required, ticket #913 (#365) [naggety]
- Refactor:route:Improve documentation (#372) [mvglasow]
- Refactor:core:Refine documentation. [mvglasow]
- Refactor:route:Improve documentation. [mvglasow]
- Once more with feeling. More indentation issues. [Charles Curley]
- Indentation fixes in my last commit. I did not re-indent the entire file. I will leave that for another time. [Charles Curley]
- The road book now respects imperial for two more columns. The first column (Command) was good for both feet and miles. However, column 2 (Length), and the 4th column (Destination Length), were showing up only in metric. This commit fixes columns 2 and 4. This should fix bug 780. [Charles Curley]
- Refactor:route:Add documentation. [mvglasow]
- Refactor:route:Add documentation. [mvglasow]
- Refactor:route:Add documentation. [mvglasow]
- Revert "Always build 'release' package for android" [Pierre Grandin]
- Revert "Fix:android:Ensures that we capture all build errors" [Pierre Grandin]
- Always build 'release' package for android. [Pierre Grandin]

- Moved Taiwan into its own map download entry. [Pierre Grandin]
- Navit medium GUI icon size fix. [Stefan Wildemann]
- Sailfish version bump. [Stefan Wildemann]
- Patch:core:More Talker IDs other than GP accepted (#314) [naggety]
- Importing wince build script. [Pierre Grandin]
- Revert last commit. [Pierre Grandin]
- Coverity upgrade (#313) [Pierre GRANDIN]
- Revert “Fix:core:Rename README.md to README...” - does not work on GitHub. [Sebastian Leske]
- Moved xpm to icons (#307) [jkoan]
- We always want the failed code from bash not from tee. [jkoan]
- Little quick change to make it working fast. [jkoan]
- Auto pitch and orientation (#306) [Johan Fitié]
- #1352 workaround. [mdankov]
- Smaller Android screenshot for README.md. [Johan Fitié]
- Pitch correction (#303) [Johan Fitié]
- [POI] Resize SVG icons to 22x22 (#304) [Johan Fitié]
- Improve cursor in Car and Car-dark layouts (#302) [Johan Fitié]
- [POI] Improve Bus stop icon (#301) [Johan Fitié]
- [POI] Improve Bicycle shop, Cemetery, Bus stop, Museum, School, Fountain and Tower icons (#300) [Johan Fitié]
- [POI] Improve Public office icon (#299) [Johan Fitié]
- [POI] Improve Bicycle shop icon (#298) [Johan Fitié]
- [POI] Improve Drinking water, Shopping, Sport and Tower icons (#297) [Johan Fitié]
- Display label for heightlines (#284) [jandegr]
- [POI] Improve Attraction, Cemetery, Heliport, Hotel, Parking and Pharmacy icons (#293) [Johan Fitié]
- [POI] Improve Emergency, Golf, Pharmacy, Picnic and Wifi icons (#291) [Johan Fitié]
- [POI] Improve Parking, Shop computer, Shop department, Zoo, TEC common, Unknown and Sport icons (#290) [Johan Fitié]
- [POI] Improve Attraction, Bicycle shop, Boat ramp, Bus, Cemetery and Cinema icons (#289) [Johan Fitié]
- [POI] Improve Car dealer, Castle, Fountain, Hospital, Parking, Peak and Pharmacy icons (#288) [Johan Fitié]
- [POI] Improve Drinking water, Fastfood, Theater, Information, Museum, Shop grocery, Taxi, Toilets and Town-hall icons (#287) [Johan Fitié]
- [POI] Improve Bar, Beer, Bus stop, Fountain, Level crossing, Library, Memorial, Post, Repair service, Shopping and Stadium icons (#286) [Johan Fitié]
- [POI] Improve Bank, Bench, Police, School, Tower, Traffic signals, Viewpoint, Shop apparel and Worship icons (#285) [Johan Fitié]
- [POI] Improve Fuel icon (#283) [Johan Fitié]
- [POI] Improve Exit and Hotel icons (#282) [Johan Fitié]

- [POI] Improve ATM, Fire brigade and Restaurant icons. [Johan Fitié]
- [POI] Improve tower icon more. [Johan Fitié]
- [POI] Improve tower icon (#280) [Johan Fitié]
- Use cached copy for libpng / tomtom (#276) [jkoan]
- Use all available Cores on CI (#273) [jkoan]
- Use datecode insted of Tags as Android id (#274) [jkoan]
- [Android] Delete MapInfoFile (.tmp.info) after successful download. [Johan Fitié]
- Sailfish config (#271) [Stefan Wildemann]
- [Android] Enable autozoom by default (#225) [Johan Fitié]
- Correctly implement draw_drag on graphics_qt5. [Stefan Wildemann]
- Qt5 qml gui - POC update (#263) [Pierre GRANDIN]
- Revert “Add some more common landuse polygons” [jandegr]
- Refactor:maptool:rename variables. [Sebastian Leske]
- QT5/qml2 UI proof of concept (#254) [Pierre GRANDIN]
- Graphics_qt5 QML: Add screen rotation support (#255) [Stefan Wildemann]
- Make the speed_warner toggle on click. [Pierre Grandin]
- Remove Qt5 fonts dependency on Freetype (#253) [Stefan Wildemann]
- Revert: move poly_water_tiled in Car layout. [Johan Fitié]
- Trac-1246: Draw background color even if text label is empty. (#234) [Robert Pohlink]
- Navigation_analyze_roundabout : central_angle may lead to division by 0. [Marc CAPDEVILLE]
- [Android] ‘Toggle POIs’ button also toggle POI labels (#226) [Johan Fitié]
- Keep active vehicle profile when deactivating vehicle (#217) [mcapdeville]
- Graphics Qt5 keyboard (#214) [Stefan Wildemann]
- Qt5 Graphics: Add QML support (#213) [Stefan Wildemann]
- Hide impossible keys at the internal keyboard instead of highlighting the possible ones (leftover from PR5) (#210) [Robert Pohlink]
- Import android.Manifest. [jandegr]
- Ask permissions on sdk >= 23. [jandegr]
- Publish arm packages only on Playstore. [Pierre Grandin]
- Revert “Temporary: Enable Qt5 graphics in navit_shipped.xml” [Wildemann Stefan]
- Temporary: Enable Qt5 graphics in navit_shipped.xml. [Wildemann Stefan]
- Refactor: qt5: follow api changes brought with commit 60bc2797f0bd71d87a4eece306e2368afe2a8866. [Stefan Wildemann]
- Qt5: Use default svg size if size not given. [Stefan Wildemann]
- Qt5 image loading more robust on navit image guessing. [Stefan Wildemann]
- Improve dash handling. Cope with impair patterns. [Stefan Wildemann]
- Ignore old fix provided by Qt5 on startup. [Stefan Wildemann]

- Clear surfaces if transparent. Fix dash setting. [Stefan Wildemann]
- Sailfish needs periodic trigger for display on. [Stefan Wildemann]
- Remove most debug messages. [Stefan Wildemann]
- Ensure that we use jarsigner w/ jdk 1.8. [Pierre Grandin]
- Extract android publishing credentials to current folder. [Pierre Grandin]
- Extract publishing credentials in the right folder. [Pierre Grandin]
- Ensures that google-api-python-client is installed for apk publishing. [Pierre Grandin]
- Sunrise near poles simplification (#206) [jandegr]
- Reducing code duplication in the tomtom builds (#201) [Pierre GRANDIN]
- Jandegr patch 1 (#203) [jandegr]
- Android patches from Jandegr (#202) [Pierre GRANDIN]
- Improved documentation. [Pierre Grandin]
- Bump zlib. [jandegr]
- Bump zlib. [jandegr]
- Removing autotools support. [Pierre GRANDIN]
- Moving the linux bin to a more generic place to match other builds. [Pierre Grandin]
- Android ci optimization (#195) [Pierre GRANDIN]
- Switch badge to trunk. [jandegr]
- Correctly escape PR message. [Pierre Grandin]
- Better PR content for automatic i18n PR. [Pierre Grandin]
- Process translation template update as part of the linux build. [Pierre Grandin]
- Ensure that we build with gtk_drawing_area enabled. [Pierre Grandin]
- Refactor:core:For plugins, use term 'category' instead of 'type'. [Sebastian Leske]
- Only setup coverity when needed. [Pierre Grandin]
- Reworking android build scripts. [Pierre Grandin]
- Ensure that we build all targets in 4 containers. [Pierre Grandin]
- Ensure that we build all targets in 4 containers. [Pierre Grandin]
- Moved win32 dependencies to the build script. [Pierre Grandin]
- Install junit-xml dbus-python in the user's env. [Pierre Grandin]
- Install ndk only when needed. [Pierre Grandin]
- Ensures that python-dev is installed for tests. [Pierre Grandin]
- Ensures that libdbus-glib-1-dev is installed for tests. [Pierre Grandin]
- Bumped libpng version to 1.6.26. [Pierre Grandin]
- Temporarily disabled translation updates. [Pierre Grandin]
- Enabling build parallelization. [Pierre Grandin]
- Ensures that python-dbus is installed, needed for tests. [Pierre Grandin]

- Reduced tomtom CI downloads verbosity. [Pierre Grandin]
- Moving linux dependencies installation to the build script. [Pierre Grandin]
- Sdk 23. [jandegr]
- Install ndk. [jandegr]
- Convert CI to trusty. [jandegr]
- Bump libpng. [jandegr]
- Switch to tomtom minimal build. [jandegr]
- Missing layer. [jandegr]
- Wrapper cleanup. [jandegr]
- Cmake based build for tomtom. [jandegr]
- Gui_internal: free heightlines and diagram_points. [jandegr]
- Autotools. [jandegr]
- Create gui_heightprofile.svg. [jandegr]
- Delete gui_heightprofile.xpm. [jandegr]
- Gui_internal_heightprofile. [jandegr]
- Create gui_heightprofile.xpm. [jandegr]
- Gui_internal: fix occasional crash of heightprofile. [jandegr]
- Gui_internal: layer control. [jandegr]
- Gui internal : some more heightprofile fixes. [jandegr]
- CI trigger. [jandegr]
- Gui_internal : fix heightprofile. [jandegr]
- Refactor:maptool:Support 56 bit node IDs. [Michael Dankov]
- Android: disable pedestrian default build. [jandegr]
- Revert "Fix:gui_internal:Hide native keyboard when displaying POI search results" [Michael Dankov]
- Refactor:port_android:Move system bars tinting code to a separate function. [Michael Dankov]
- Remove debug message. [pohlinkzei]
- Refactor:gui_internal:Remove out-ifdefed code and unused widget states. [Sebastian Leske]
- Refactor:gui_internal:Code cleanup. [Sebastian Leske]
- Typo. [jandegr]
- Layoutswitcher command. [jandegr]
- Delete android_all_densities.xslt. [jandegr]
- Android: remove unused and unmaintained XSLT files. [jandegr]
- Delete NavitGraphics2.java. [jandegr]
- Delete NavitAndroidOverlay.java. [jandegr]
- Android: remove unused classfiles. [jandegr]
- Refactor:gui_internal:Determine menu widget size based on parent size. [mvglasow]

- Importing 241569203/pl from launchpad. [Pierre GRANDIN]
- No-HOV. [jandegr]
- No-HOV. [jandegr]
- Refactor:core:Document navit_get_cursor_pnt() [mvglasow]
- Refactor:gui_internal:Document gui_internal_menu.c. [mvglasow]
- Refactor:core:Add Doxygen docs, #define IMAGE_W_H_UNSET for image dimensions. [Sebastian Leske]
- Refactor:core:Update Doxyfile to Doxygen 1.8.9.1. [Sebastian Leske]
- Refactor:core:Fix Doxygen comments; @code only for code blocks. [Sebastian Leske]
- Refactor:osd:Document osd_set_std_graphic. [mvglasow]
- Refactor:osd:Minor documentation improvements. [mvglasow]
- Refactor:core:Document window.h. [mvglasow]
- Refactor:core:Fix GCC warning about uninitialized data.dangerous_goods. [Sebastian Leske]
- Refactor:core:Remove out-ifdefed code. [Sebastian Leske]
- Archive the xpm's from the linux build on CI. [Pierre Grandin]
- Archive the xpm's from the linux build on CI. [Pierre Grandin]
- Applying patch from trac-832. [Pierre GRANDIN]
- Refactor:osd_core:Document osd_button_adjust_sizes() [mvglasow]
- Trac-1064 render OSM shop=mall as poi_mall. [Pierre GRANDIN]
- Bumping the trunk to 0.5.1. [Pierre GRANDIN]

6.6 v0.5.0 - 2015-12-31

This release was done before the adoption of this changelog format. Use [this v0.5.0-rc.2 to v0.5.0 comparison link](#) to view the corresponding changes.

6.7 [v0.5.0-rc2] - 2015-09-02

This release was done before the adoption of this changelog format. Use [this v0.5.0-rc.1 to v0.5.0-rc.2 comparison link](#) to view the corresponding changes.

6.8 [v0.5.0-rc1] - 2015-08-08

This release was done before the adoption of this changelog format. Use [this v0.5.0-beta.1 to v0.5.0-rc.1 comparison link](#) to view the corresponding changes.

Programming guidelines

NAVIT is a team-project, thus a lot of coders are working on it's development and code base. To get a unified coding style and make it easier for everybody to work with parts, that someone else wrote, we tried to specify the formatting of our source and how we deal with third party modules as following.

7.1 Enforced guidelines via CircleCI

The 1st step of the checks enforced after a PR is created or updated is what we call "sanity checks". Those enforce the coding style for our C and Java files. During this phase we have several steps:

- verification that the modified files don't contain trailing spaces. You can use `sed 's/s*$/' -i "$f"` if you want to clean your files before pushing your PR.
- verification that the style of our C and C++ code is respected using `astyle` (to the exception of the following folders: `navit/support/`, `navit/fib-1.1/`, `navit/traffic/permanentrestrictions/`). You can use the following command on the files you are modifying (replacing `$f` by your file name): `astyle -indent=spaces=4 -style=attach -n -max-code-length=120 -xf -xh "$f"`
- check for compliance with the DTD using `xmllint` on the modified files. You can check this locally by using: `xmllint -noout -dtdvalid navit/navit.dtd "$f"`
- verification that the style of our Java code follows our standards using `./gradlew checkstyleMain`

Note: When you submit a PR make sure your branch is up-to-date with the trunk branch to avoid having checks on files you did not modified.

7.2 Coding Style

We try to follow those simple rules:

- indentation is done using 4 spaces

- the return type of a function is specified on the same line as the function name
- the open brackets should be at the end of the line (on the same line as the function name or the if/for/while statement)
- out line length is of 120 characters

To help us keeping a coherent indentation, we use `astyle` on C, C++ and java files. Usage example:

```
astyle --indent=spaces=4 --style=attach -n --max-code-length=120 -xf -xh my_file.c
```

Note: Because of the bug: [<https://sourceforge.net/p/astyle/bugs/230/>] (<https://sourceforge.net/p/astyle/bugs/230/>) on `astyle`, we cannot rely on `astyle` to handle properly the line length of 120 characters that we choose. It would be recommended to set that line length in the editor you are using.

7.2.1 Character encoding and line breaks

All non-ascii-strings must be utf-8 encoded. Line breaks consist of a linefeed (no carriage return).

7.2.2 C Standard

C95. That means:

- No inline declarations of variables

Instead of

```
{
    do_something();
    int a=do_something_else();
}
```

use

```
{
    int a;
    do_something();
    a=do_something_else();
}
```

* No dynamic arrays

Instead of

```
void myfunc(int count) {
    struct mystruct x[count]
}
```

use

```
void myfunc(int count) {
    struct mystruct *x=g_alloca(count*sizeof(struct mystruct));
}
```

* No empty initializers

Instead of

```
struct mystruct m={};
```

use

```
struct mystruct m={0,};
```

- Use `/*` and `*/` for comments instead of `//`

Note: The restriction to C95 exists mainly to help the [[WinCE]] port, which uses a compiler without full support for C99. Once all platforms supported by Navit use a compiler capable of C99, this decision may be reconsidered.

7.2.3 Use of libraries

- Navit uses [GLIB](#) extensively. In general, code should use GLib's functions in preference to functions from `libc`. For example, use `g_new()` / `g_free()` / `g_realloc()`, rather than `malloc()` / `free()` / `realloc()`, `g_strdup()` rather than `strdup()`, `g_strerror0()` rather than `strcmp()` etc.
- Unfortunately, not all platforms that Navit runs on have a native GLib version. For these platforms, there is code replacing these libraries under `navit/support/`. Take care to only use functions from GLib (or other libraries), that is also present under `navit/support/`. If you need something that is not present there, please discuss it on IRC - it may be possible to add it to the support code.

7.2.4 Comments

General guidelines

- Comments for the entire *file* and *classes/structs/methods/functions* is the 'minimum requirement'. Examples see below.
- Please comment your code in a significant and reasonable way.
- A quick description of (complicated) algorithms makes it easier for other developers and helps them to save a lot of time.
- Please add a doxygen description for all function you should add. You are welcome to add it too to older functions. Doxygen result can be found [there](#)

Example :

```
/**
 * @brief Change the current zoom level, zooming closer to the ground.
 *
 * This sentence starts the "detailed" description (because this is a new
 * paragraph).
 *
 * @param navit The navit instance
 * @param factor The zoom factor, usually 2
 * @param p The invariant point (if set to NULL, default to center)
 * @returns nothing
 */
void navit_zoom_in(struct navit *this_, int factor, struct point *p)
```

Templates

This is an example how you could (should) comment your files and functions. If you have any suggestions for improvement, please feel free to [[Talk:Programming_guidelines|discuss]] them with us. These templates should be doxygen-conform - if not, please correct them. A more comprehensive overview of possible documentation can be found [here](#).

Files

```
/** @file can.cpp
 * @brief CAN-Camera Framework :: CAN container class and high level functions
 *
 * Some documentation regarding this file.
 *
 * @Author Stefan Klumpp <sk@....>
 * @date 2008
 */
#include "can.h">
.
.
.
```

Classes/Structs/Functions/Methods

```
/**
 * @brief A short description of this function
 *
 * A lot more of documentation regarding this function.
 * @param raw_data Some string to pass to the function
 * @return Nothing
 */
void CanData::processData(string &raw_data) {
.
.
.
}
```

7.2.5 Java standards

For the Java code we follow the Google coding conventions from Google Java Style that can be found at <https://google.github.io/styleguide/javaguide.html>.

This style is enforced by using Checkstyle. Its definition file can be found at the root of the repository: [checkstyle.xml](#)

Note: Checkstyle 8.25 minimum is required if you want to run checkstyle without using gradle.

Please add yourself to the list of authors, if you make a significant change.

Commit guidelines

Here are the current guidelines if you want to commit something. Please also read the [programming guidelines](../programming_guidelines.html)

8.1 ‘Core’ components changes

Do not modify a ‘core’ component without discussing it first with the project leads.

Core components include data structures, configuration handling. If you are unsure, just ask.

8.2 Commit per feature

When committing, try to have one commit per feature (or per meaningful part of a larger feature). The goal is to always have working code; at least make sure each commit leaves the repository in a compilable state.

Also avoid putting multiple, independent changes into one commit. Thus if you have multiple, independent changes in your local working copy, avoid committing a whole folder at once, especially Navit’s Sourcecode root. Instead, explicitly select the files for each commit.

8.3 Format of the commit log

Since we are too lazy to maintain a Changelog, we have a script which parses the commit logs and generate a Changelog for us.

We have agreed about using the following syntax : `<Action>:<component>:<log message>[!Optional comments]`

Examples :

- Fix:Core:Fixed nasty bug in ticket #134
- Fix:GTK:Fixed nasty bug about destination button!Thanks someguy for the patch!

Action can be something like:

- Fix (bug fix)
- Add (new feature)
- Patch
- Refactoring (does not change behavior of the program)

It allows the changes to be sorted by categories

The most common components are:

- core
- gui/gtk
- gui/internal
- graphics/gtk
- graphics/qt_qpainter
- graphics/opengl
- mapdriver
- tools

The comment part is optional. Useful for when applying a patch for example, and giving credits. The part after | will not appear in the wiki.

About the log message, it's up to you :)

9.1 Build dependencies

Note that most dependencies are optional, meaning Navit will build without them, but you may find that you have some crucial features missing such as the GUI.

In general you will need one of *ksvgtopng*, *rsvg-convert* or *Inkscape* to build pre-scaled icons in the xpm directory - the build process will detect if you have one of those installed, and warn you otherwise.

Please see platform specific sections such as Nokia N8x0 for their additional development environment dependencies.

To build with CMake you will need *CMake 2.6* or newer.

9.1.1 OpenSuse dependencies

Compilation tools:

- libtool
- automake
- autoconf
- gettext-devel
- glib2-devel
- gcc

Optionals:

- To build mapproot: * protobuf-c * libprotobuf-c-devel
- GTK Gui: * gtk2-devel
- OpenGL Gui: * SDL-devel * cegui-devel (AT LEAST 0.5). 0.5 is in packman repository. One click install for [10.3](http://api.opensuse-community.org/searchservice//YMPs/openSUSE_103/21b23afee0c62d4b5350bff51ac7aa41e2c28522) [10.2](<http://packages.opensuse-community.org/>)

aluminium.png) * freeglut-devel * QuesoGLC (at least 0.6) <http://www.kazer.org/navit/quesoglc-0.7.0-1.i586.rpm> * gcc-c++

9.1.2 Gentoo dependencies

There is a Gentoo ebuild for navit in Gentoo's Bugzilla : http://bugs.gentoo.org/show_bug.cgi?id=176552

If you want, you can vote for it so it gets included in portage : http://bugs.gentoo.org/votes.cgi?action=show_user&bug_id=176552#vote_176552

You can also try the ebuild in the overlay : [sunrise overlay](<http://www.gentoo.org/proj/en/sunrise/>). The ebuild is based on the svn to have the latest version of navit.

9.1.3 Debian / Ubuntu dependencies

It compiles flawlessly on a Lenny (5.0) or later and on *Ubuntu 12.04 LTS* (Precise Pangolin) or later, once all dependencies installed.

Note that this section is for build Navit with CMake. These are not "all" packages that you need, only the packages that must be installed, i.e. that are not part of the default (desktop) install. If you removed packages after installation, you may have to re-install them.

Absolute minimum requirements:

```
gcc cmake zlib1g-dev libpng12-dev libgtk2.0-dev librsvg2-bin
```

Note that not all these packages are strictly required (for example, mapprool can be built without installing GTK+), but this is the smallest practical set of packages if you want to run Navit.

- Translations for the user interface: *gettext*
- Mapprool: *protobuf-c-compiler libprotobuf-c-dev*
- GTK+ is included in minimum requirements. *libimlib2-dev* is needed to enable `draw_image_warp` function which, in turn allows to use raster maps as discussed in [track #1285](<http://trac.navit-project.org/ticket/1285>)
- SDL: *libsdl-image1.2-dev libdevil-dev libglc-dev freeglut3-dev libxmu-dev libfribidi-dev*
- OpenGL graphics: *libglc-dev freeglut3-dev libgl1-mesa-dev libxft-dev libglib2.0-dev libfreeimage-dev*
- QT: *libqt4-dev* (This package will pull in all the required packages as dependencies.)
- gpsd: *gpsd gpsd-clients libgps-dev* (optional, but certainly nice to have)
- espeak: *espeak* (optional)
- speechd: *libspeechd-dev* (optional, you are better off with using espeak)
- dbus: *libdbus-glib-1-dev* (optional, you most likely don't need this.)
- python: *python-dev* (optional, you most likely don't need this.)
- saxon: *libsaxonb-java* (only required for android)

Everything in one command:

```
sudo apt-get install cmake zlib1g-dev libpng12-dev libgtk2.0-dev librsvg2-bin \
  g++ gpsd gpsd-clients libgps-dev libdbus-glib-1-dev freeglut3-
↳ dev libxft-dev \
  libglib2.0-dev libfreeimage-dev gettext protobuf-c-compiler
↳ libprotobuf-c-dev
```


9.1.4 Fedora dependencies

Compilation tools:

- gettext-devel (provides autopoint)
- libtool (will install a bunch of other needed packages)
- glib2-devel
- cvs
- python-devel

OpenGL GUI:

- cegui-devel
- freeglut-devel
- quesoglc-devel
- SDL-devel
- libXmu-devel

GPSD Support:

- gpsd-devel

GTK Gui:

- gtk2-devel

Speech support:

- speech-dispatcher-devel

Installing all dependencies:

```
sudo yum install gettext-devel libtool glib2-devel cegui-devel freeglut-devel_  
↪quesoglc-devel SDL-devel libXmu-devel gpsd-devel gtk2-devel speech-dispatcher-devel_  
↪cvs python-devel saxon-scripts
```

9.2 Taking care of dependencies

9.2.1 Getting Navit from the GIT repository

First, let's make sure we are in our home directory: this is only for the sake of making this tutorial simple to follow. You can save that directory anywhere you want, but you will have to adapt the rest of the instructions of this guide to your particular case.

```
cd ~
```

Now, let's grab the code from Git. This assumes that you have git binaries installed.

```
git clone https://github.com/navit-gps/navit.git
```

9.2.2 Compiling

GNU autotools was the old method but is removed in favour of CMake.

CMake builds Navit in a separate directory of your choice - this means that the directory in which the Git source was checked out remains untouched.

```
mkdir navit-build
cd navit-build
```

Once inside the build directory just call the following commands:

```
cmake ~/navit
make
```

Note that CMake will autodetect your system configuration on the first run, and cache this information. Therefore installing or removing libraries after the first CMake run may confuse it and cause weird compilation errors (though installing new libraries should be ok). If you install or remove libraries/packages and subsequently run into errors, do a clean CMake run:

```
rm -r ~/navit-build/*
cmake ~/navit
```

9.2.3 Running the compiled binary

It is advised to just run this binary locally at the moment (i.e. not to install system-wide). Note that for this to work, Navit must be run from the directory where it resides (that is, you must first change your working directory, as described above). If Navit is run from another directory, it will not find its plugins and image files, and will not start.

Here, I am skipping the usual *make install* because we don't need to install navit systemwide for this example.

To execute navit, you can simply click on the binary file (if you are sure it is compiled properly) and it should launch. If you prefer to launch it from a terminal, you need to go into the directory containing the binary, first, like so:

```
cd ~/navit/navit/
./navit
```

9.2.4 Updating the GIT code

You don't need to recompile everything to update navit to the latest code; with *git pull* only the edited files will be downloaded. Just go to the navit directory (e.g. *~/home/CHANGE/~/navit*) and run:

```
git pull
```

You then only need to run *make* again from your binary folder (*navit-build* in the cmake example, or the current folder when using autotools).

9.2.5 Prebuild binaries

[[Download Navit|Prebuilt binaries]] exist for many distributions.

9.2.6 Configuring the beast

This is [Configuration](<https://wiki.navit-project.org/index.php/Configuration>), young padawan. Good luck :)

You can also check a [post describing a Navit configuration on Ubuntu Jaunty](<http://www.len.ro/2009/07/navit-gps-on-a-acer-aspire-one/>).

10.1 Developing for Android

10.1.1 Pre-requisites

For Navit

- Ensure that your system has the correct dependencies installed to build Navit.
- Checkout the latest copy of Navit from git. The instructions on this page assume you have checked-out a copy to `~/src/navit/`

```
cd ~/src
git clone git@github.com:navit-gps/navit.git
```

For Android

- Download the Android [SDK](#) and [NDK](#)
- Unzip the SDK and NDK to a directory of your choice. The following instructions assume that the SDK and NDK have been unzipped to `~/src`.
- Ensure that the following paths are on your PATH environment variable: `path-to-sdk/tools path-to-sdk/platform-tools path-to-ndk/toolchains/arm-linux-androideabi-4.4.3/prebuilt/linux-x86/bin`
- On Ubuntu you can accomplish this using the following command (assuming the SDK and NDK have been unzipped to `~/src`):

```
export ANDROID_NDK=~/src/android-ndk-r8
export ANDROID_SDK=~/src/android-sdk-linux
export PATH=$PATH:$ANDROID_NDK/toolchains/arm-linux-androideabi-4.4.3/prebuilt/linux-
↳x86/bin:$ANDROID_SDK/tools:$ANDROID_SDK/platform-tools
```

Double-check the version numbers in the paths and adapt if required. If you are running a 64-bit version of the NDK, replace *linux-x86* with *linux-x86_64*.

Prepare Android SDK:

- run android in command line - this will show a GUI for preparing Android SDK
- select Android 2.2 (API 8) and click ‘Install N packages...’ button - this will download SDK packages that support API 8

10.1.2 Compiling

We have recently switched from CMake to gradle for the Android build. CMake is no longer supported for the Android build, see the section on gradle for details.

Assuming you’ve followed the previous steps, you’re probably setup to start compiling. Ensure that:

- You have a java-jdk installed on your system. On Ubuntu: `sudo apt-get install openjdk-6-jdk`. When multiple version of java are installed `update-java-alternatives -s java-1.6.0-openjdk-i386`
- On Ubuntu, ensure that ant1.8 is installed: `sudo apt-get install ant1.8`
- Make sure that you have saxonb-xslt installed: `sudo apt-get install libsaxonb-java`

10.1.3 With cmake

- Ensure that you have **CMake 2.8** installed.
- Create a build directory - this will be the directory into which the Android version of Navit will be built. Assume we’ve made one in `~/src` as follows:

```
cd ~/src
mkdir android-build
```

- Optional: Add a `SOURCE_PATH` environment variable to your system, pointing to the directory in which you checked out a copy of Navit. You can leave out this step, but make sure you change `$SOURCE_PATH` in the next steps to the actual path of the directory. `export SOURCE_PATH=~/src/navit`
- Step into the build directory: `cd ~/src/android-build`
- Run CMake (ensure that you’re in the build directory when you do this!):

```
cmake -DCMAKE_TOOLCHAIN_FILE=$SOURCE_PATH/Toolchain/arm-eabi.cmake -DCACHE_SIZE=
↪ "(20*1024*1024)" -DVOID_FLOAT=1 -DANDROID_PERMISSIONS="CAMERA" -DANDROID_API_
↪ VERSION=8 -DXSLT_PROCESSOR=/usr/bin/saxonb-xslt $SOURCE_PATH
```

- Build the apk package:

```
make
make apkg
```

* The previous commands have now created a package called ``Navit-debug.apk`` in the_

↪ following directory: `~/src/android-build/navit/android/bin``

* Copy the package to your device (i.e. sd-card) and run it from there (through a_

↪ file-manager, **for** example), or

* If debug bridge (adb) is enabled run: ``adb install navit/android/bin/Navit-debug.`

↪ `apk``

* Later, to reinstall already installed Navit app run: ``adb install -r navit/android/`

↪ `bin/Navit-debug.apk``

10.1.4 With cmake on Windows

- install CMake 2.8 if you haven't already, add folder with exe to PATH
- install MinGW or use MinGW included in the git windows folder (e.g. C:\msysgitmingw), add folder with exe to PATH
- install saxon .NET version from <http://sourceforge.net/projects/saxon/files/Saxon-HE/9.3/SaxonHE9-3-0-4N-setup.exe/download>,
- rename *transform.exe* to *saxon.exe*, add folder with exe to PATH
- create a build dir (i.e. *mkdir android-build*)
- run from build dir (replace *\$SOURCE_PATH*):

```

cmake -DCMAKE_TOOLCHAIN_FILE=$SOURCE_PATH/Toolchain/arm-eabi.cmake -DCACHE_SIZE=
↪ "(20*1024*1024)" -DANDROID_FLOAT=1 -DANDROID_PERMISSIONS="CAMERA" $SOURCE_PATH -G
↪ "MinGW Makefiles"

* run `MinGW32-make`
* run `MinGW32-make apkg`
* install `Navit-debug.apk` (in `<build path>/navit/android/bin` ) to your device
  * copy `navit/android/bin/Navit-debug.apk` to your device (i.e. sd-card) and run it
  ↪ from there or
  * if debug bridge (`adb`) is enabled run `adb install navit/android/bin/Navit-debug.
  ↪ apk`

```

10.1.5 With gradle

Note: this section is still under construction and may not yet be accurate!

These setup instructions are for a machine that does not have Android Studio installed. If you have Android Studio, some of them may not be necessary (or can be accomplished in a different way).

Make sure you have the following Android SDK components installed (if not, install them using Android SDK Manager):

- Android SDK Platform-tools, version 25.0.3 or later
- Android SDK Build-tools, version 27.0.3

Make sure you have the *ANDROID_HOME* environment variable set and pointing to your Android SDK dir. On Linux, this can be accomplished by adding the following line at the bottom of your *.bashrc* file in your home dir: *export ANDROID_HOME="\$HOME/bin/android-sdk-linux_86"* (use the actual path to your SDK install here)

You need to enter the command in your current shell as well in order for it to take effect there as well.

On Ubuntu 18.04 or later (or if your default JRE is Java 9 or later), edit *\$ANDROID_HOME/tools/bin/sdkmanager*. Change line #31 to read:

```

DEFAULT_JVM_OPTS='"-Dcom.android.sdklib.toolsdir=$APP_HOME" -
↪ XX:+IgnoreUnrecognizedVMOptions --add-modules java.se.ee'

```

Now run *\$ANDROID_HOME/tools/bin/sdkmanager "cmake;3.6.4111459"* and accept the license agreement. (If the package is not found, run *\$ANDROID_HOME/tools/bin/sdkmanager -list | grep cmake* and install the cmake version reported there.)

Make sure you have NDK version 12 or later (if you don't, install it with *\$ANDROID_HOME/tools/bin/sdkmanager "ndk-bundle"*.)

If you did **not** install NDK through *sdkmanager*, make sure you have the *ANDROID_NDK_HOME* environment variable set and pointing to your Android NDK dir. (If not, add it as described above.)

Change to the Navit source dir and run *./gradlew build*.

So far, two issues have been observed with the build:

- Bitmap resources are missing from the APK. A workaround is described [here](#) integration of these steps into gradle is being worked on.
- Building *vehicle/gpsd* and *map/garmin* fails on Android. As a workaround, edit *CMakeLists.txt*, inserting the following two lines in the *if(ANDROID)* block (around line 710):

```
set_with_reason(vehicle/gpsd "Android detected" FALSE)
set_with_reason(map/garmin "Android detected" FALSE)
```

10.2 Testing an alternative build

If you want to try an alternative build (e.g. Jan's builds with alternative routing) you can do it by :

- enable unsigned apk installation ([example build](#))
- installing an alternative apk (e.g. [an APK from CircleCI](#))
- you will probably need an alternative map to match the application requirements (such as [this one](#))

Here are some notes about running navit under Apple Mac OSX.

11.1 What you will need

You need Xcode Tools and MacPorts in order to install navit.

MacPorts developers suggest to install Xcode Tools from <http://developer.apple.com/tools/xcode/> and not from the Mac OSX install disk. See [Mac-How](#)

Make sure you don't have fink installed on your system, it can confuse MacPorts package building and installation.

- GTK Gui: You should only need gtk2 and glib2 via macPorts
- SDL Gui: Untested yet.

11.2 Installation instruction

Download Xcode Tools from <http://developer.apple.com/tools/xcode/> and install it with X11 SDK

Download and Install MacPorts from <http://www.macports.org/>, or update your version

```
sudo port -d selfupdate
```

Open up a terminal

make sure your PATH variables has `/opt/local/bin` and `/opt/local/sbin` in it:

```
echo $PATH
```

Install automake, wget, libtool, gpsd (if you want gps support), gtk2 and glib2 (for gkt GUI) with

```
sudo port install automake wget gpsd gtk2 glib2 libtool
```

Download navit or checkout it from Git

```
git clone git@github.com:navit-gps/navit.git
```

You may also need a header file to handle endian issues (for PPC only)

```
wget https://navit.svn.sourceforge.net/svnroot/navit/tags/R0_1_0/navit/projs/  
↳CodeBlocks/Win32Extra/byteswap.h
```

If you want to install navit along the MacPorts packages, you need to use the /opt/local directory as prefix:

```
./autogen.sh && ./configure --prefix=/opt/local --disable-binding-python
```

type *make* to build NavIt, and *sudo make install* to install it.

Then, you may edit and adapt your *navit.xml* file. The XML matype is not supported, however normal Navit binfile works perfectly.

11.3 Speech

If you want (spoken) directions, get *espeak* from <http://espeak.sourceforge.net>, install as advised and use the following snippet in your *navit.xml*:

```
<speech type="cmdline" data="speak -vde+f4 '%s'"/>
```

This will tell *espeak* to use a female (f) german (de) voice.

11.4 Using xcode

Download one of the [Git sources](#) that don't contain *autogen.sh*.

Open X-Code and create a new project. Cocoa will suffice

Add in a new target by clicking the triangle next to "Targets" and selected the location of the navit folder. Delete the previous target.

Delete the default files, and add in the navit files.

In a terminal, go into the navit folder.

```
./configure --disable-binding-python --disable-sample-map --disable-maptool
```

xcode can now build the navit

You can also use CMake.

```
cd navit && cmake -G Xcode .
```

11.5 Something went wrong?

Please let us know by filing an issue on Github or reach out on IRC.

This is a tutorial for Navit on WinCE/WinMobile. If you just want to download a cab file see [[WinCE]].

This page explains how to build Navit for WinCE/WinMobile with *cegcc* <<http://cegcc.sourceforge.net>>.

In November 2009 versions compiled using *arm-cegcc-gcc* (both revision 1214 and release 0.59.1) had problems (threw `exception_datatype_misalignment` and caused access violations).

Using the variant *arm-mingw32ce* of CeGCC 0.59.1 it was possible to build a working executable which can be debugged.

The automatic builds from the subversion repository seem to use an adjusted? version *arm-wince-mingw32ce* (see *build logs* <<http://download.navit-project.org/logs/navit/wince/svn>>).

12.1 Building using arm-mingw32ce

Install arm-mingw32ce

```
mkdir -p navit
cd navit
export NAVIT_PATH=`pwd`
wget -c https://sourceforge.net/projects/cegcc/files/cegcc/0.59.1/mingw32ce-0.59.1.
↳tar.bz2/download
tar xjf mingw32ce-0.59.1.tar.bz2
```

Building Navit using Cmake

```
#!/bin/bash
export NAVIT_PATH="/usr/src/navit"
export MINGW32CE_PATH="/opt/mingw32ce"
export PATH=$PATH:$MINGW32CE_PATH/bin

cd $NAVIT_PATH
if [ ! -e build ]; then
```

(continues on next page)

(continued from previous page)

```

mkdir build;
fi;
cd build

cmake \
-DMAKE_TOOLCHAIN_FILE=$NAVIT_PATH/Toolchain/arm-mingw32ce.cmake \
-Dsvg2png_scaling:STRING=0,16 \
-Dsvg2png_scaling_nav:STRING=32 \
-Dsvg2png_scaling_country:STRING=16 \
$NAVIT_PATH
make

```

For sequential builds it is sufficient to issue “make” in the build directory. A rerun of cmake is only necessary if parameters are changed.

12.2 Remote Debugging

Download the debugger provided by the CeGCC project:

```

cd $NAVIT_PATH
wget -c https://sourceforge.net/projects/cegcc/files/cegcc/0.59.1/gdb-arm-0.59.1.tar.
↳bz2/download
tar xjf gdb-arm-0.59.1.tar.bz2

```

Start navit (from SD card) in debug server at target (using TCP port 9999):

```
gdbserver :9999 "\Mounted Volume\navit\navit.exe"
```

Execute remote debugger at host, if target’s IP address was 192.168.1.112:

```

$NAVIT_PATH/opt/mingw32ce/bin/arm-mingw32ce-gdbtui $NAVIT_PATH/navit/navit.exe -eval-
↳command="target remote 192.168.1.112:9999"

```

12.3 Building using arm-cegcc

12.3.1 Building cegcc

Set the install path to where you want to install *cegcc* <<http://cegcc.sourceforge.net> cegcc>:

```

export CEGCC_PATH=/usr/local/cegcc
svn co -r 1214 https://cegcc.svn.sourceforge.net/svnroot/cegcc/trunk/cegcc
mkdir -p cegcc-builds
cd cegcc-builds
../cegcc/src/build-cegcc.sh --prefix=$CEGCC_PATH --components="binutils bootstrap_gcc_
↳w32api newlib dummy_cegccdll gcc cegccdll cegccthrdll libstdcpdll profile"

```

If you get an error like “‘makeinfo’ is missing on your system” although makeinfo is available (happened with openSUSE 11.2 and Debian Lenny, both 32 bit), add a workaround to the script src/newlib/missing. Insert a new line after the line “makeinfo”:

```
"$@" && exit 0
```

If you get an error like *arm-cegcc-windres: Can't detect architecture*, apply the patch file you find on http://sourceforge.net/tracker/?func=detail&atid=865516&aid=2574606&group_id=173455

12.3.2 Building libraries

November 2009: The libraries below are *not needed* anymore since navit brings its own version of glib.

The libraries require additional (not published or not existing) patches to build. Just skip to section Building Navit.

These are the libraries needed and versions which should work:

- zlib-1.2.3
- libiconv-1.9.1
- gettext-0.17
- libpng-1.2.34
- tiff-3.8.2
- glib-2.18.4

The current versions of these libs don't need many changes, but they all don't know anything about cegcc. Until I found a way to upload the patches, you have to edit the code yourself. Just add `| -cegcc*` to the line containing `-cygwin*` of all files named `config.sub`. Here is the example for `libiconv-1.9.1_cegcc.patch`:

```
diff -ur libiconv-1.9.1/autoconf/config.sub libiconv-1.9.1_cegcc/autoconf/config.sub
--- libiconv-1.9.1/autoconf/config.sub 2003-05-06 11:36:42.000000000 +0200
+++ libiconv-1.9.1_cegcc/autoconf/config.sub 2009-02-06 20:22:14.000000000 +0100
@@ -1121,7 +1121,7 @@
     | -ptx* | -coff* | -ecoff* | -winnt* | -domain* | -vsta* \
     | -udi* | -eabi* | -lites* | -ieee* | -go32* | -aux* \
     | -chorusos* | -chorusrdb* \
-   | -cygwin* | -pe* | -psos* | -moss* | -proelf* | -rtems* \
+   | -cygwin* | -pe* | -psos* | -moss* | -proelf* | -rtems* | -cegcc* \
     | -mingw32* | -linux-gnu* | -uxpv* | -beos* | -mpeix* | -udk* \
     | -interix* | -uwin* | -mks* | -rhapsody* | -darwin* | -opened* \
     | -openstep* | -oskit* | -conix* | -pw32* | -nonstopux* \
diff -ur libiconv-1.9.1/libcharset/autoconf/config.sub libiconv-1.9.1_cegcc/
↳libcharset/autoconf/config.sub
--- libiconv-1.9.1/libcharset/autoconf/config.sub 2003-05-06 11:36:42.000000000 +0200
+++ libiconv-1.9.1_cegcc/libcharset/autoconf/config.sub 2009-02-06 20:23:39.
↳000000000 +0100
@@ -1121,7 +1121,7 @@
     | -ptx* | -coff* | -ecoff* | -winnt* | -domain* | -vsta* \
     | -udi* | -eabi* | -lites* | -ieee* | -go32* | -aux* \
     | -chorusos* | -chorusrdb* \
-   | -cygwin* | -pe* | -psos* | -moss* | -proelf* | -rtems* \
+   | -cygwin* | -pe* | -psos* | -moss* | -proelf* | -rtems* | -cegcc* \
     | -mingw32* | -linux-gnu* | -uxpv* | -beos* | -mpeix* | -udk* \
     | -interix* | -uwin* | -mks* | -rhapsody* | -darwin* | -opened* \
     | -openstep* | -oskit* | -conix* | -pw32* | -nonstopux* \
```

zlib

```
wget http://www.zlib.net/zlib-1.2.3.tar.gz
tar xzf zlib-1.2.3.tar.gz
cd zlib-1.2.3
export PATH=$CEGCC_PATH/bin:$PATH
CC=arm-cegcc-gcc AR="arm-cegcc-ar r" RANLIB=arm--cegcc-ranlib ./configure --prefix=
↳$CEGCC_PATH
make
make install
```

libiconv

```
wget http://ftp.gnu.org/pub/gnu/libiconv/libiconv-1.9.1.tar.gz
tar xzf libiconv-1.9.1.tar.gz
patch -d libiconv-1.9.1 -p1 < libiconv-1.9.1_cegcc.patch
cd libiconv-1.9.1
./configure --host=arm-cegcc --prefix=$CEGCC_PATH
make
make install
```

gettext

workaround for *plural-eval.h:50: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'sigfpe_exit'* extend
gettext-tools/src/plural-eval.h line 32 to *#if defined _MSC_VER || defined __MINGW32__ || defined __CEGCC__* dito
for gettext-tools/gnulib-lib/wait-process.c line 31

```
wget http://ftp.gnome.org/pub/gnome/binaries/win32/dependencies/gettext-0.17.tar.gz
tar xzf gettext-0.17.tar.gz
cd gettext-0.17
patch -p1 < ../gettext-0.17_cegcc.patch
./configure --host=arm-cegcc --prefix=$CEGCC_PATH
make
make install
```

libpng

```
wget http://prdownloads.sourceforge.net/libpng/libpng-1.2.34.tar.gz?download
tar xzf libpng-1.2.34.tar.gz
cd libpng-1.2.34
patch -p1 < ../libpng-1.2.34_cegcc.patch
./configure --host=arm-cegcc --prefix=$CEGCC_PATH
CFLAG="-I $C_INCLUDE_PATH" make
make install
```

libtiff

```
libtool: link: CURRENT ` ' must be a nonnegative integer
```

```
wget http://libtiff.maptools.org/dl/tiff-3.8.2.tar.gz
tar xzf tiff-3.8.2.tar.gz
cd tiff-3.8.2
patch -p1 < ../tiff-3.8.2_cegcc.patch
./configure --host=arm-cegcc --prefix=$CEGCC_PATH
make
make install
```

glib

```
gatomic.c:570: Error: no such instruction: `swp %eax,%eax,[%esi]'
```

```
wget http://ftp.gnome.org/pub/gnome/sources/glib/2.18/glib-2.18.4.tar.bz2
tar xjf glib-2.18.4.tar.bz2
cd glib-2.18.4
patch -p1 < ../glib-2.18.4_cegcc.patch
./configure --host=arm-cegcc --prefix=$CEGCC_PATH
make
make install
```

12.4 Building Navit

```
git clone https://github.com/navit-gps/navit.git
cd navit/navit
```

Add `I-cegcc*` to all files named `config.sub` as for the libraries.

```
WINDRES=arm-cegcc-windres ./configure --disable-vehicle-file --host=arm-cegcc --
↪prefix=$CEGCC_PATH 2>&1 | tee configure-cegcc.log
```

Add to `navitsupportwordexpglob.h`: `|| defined __CEGCC__`

Change include in `navitvehiclewincevehicle_wince.c`: `#include <sys/io.h>`

Add to `navitfile.c`: `&& !defined __CEGCC__`

```
make -j
```


13.1 Compiling using CMake

At the moment, compiling with CMake seems to be the only way to create a runnable binary in Windows.

13.2 Compiling / debugging using CodeBlocks & mingw compiler

Up to and including release 0.0.4 the Win32 builds were supported using the CodeBlocks/mingw development environment, in combination with the glade for win32 GTK development toolkit. For release 0.1.0 and later use native mingw (see below) or cygwin (see below).

13.2.1 Downloads

In order to compile the win32 version of Navit, you need the following software development tools:

- Glade/gtk+ toolkit for win32 from [Glade 3.43 / Gtk 2.12.9](#) and [SourceForgeDownload: Gtk+ 2.10.11](#)
- ming compiler from [Website: mingw](#) or [SourceForgeDownload: MinGW](#) (select Automated MinGW installer).
- CodeBlocks IDE from [CodeBlocks download page](#) (select recent development snapshot) or [SourceForge-Download: CodeBlocks](#)

13.2.2 Installation

Install the packages mentioned above. After everything has been installed you can open the navit.workspace file in CodeBlocks:

Warning: Not up to date! Directory projCodeBlocks was deleted in 2009

13.2.3 Compiling

Warning: Not up to date! Directory projCodeBlocks was deleted in 2009

To compile:

- Start the CodeBlocks application
- Open the navit.workspace file (located in projCodeBlocks directory)
- Set the GTK_DIR environment variable in CodeBlocks (Setting/Environment, and select environments variables)
- the GTK_DIR should point to where you have installed the Glade/Gtk toolkit package (e.g. d:gtk)

Now you should be able to build/debug the navit project:

Note:

ZLIB -lzdll message *Settings > Compiler and Debugger. > Global compiler settings* In the Linker settings TAB (Add C:MinGWlibz.dll.a

SAPI You need to download and install the [Microsoft Speech SDK 5.1](#) for this project to build.

13.2.4 Running from debugger

In order to run navit from the CodeBlocks debugger, you have to:

- Copy the navit.xml file from the source directory into the projCodeBlocks directory
- Copy the xpm directory from the toplevel directory into the projCodeBlocks directory
- Modify the navit.xml to set the map (currently only OSM binary files are supported)

13.3 Compiling and running using cygwin

13.3.1 Download cygwin

Download the cygwin setup.exe from <http://cygwin.com/setup.exe>

Note: I have been unable to build with cygwin according to these instructions. I suggest you only try it if you are knowledgeable and can improve the instructions. --[[User:Nop|Nop]] 13:01, 7 November 2010 (UTC)

13.3.2 Dependencies

You will probably need the following packages from cygwin :

- automake
- autoconf

- gtk2-x11-devel
- libQt4Gui-devel
- libQtSql4-devel
- gcc
- g++ (for qt rendered)
- gettext-devel
- diffutils
- pkgconfig
- xorg-x11-devel
- glib2-devel
- pango-devel
- atk-devel
- libtool
- make
- rsvg
- wget
- cvs because of autopoint

13.3.3 Prepare the build

When using cygwin 1.7 you can skip this block and continue at cygwin 1.7

Edit configure.in and add the following to CFLAGS at line 10:

```
-I/usr/include/glib-2.0 -I/usr/lib/glib-2.0/include
```

It should look like this :

```
CFLAGS="$CFLAGS -Wall -Wcast-align -Wmissing-declarations -Wmissing-prototypes -
↪Wstrict-prototypes -Wpointer-arith -Wreturn-type -D_GNU_SOURCE -I/usr/include/glib-
↪2.0 -I/usr/lib/glib-2.0/include"
```

Now run *autogen.sh* && *./configure*

If you get: *checking for X... no* try adding the following parameters to *./configure* : *-x-libraries=/usr/X11R6/lib*
-x-include=/usr/X11R6/includes

Cygwin 1.7

With cygwin 1.7 is fairly easy to build navit. Install all the required packages(some has different names now). Run the autogen script first *./autogen.sh* and then configure with the following options: *./configure --disable-binding-python --disable-plugins*

13.3.4 Build navit

Skip for cygwin 1.7

Currently, building navit will fail at this point, because we haven't found an implementation of the wordexp function for cygwin.

Here's a message in that thread from an actual competent Cygwin user: <http://www.mail-archive.com/cygwin@cygwin.com/msg16750.html>

The implication of that is a "C library". A "C library" is an "implementation" of reusable code. It consists of a library file that contains the compiled object code and a header file with the matching declarations that goes along with it. The library is implemented as a static archive at build time and simply linked into the app binary. There's nothing to include in that case – it's already in there.

Cygwin 1.7

Just type *make* and *make install*. You can use *stow* for easy install and uninstall stuff without using packagemangement.

13.3.5 Configuration GPS

Note: If this works at all, it's only when running under cygwin. See above for the proper Win32 configuration. –[[User:NoplNop]] 13:04, 7 November 2010 (UTC)

If you have a gps cable device which spits out NMEA data, you can configure it like under unix. Beware of the following enumer

- ComPort1==ttyS0
- ComPort2==ttyS1
- ...

Example:

```
<vehicle name="GPSOnCom3" profilename="car" enabled="yes" active="1" source="file:/  
↪dev/ttyS2" baudrate="38400" color="#0000ff"/>
```

13.3.6 Running under Cygwin

To run navit under cygwin you need to install the cygwin xorg-server. Than just run navit.

13.3.7 Make a redistributable package

Please read and understand <http://cygwin.com/licensing.html> so that you don't infringe Cygwin's intellectual property rights (copyleft) when you distribute the package you've built. Then follows: <http://cygwin.com/setup.html>

13.4 Compiling a native binary using mingw

The main advantage of this method is that it will produce a redistributable binary.

13.4.1 Downloads

In order to compile the win32 version of Navit, you need the following software development tools

- GTK+ toolkit for win32 from [Glade/GTK+](#) (select gtk+-win32-devel)
- MinGW from [MinGW](#) (select Automated MinGW installer)
- MSYS from [MSYS Base System](#)
- msysCORE from [MSYSCore Base System](#)
- diffutils from [diffutils](#)
- autoconf, autogen, automake, gettext and libtool from [MSYS Supplementary Tools](#)
- libiconv from [MinGW's libiconv](#)

Probably the easiest way to obtain and install all the MSYS packages is to follow the instructions [here](#)

For speech support, one option is to use the “cmdline” speech type (refer to [\[\[Configuration\]\]](#)) and a utility such as a Windows port of [Say](#)

13.5 TroubleShooting

```
/bin/m4: unrecognized option '--gnu'
```

Wrong version of m4, use 1.4.13

```
Can't locate object method "path" via package "Request (perhaps you forgot to load
↳ "Request"?)
```

Wrong version of Autoconf, make sure the latest version is installed, plus the wrapper (version 1.7). Also delete autom4te.cache.

```
command PKG_MODULE_EXISTS not recognized
```

For some reason the necessary file “pkg.m4” containing various macros is missing. Find it and put it in ./m4

13.6 Cross-Compiling win32 exe using Linux Ubuntu 14.04.1

This is a quick walk-thru on compiling a win32 exe using Ubuntu as development machine.

13.6.1 Set up Ubuntu to build Linux version

First, setup compiling in linux ubuntu explained in [Linux Development](#) Here is a quick walk-thru:

Get all the dependencies for Ubuntu in one command:

```
sudo apt-get install cmake zlib1g-dev libpng12-dev libgtk2.0-dev librsvg2-bin \
  g++ gpsd gpsd-clients libgps-dev libdbus-glib-1-dev freeglut3-dev libxft-dev \
  libglib2.0-dev libfreeimage-dev gettext
```

get the latest source from git. First, cd into root: `cd ~`

Now, let's grab the code from SVN. This assumes that you have subversion installed. This will download the latest SVN source and put in in folder "navit-source". You can use any location you want for the source, just to keep it simple we place it right in the root.

```
git clone git@github.com:navit-gps/navit.git navit-source
```

Create a directory to put the build in and cd into it:

```
mkdir navit-build  
cd navit-build
```

Start compiling and build navit:

```
cmake ~/navit-source && make
```

At the end of the process navit is built into navit-build/. You can start navit to see if all worked well:

```
cd ~/navit-build/navit/  
./navit
```

13.6.2 Building the win32 exe

Now that we have set up the basic building environment we can build a win32 exe using the next walk-thru.

Install ming32 and the dependencies:

```
sudo apt-get install mingw32 libsaxonb-java librsvg2-bin mingw32-binutils mingw32-  
↳runtime default-jdk
```

now cd into the source:

```
cd ~  
cd navit-source
```

We are going to place the build directory within the source directory. First, make the build directory and cd into it:

```
mkdir build  
cd build
```

From within the build directory start compiling and building:

```
cmake -DCMAKE_TOOLCHAIN_FILE=../Toolchain/mingw32.cmake ../
```

And then make the actual build:

```
make -j4
```

The `-j4` part is used to define the amount of processors the process can use. So if you have a dual-core pc use `-j2` If `-j4` fails, try `-j2` and if that fails try `make` alone.

13.6.3 Known "bugs"

The "locale" folder is generated one level up. because of that the languages in navit are not working Cut and paste (or move) the "locale" folder to the navit folder. This should be investigated and fixed so the folder is in the correct place

after a build. So move *navit-source/build/locale/* to *navit-source/build/navit/locale*

You can run

```
mv navit-source/build/locale/ navit-source/build/navit/
```

The country-flags images in the “town” search are not displayed. This could be due to a conversion error during build, has to be investigated and solved but doesn’t inflict with the use of navit.

There are a lot of empty folders that are not of use. Also there are cmake folders and files in every folder. You can delete those without any problem.

13.7 Windows Mobile/Windows CE

WinCE Development may have details that are relevant for compilation on WindowsCE / Windows Mobile.

You can download now [cab](#) or [zip](#) file for Windows Mobile and WindowsCE ! Highest number is the newest version of NavIt.

Download it and save on your Storage Card. Install it.

Now you have NavIt on your PDA or Mobile Phone.

This is a manual for self compiling (navit.exe)

You need to have a Linux (like Ubuntu). If you didn’t have Linux, start your Linux on Live-CD.

Compiling navit for wince using <http://cegcc.sourceforge.net/>. Download latest cegcc release and install it.

In November 2009 versions compiled using arm-cegcc-gcc (both revision 1214 and release 0.59.1) had problems (threw exception_datatype_misalignment and caused access violations).
 Using the variant arm-mingw32ce of CeGCC 0.59.1 it was possible to build a working executable which can be debugged (see *WinCE Development*).

Source [cegcc-arm](#) and [mingw](#) (TODO dead link)

Current installs in /opt/cegcc. Setup a cross-compile environment:

Example setcegccenv.sh:

```
#!/bin/bash
export PATH=$PATH:/opt/cegcc/bin/
export CEGCC_PATH=/opt/cegcc
export WINCE_PATH=/opt/wince
export PATH=$CEGCC_PATH/bin:$PATH
export CPPFLAGS="-I$WINCE_PATH/include"
export LDFLAGS="-L$WINCE_PATH/lib -L$CEGCC_PATH/lib"
export LD_LIBRARY_PATH="$WINCE_PATH/bin"
export PKG_CONFIG_PATH="$WINCE_PATH/lib/pkgconfig"
export PKG_CONFIG_LIBDIR="$WINCE_PATH/lib/pkgconfig"
```

For installation, compiling and configuring please see manual for NavIt on Linux.

Then autogen.sh and configure navit. Example configure for wince:

```
./configure \
  RANLIB=arm-cegcc-ranlib \
  CXX=arm-cegcc-g++ \
  CC=arm-cegcc-gcc \
  --host=arm-pe-wince \
  --disable-readline \
```

(continues on next page)

(continued from previous page)

```

--disable-dynamic-extensions \
--disable-largefile \
--enable-tempstore \
CFLAGS="-I/opt/wince/include -mwin32 -DWIN32 -D_WIN32_WCE=0x0400 -D_WIN32_IE=0x0400_
↪-Wl,--enable-auto-import" \
LDFLAGS="-L/opt/wince/lib" \
--prefix=/opt/wince/ \
WINDRES=arm-cegcc-windres \
--disable-vehicle-demo \
--disable-vehicle-file \
--disable-speech-cmdline \
--disable-speech-speech-dispatcher \
--disable-postgresql \
--disable-plugins \
--prefix=/opt/wince \
--disable-graphics-qt-qpainter \
--disable-gui-sdl \
--disable-samplemap \
--disable-gui-gtk \
--disable-gui-internal \
--disable-vehicle-gypsy \
--disable-vehicle-file \
--disable-vehicle-demo \
--disable-binding-dbus \
--enable-avoid-unaligned \
--enable-avoid-float

```

If example did not run, do this:

```

./configure \
RANLIB=arm-mingw32ce-ranlib \
CXX=arm-mingw32ce-g++ \
CC=arm-mingw32ce-gcc \
--host=arm-pe-wince \
--disable-readline \
--disable-dynamic-extensions \
--disable-largefile \
--enable-tempstore \
CFLAGS="-mwin32 -DWIN32 -D_WIN32_WCE=0x0400 -D_WIN32_IE=0x0400 -Wl,\
--enable-auto-import" WINDRES=arm-mingw32ce-windres \
--disable-vehicle-demo \
--disable-vehicle-file \
--disable-speech-cmdline \
--disable-speech-speech-dispatcher \
--disable-postgresql \
--disable-plugins \
--prefix=/opt/wince \
--disable-graphics-qt-qpainter \
--disable-gui-sdl \
--disable-samplemap \
--disable-gui-gtk \
--disable-gui-internal \
--disable-vehicle-gypsy \
--disable-vehicle-file \
--disable-vehicle-demo \
--disable-binding-dbus \
--enable-avoid-unaligned \

```

(continues on next page)

(continued from previous page)

```
--enable-avoid-float \
--enable-support-libc \
PKG_CONFIG=arm-mingw32ce-pkgconfig
```

This is basic just to view the maps. Then: *make* As usual, *osm2navit.exe* will fail to compile. *cd navit && make navit.exe* You find *navit.exe* under (your directory)/*navit/navit/navit.exe*

Install *sync* on your system.

For installation you need packages *librapi*, *liprapi2*, *pyrapi2*, *libsinc*. Package *synce-0.9.0-1* contains *librapi* and *libsinc*. You do not need to install it again!

Sources: [Sync](#) If link is crashed, use this: [Sync Link2](#) *libsinc*: [libsinc](#) *pyrapi2*: [pyrapi2](#) *librapi2* [librapi2](#)

Once you have *navit.exe* ready, copy */opt/cegcc/arm-cegcc/lib/device/*.dll* on your device.

For Debian use:

```
synce-pcp /opt/cegcc/arm-cegcc/lib/device/cegcc.dll ":/windows/cegcc.dll"
synce-pcp /opt/cegcc/arm-cegcc/lib/device/cegccthrd.dll ":/windows/cegccthrd.dll"
```

All other Linux/Unix systems use:

```
pcp /opt/cegcc/arm-cegcc/lib/device/cegcc.dll ":/windows/cegcc.dll"
pcp /opt/cegcc/arm-cegcc/lib/device/cegccthrd.dll ":/windows/cegccthrd.dll"
```

Synchronisation with a graphic surface, if connection to device failed:

Packages *RAKI* and *RAPIP* you can use.

RAKI you have in packages *synce-kde* (see [Sync](#)).

RAKI is like *Active Sync*, *RAPIP* is a little bit like *fish://* under *Konqueror*.

Under *SuSE Linux* you can run *kitchensync* (not for all PDA).

For synchronisation you can also use *kpilot* under *Suse Linux* (runs not with all PDA) or *Microsoft Active Sync* under *Windows* (free download at [Microsoft homepage](#)).

You can put your memory card in card reader and copy data. Over console you must type in *sync* before you remove memory card.

Install *navit.exe*.

Debian:

```
synce-pcp navit.exe ":/Storage Card/navit.exe"
```

All other:

```
pcp navit.exe ":/Storage Card/navit.exe"
```

Prepare a *navit.xml.wince*

Change *gui* to *win32* and *graphics* to *win32*.

Fix the paths to your maps “*/Storage Card/binfilemap.bin*”

Debian:

navit

```
synce-pcp binfilemap.bin ":/Storage Card/binfilemap.bin"  
synce-pcp navit.xml.wince ":/Storage Card/navit.xml"
```

All other:

```
pcp binfilemap.bin ":/Storage Card/binfilemap.bin"  
pcp navit.xml.wince ":/Storage Card/navit.xml"
```

For a start best use the samplemap. Now you can launch navit.exe on the device.