

Windigipet 2025 – FlytLoks documentation

24-9-2025 – Jens Krogsgaard, Copenhagen - Denmark

Table of contents

1.	Technical Documentation.....	1
1.1.	Purpose	1
1.2.	Functionality	2
1.3.	Architecture	3
1.4.	Limitations.....	4
2.	User Guide.....	5
3.	Distribution Guide	7

1. Technical Documentation

1.1. Purpose

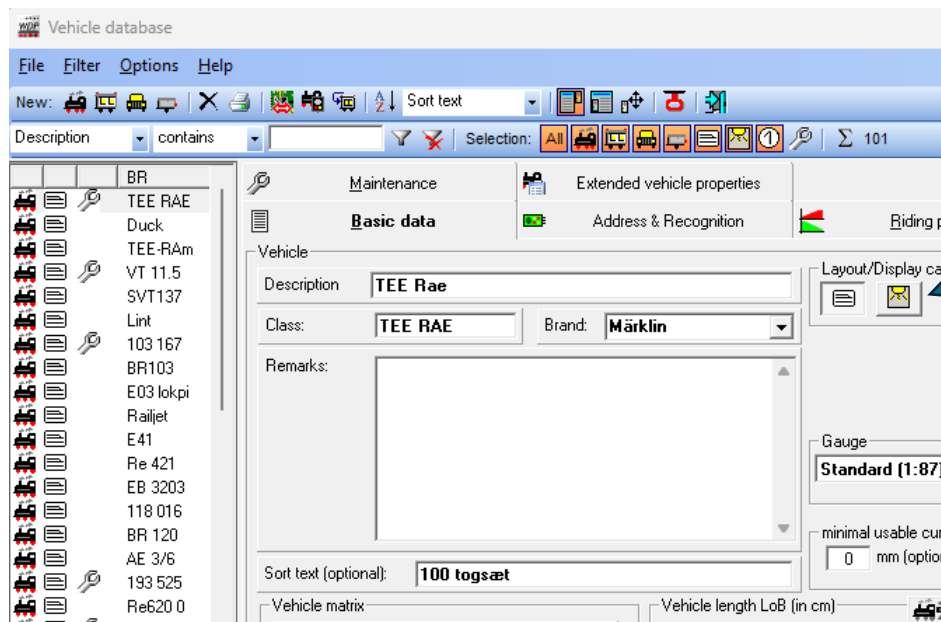
The program is used to change the status of locomotives and wagons in **Windigipet 2025** projects.

It provides a graphical user interface (GUI) where the user can move vehicles between:

- **Layout** (active on the model railway)
- **Display case** (stored, not in operation)

Status values in Windigipet's .dat files:

- **Locomotives:**
 - 0 = Layout
 - 1 = Display case
- **Wagons:**
 - 2 = Layout
 - 3 = Display case



1.2. Functionality

- **Data loading**

The program reads .dat files from the Vehicles folder of the selected Windigipet project. Files named WDVehicles_Stat_*.dat are parsed, and vehicles are loaded into two lists (Layout / Display case).

ID	Beschreibung	Standort	Baureihe	Decode	Digital	col7	col8
10	MY 1103		0 MY 1103	G	12	0	
100	Målevogn		2 Målevogn	7	33	0	
102	N 209 DSB		0 N 209 DSB	G	51	0	
104	MY 1106 GL 2		1 my 1106 G	7	57	0	
108	V200 gl		1 V200 gl	7	151	0	
109	RE4/4 I 10033		0 RE4/4 I 1	G	41	0	
110	TEE bavaria vogn		2 TEE bavar	7	161	0	
111	Gul MY		0 Gul MY	G	78	0	
113	Elektrische Lokom		0 Reihe Ae	1	68	0	

Dat – file. One file for each lok. Standort = Status

- **GUI display**

Tkinter is used to show two tables (left = Layout, right = Display case) with columns for:

- Description
- BR (Litra)

- Address

- Brand

Each vehicle is displayed with its ID and, if available, an image (from the Lokbilder folder).

- **Moving vehicles**

The user can select one or more vehicles and move them:

- → right (from Layout to Display case)
- ← left (from Display case to Layout, max 250 vehicles)

When moved, the *Standort* field (column 3) in the corresponding .dat file is updated.

- **Language support**

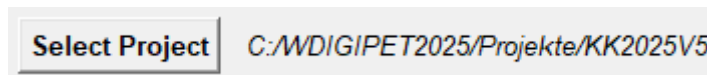
The program supports three languages (Danish, English, German).

All GUI text is updated dynamically when a flag button (DK/GB/DE) is clicked.



- **Project selection and memory**

- The user selects the Windigipet project directory through a dialog.
- The last used project path is saved in a config file (last_dir.txt) in %APPDATA%\FlytLok.



1.3. Architecture

- **Main components**

- GUI built with tkinter and ttk.
- Image handling via Pillow (PIL).
- Key functions include:
 - load_data(): Reads vehicle data from .dat files.
 - insert_row(): Inserts a vehicle into the GUI table.
 - move_right() / move_left(): Moves vehicles between Layout and Display case.
 - update_row_file(): Updates the *Standort* value in the data file.
 - sort_column(): Allows sorting of table columns.
 - update_language(): Updates all text elements in the GUI to the selected language.

- **Data handling**

- All vehicles are stored in a list (all_rows) with metadata.
- Treeview tables (Layout / Display case) are synchronized with .dat file contents.

- **File structure**

- Program file: FlytLoksV4.py
- Config file: %APPDATA%\FlytLok\last_dir.txt
- Project data: <Project>\Vehicles\WDVehicles_Stat_*.dat
- Vehicle images: <Project>\Lokbilder*.png/jpg

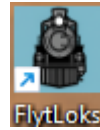
1.4. Limitations

- Maximum 250 vehicles on Layout.
- Requires correct Windigipet project structure (Vehicles folder).
- If no vehicle image is found, a gray placeholder is shown.

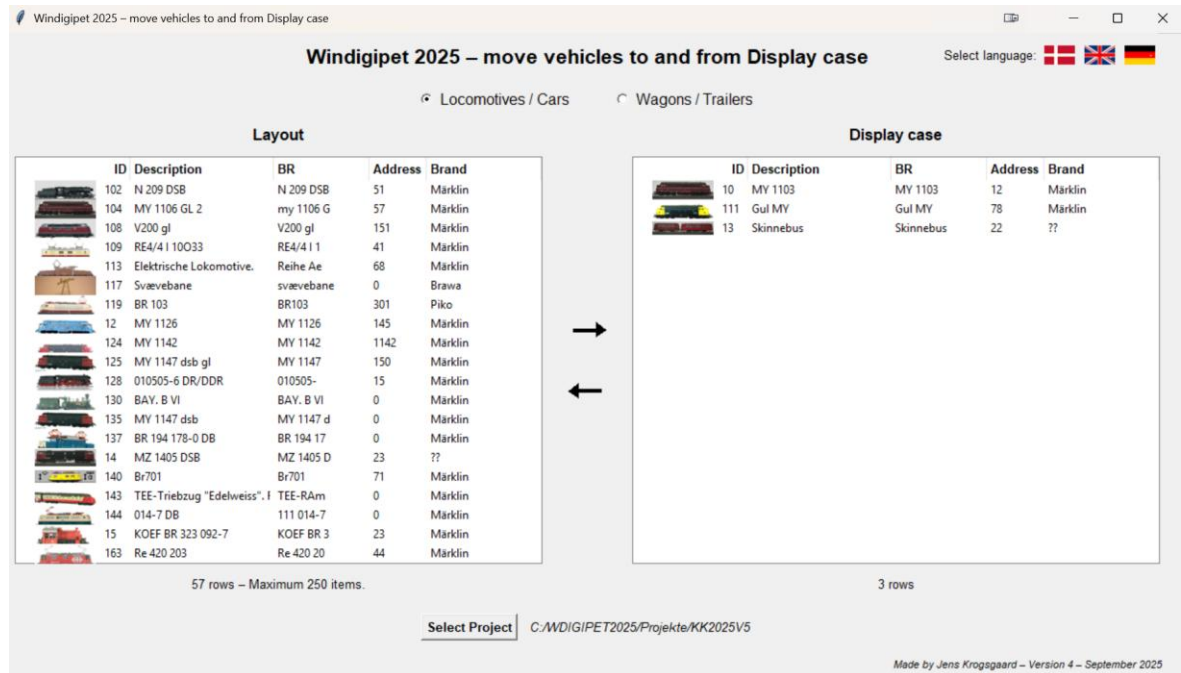
2. User Guide

Starting the program

- Click on the Icon (after installation).



- Python FlytLoksV4.py (before installation)
- The window shows two lists: **Layout** (left) and **Display case** (right).



Select a project

- Click **"Select Project"**.
- Choose the Windigipet project's main folder (not the Vehicles folder).
- The program automatically loads vehicles from the project.



Switch between Locomotives and Wagons

- Use the radio buttons at the top to select:
 - Locomotives / Cars**
 - Wagons / Trailers**



Move vehicles

- Select one or more vehicles in a list.

- Use the arrow buttons in the middle:
 - →: Move from Layout to Display case
 - ←: Move from Display case to Layout (max 250 items)

21	Lint	Lint	48	Märklin	30
24	V 60 706	V 60 706	60	Märklin	74
25	BR 50	BR 50	50	Märklin	99
28	CFL Serie 5603	Reihe S2	52	Märklin	102
87	WLE diesel	WLE diese	72	Märklin	104
108	V200 gl	V200 gl	151	Märklin	
109	RE4/4 1 10033	RE4/4 1 1	41	Märklin	
113	Elektrische Lokomotive.	Reihe Ae	68	Märklin	
117	Svævebane	svævebane	0	Brawa	

View details

Each vehicle entry includes:

- ID
- Description
- BR (Litra)
- Address
- Brand
- Optional picture (if available in Lokbilder)

Layout					
	ID	Description	BR	Address	Brand
	213	001-1 DB	Duck	0	Märklin
	128	010505-6 DR/DDR	010505-	15	Märklin
	144	014-7 DB	111 014-7	0	Märklin

Click a column header to sort the data by that column

Change language

- Click the flag icons in the top-right corner:
 - dk Danish
 - gb English
 - de German

The entire GUI switches language instantly.



Exit

- Close the program.
- The last selected project is saved and will reload automatically at next startup.

3. Distribution Guide

The program can be packaged as a **Windows installer application** with icons and language files.

Step 1 – Prepare files

- FlytLoksV4.py
- Program icon: ikon_tog.ico
- Language flags: flag_da.png, flag_en.png, flag_de.png

Step 2 – Create .exe with PyInstaller

Install PyInstaller if not already installed:

```
pip install pyinstaller
```

Build the executable:

```
pyinstaller --noconsole --onefile --name FlytLoks ^  
--add-data "flag_da.png;" ^  
--add-data "flag_de.png;" ^  
--add-data "flag_en.png;" ^  
--icon=ikon_tog.ico ^  
FlytLoksV4.py
```

The result is placed in the folder: dist\FlytLoks.exe.

Step 3 – Create Setup file (.iss)

Example setup.iss file

```
setup  
  
:  
  
[Setup]  
AppName=FlytLoks  
AppVersion=4.0  
DefaultDirName={commonpf}\FlytLoks  
DefaultGroupName=FlytLoks  
OutputDir=output  
OutputBaseFilename=setup_FlytLoks  
  
[Files]  
Source: "dist\FlytLoks.exe"; DestDir: "{app}"; Flags: ignoreversion
```

[Icons]

Name: "{group}\FlytLoks"; Filename: "{app}\FlytLoks.exe"

Name: "{commondesktop}\FlytLoks"; Filename: "{app}\FlytLoks.exe"; Tasks: desktopicon

[Tasks]

Name: "desktopicon"; Description: "Create desktop icon"; GroupDescription: "Extra icons:"

Step 4 – Compile installer

- Install **Inno Setup Compiler** from jrsoftware.org.
- Open setup.iss in the compiler.
- Select **Build → Compile**.
- In the output folder you now have: setup_FlytLoks.exe.

Step 5 – Install on other machines

- Copy setup_FlytLoks.exe to another PC.
- Run the installer – it will:
 - Install FlytLoks into C:\Program Files\FlytLoks
 - Create a Start Menu shortcut
 - Optionally create a Desktop shortcut

 You now have a fully packaged, installable Windows application that can be shared with other users.