Navdatareader User Manual

Navdatareader is a command line tool that allows to create a full SQLite navigation database for [Little Navmap](#) from various sources which are FSX (all versions), Prepar3D (all versions from v3 up), Microsoft Flight Simulator 2020, Microsoft Flight Simulator 2024, X-Plane 11, X-Plane 12 and Navigraph.

The tool allows detailed configuration of included content as well as simple validation and reporting mechanisms.

See also the chapters [Files](#) and [Load Scenery Library Dialog](#) in the *Little Navmap* manual for more information.

General

Loading a scenery library can take from 2 to 5 minutes depending on your setup and amount of scenery add-ons. You can speed this up by excluding directories containing neither airport nor navigation data in the `navdatareader.cfg` configuration file.

All airports that are not located in the default `Scenery` directory of FSX/P3D or are located in the `Custom Scenery` directory of X-Plane are considered add-on airports and will be marked appropriately. The same applies to MSFS airports in folder `Community`. Directories can be excluded from this behavior in the `navdatareader.cfg` configuration file. This can be useful if add-ons only correct airport elevation and these airports should not be marked as add-on airports.

FSX and Prepar3D Navdata

All scenery directories and all BGL files referenced in the `scenery.cfg` file will be scanned for airport, navdata and airspaces. This includes stock, add-on data and navdata updates.

Microsoft Flight Simulator 2020

The simulator is detected automatically. All airports from `Community`, `Official\*`, `Official\fs-base`, `Official\fs-base-nav` are read.

Microsoft Flight Simulator 2024

The simulator has to run to populate the database through SimConnect. Add-on airports cannot be detected. See [FAQ - MSFS 2024 Scenery Library Limitations](#).

X-Plane Airports and Navdata

The *Navdatareader* reads airport and navaid data from X-Plane's `*.dat` files. To check a version of a file you can open it in a text editor that is capable of dealing with large files. The first lines of the file will look like:

```
A
1100 Generated by WorldEditor 1.6.0r1

1   1549 0 0 0A4 Johnson City STOLport
...
```

The first number in the second line is the file version which is `1100` in the example above.

The *Navdatareader* can read the following X-Plane scenery files:

- **Airports (**`apt.dat`**):** Version 850 up to 1200. This covers X-Plane 10 airports and older add-on scenery. Newer files than 1200 might work but are not tested.
- **Navdata (**`earth_awy.dat`**,** `earth_fix.dat` **and** `earth_nav.dat`**):** Version 850 up to
- This excludes X-Plane 10 navdata files. Newer files than 1200 might work but are not tested.
- **Procedures (**`ICAO.dat` **in the** `CIFP` **directory):** All procedures from X-Plane 11 and X-Plane 12.
- **Airspaces (**`*.txt`**):** The included `usa.txt` and all files in OpenAir format. See next chapter for more information.

Additionally the files `user_fix.dat` and `user_nav.dat` in the X-Plane directory `Custom Data` are read.

All files in [OpenAir airspace format](#) will be loaded when reading the X-Plane scenery library too.

Note that airspace files can have errors which may prevent the loading of an airspace file. These hard errors are reported after loading the scenery library. Other errors only affecting single airspaces or the geometry are reported in the log file only.

X-Plane comes with a single airspace file that can be found in `YOUR_XPLANE_DIRECTORY/Resources/default data/airspaces/usa.txt`. Additional airspace files can be downloaded from the [OpenAirspace Directory](#), [Soaring Services](#), [openAIP](#) or [Luftraumdaten Deutschland](#) for example.

Airspace files must have a `.txt` extension and are loaded from the following directories by the *Navdatareader*:

- `YOUR_XPLANE_DIRECTORY/Resources/default data/airspaces`
- `YOUR_XPLANE_DIRECTORY/Custom Data/Airspaces`
- `YOUR_ACCOUNT_NAME/Documents/Little Navmap/X-Plane Airspaces` where `Documents` is the documents directory in your language.

The files can be encoded in any [UTF](#) format but must have a [BOM](#) to be recognized properly. Otherwise Windows ANSI coding (`Windows-1252`) is used. Special characters like umlauts or accents are not displayed correctly in names if the encoding is not correct. All other functionality is unaffected.

You can convert the files using any advanced editor like [Notepad++](#) for example.

Airspaces will appear as duplicates in the map if an airspace file is found in more than one of these directories.

Error Reporting

A return value of `0` indicates a successful compilation.

The program will stop with a message `*** Compilation failed` and the return value on the command line will be `1` if an error occurred during compilation. The database file will be renamed to include the word `BROKEN`.

Check the log files `abarthel-navdata_reader.log` and/or `abarthel-navdata_reader-err.log` and look for messages of severity `WARN`, `CRIT` or `FATAL` in case of failure.

See chapter `Files` below for information on log files.

Command Line Options

Usage: `navdatareader [options]`

-h, –help

Displays a brief help for the command line options.

-v, –version

Displays version and revision information about the program.

-f, –flight-simulator *type*

Required option. Flight simulator type or other data source. Valid values are:

- `FSX`: Microsoft Flight Simulator X. SP2 or Acceleration.
- `FSXSE`: Flight Simulator - Steam Edition.
- `P3DV3`: Lockheed Martin Prepar3D v3
- `P3DV4`: Lockheed Martin Prepar3D v4
- `P3DV5`: Lockheed Martin Prepar3D v5
- `MSFS`: Microsoft Flight Simulator 2020
- `MSFS24`: Microsoft Flight Simulator 2024
- `XP11`: X-Plane 11.
- `XP12`: X-Plane 12.
- `DFD`: A SQLite database containing navigation data.

-s, –scenery *scenery.cfg file*

Optional FSX or Prepar3D `Scenery.cfg` file. The location of this files is automatically determined by registry entries if not given.

The typical locations of the `Scenery.cfg` for Windows 7/8/10 are:

- **Flight Simulator X:** `C:\ProgramData\Microsoft\FSX\Scenery.cfg`
- **Flight Simulator - Steam Edition:** `C:\ProgramData\Microsoft\FSX-SE\Scenery.cfg`
- **Prepar3D v3:** `C:\ProgramData\Lockheed Martin\Prepar3D v3\Scenery.cfg`
- **Prepar3D v4:** `C:\ProgramData\Lockheed Martin\Prepar3D v4\Scenery.cfg`

-b, –basepath *base path*

Required option for X-Plane data compilation. Optional for FSX, P3D and MSFS. Unused for DFD compilation.

FSX or Prepar3D base path for BGL files. This path is automatically determined by registry entries if not given.

For MSFS a path to the folders containing `Community` and `Official`. This is `...\Microsoft.FlightSimulator_8wekyb3d8bbwe\LocalCache\Packages`, for example.

The X-Plane base path cannot be recognized automatically. You have to provide it using this option. On Windows that can be a path like `C:\Simulators\X-Plane 11` where the executable is `C:\Simulators\X-Plane 11\X-Plane.exe`.

-d, –source-database *database*

Required option for a DFD based compilation. Path and name of an extended DFD database. Unused by other compilation types.

-o, –output *database file*

Optional output Sqlite database filename. The default is `navdata.sqlite` in the current directory.

-c, –config *configuration file*

Optional. Extended configuration file for the *Navdatareader*. Default is to use built-in `navdatareader.cfg`.

Examples

Read a FSX scenery database from the default places which are determined by the registry entries of FSX and write the *Little Navmap* database to

`navdata.sqlite` in the current directory:

```
navdatareader -f FSX
```

Read a Prepar3D version 4 database from non standard places. A Prepar3D installation is not needed in this case. The scenery database files are sufficient. The *Little Navmap* database is written to `navdata.sqlite` in the current directory:

```
navdatareader -f P3DV4 -b C:\Simulators\P3DV4 -s C:\ProgramData\P3DV4\Scenery.CFG
```

Read X-Plane scenery data from `C:\My Programs\X-Plane 11` and write the *Little Navmap* database to `D:\navdata-xp11.sqlite`:

```
navdatareader -f XP11 -b "C:\My Programs\X-Plane 11" -o D:\navdata-xp11.sqlite
```

Read a DFD SQLite database and write the *Little Navmap* database to `D:\Output Data\Little Navmap\navdata-1611.sqlite`:

```
navdatareader -f DFD -d "C:\Input Data\dfd1611.sqlite" -o "D:\Output Data\Little Navmap\navdata-1611.sqlite"
```

Files

Logs

The program writes two log files into the current execution directory besides printing all log messages on the console:

- `abarthel-navdata_reader.log`: Contains all messages.
- `abarthel-navdata_reader-err.log`: Contains warning, critical and fatal messages only.

Log file names are rolled on each execution of the program. A maximum of two log files is kept.

Navdatareader Configuration

This configuration file (option `-c`) allows an in-depth configuration of the included features. Note that not all features/filters are available for all simulator/source types.

This file is normally only needed for debugging or development purposes.

The file is Ini-style and all options are commented.

```
# Database - Currently not used - SQLite only
[Database]
Type=QSQLITE
ConnectionName=
ConnectionOptions=
HostName=
Port=
UserName=
Password=

[Options]

# Show statistics
DatabaseReport=true

# Fail if most important tables are not populated
BasicValidation=true

# Process airport deletesDatabase - only FSX/P3D
ProcessDelete=true

# filter out dummy runways - only FSX/P3D
FilterRunways=true

# Write incomplete objects - only FSX/P3D
SaveIncomplete=true

# Connect airways
ResolveRoutes=true

# Create routing application tables
CreateRouteTables=true

# Debugging options - slows compilation down and creates huge log files
Verbose=false

# Debugging option - Commit after each update - very slow
Autocommit=false

# Delete duplicates
Deduplicate=true

# Vaccuum database
VacuumDatabase=true

# Gather schema statistics
```

```
AnalyzeDatabase=true

# Tables names and minimum number of rows
# An exception is throw if the tables have less rows
# The numbers below are the common minimums for bare FSX and DFD database
[BasicValidationTables]
airport=13000
airport_file=0


[Filter]
# Not all filters are applied to the X-Plane compilation and none to the DFD compilation.
#
# If include filter is set only the listed features are loaded
# If exclude is set all listed features as excluded
# Exclude has priority before include
# All filters are case insensitive, comma separated and use globbing expressions
# ? Matches any single character
# * Matches zero or more of any characters
# [...] Sets of characters can be represented in square brackets

# Filter files by name excluding path  - only X-Plane and FSX/P3D
# Examples: BNXWorld?.bgl or APX45*.bgl or FTX_NZSI_Unlisted_Airstrips.BGL
IncludeFilenames=

# FSX filename prefixes: - only X-Plane and FSX/P3D
# Exclude DEM, bridges and other scenery per default
ExcludeFilenames=brx*,obx*,cvx*,dem*

# Filter files by path relative from scenery.cfg minus the scenery directory - only X-Plane and
```

FSX/P3D # Example: Addon Scenery/JetwayExclude/,*Scenery/0501/*,Scenery/0601/* IncludePathFilter= ExcludePathFilter=

```
# Filter airports three or four letter ICAO ident - only X-Plane and FSX/P3D
IncludeAirportIcaoFilter=
#EDVK EDXW
ExcludeAirportIcaoFilter=

# Filter by object type. Some combinations of this can result in unexpected behaviour  - only
```

X-Plane and FSX/P3D # that is caused by the hierarchical dependency of objects. E.g. filtering out runways # will also remove approaches and ILS. # Possible values are: # AIRPORT, RUNWAY, HELIPAD, START, APPROACH (and transitions), APPROACHLEG (and transition legs), # COM, PARKING, ILS, VOR, NDB, WAYPOINT, AIRWAY, MARKER, APRON, APRON2, APRONLIGHT, FENCE, TAXIWAY, ROUTE, # GEOMETRY (apron and apron light geometry), BOUNDARY (airspaces) IncludeBglObjectFilter= ExcludeBglObjectFilter=FENCE,APRONLIGHT,APRON2,TAXIWAY_RUNWAY,VEHICLE

```
# Filters that define if an airport is defined as addon or not
IncludeAddonPathFilter=
ExcludeAddonPathFilter=Scenery/*
```

Links

The source code for this application is available at Github:

- atools
- navdatareader

All manuals are also available online:

- Navdatareader User Manual
- Little Navmap User Manual

License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses.