Convert LittleNavMap Flight Plan to Mission-X

Written: Saar.N Version 1.0 20-dec-2021

Preface

Creating a mission from scratch can be a pain. Like in any project, there is the phase of gathering the information, laying down the foundations and then building on them.
In the case of a mission, you need to decide:

- The type of mission you want.
- Where you want to do it.
- What are the steps to achieve it ?
- The amount of details you would like to add so it will be complete: 3D Objects, sceneries and sound effects.

This will be translated to a set of flight legs. In Mission-X a flight leg can be a physical location where you need to reach with your plane or it with the "camera view", if you so choose, but most of the missions will be conducted from the airplane itself.

Flight plans you can find in abundance or easily create in online sites. The main problem is that there are many formats and it is hard to write a converter to all of them. This is where ***LittleNavMap*** is falling into place. It can import flight plans from known simulators and export to same format or write to its own format: "*.lnmpln*".
Fortunately LittleNavMap is a one stop shop for flight planning, it allows you to: create your flight plan, track your plane from the simulator, it has an overlay map, elevation data and much more. What I would like to do is leverage this tool and use it as a building block for the mission's flight plan.

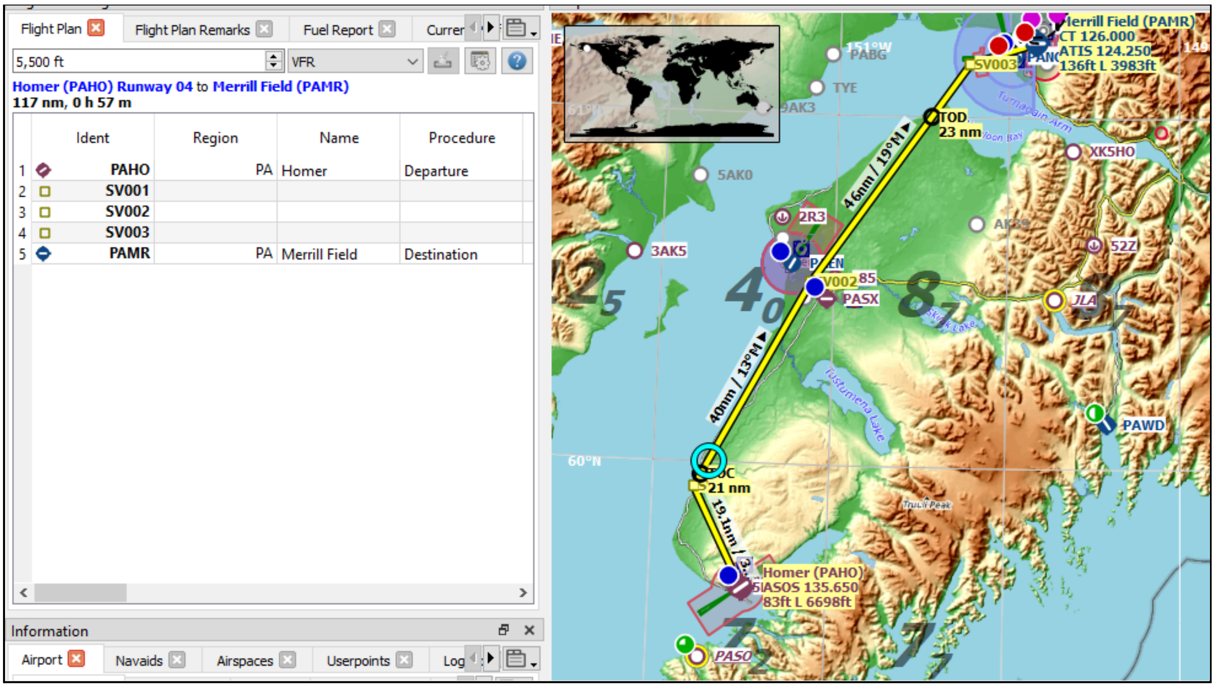
What to expect from the conversion implementation:
Currently there is a distinct separation between LittleNavMap and the plugin.
LittleNavMap: define flight plan or use it to convert other flight plans to LNM format.Mission-X Plugin: Parse LNM plan file and creates a mission file out of it.
You are expected to thicken the flight plan with some messages or special events and thus tailor it to the "adventure" you would like to fly and share with others.

Little NavMap was written by Alexander Barthel

Prepare your Flight Plan in LittleNavMap

The first step is to open LittleNavMap and prepare a new flight plan.
The flight plan can be based on a real world route, but it can also be fictional to create a “bush like adventure” (which means that some waypoints might be on rural ground).

In this example, I’m going to load a flight plan done by Chris Palmer ([Angle of Attack]()) when he embarked on his ~3000nm adventure from Homer Alaska to Oshkosh.



5,500 ft
VFR

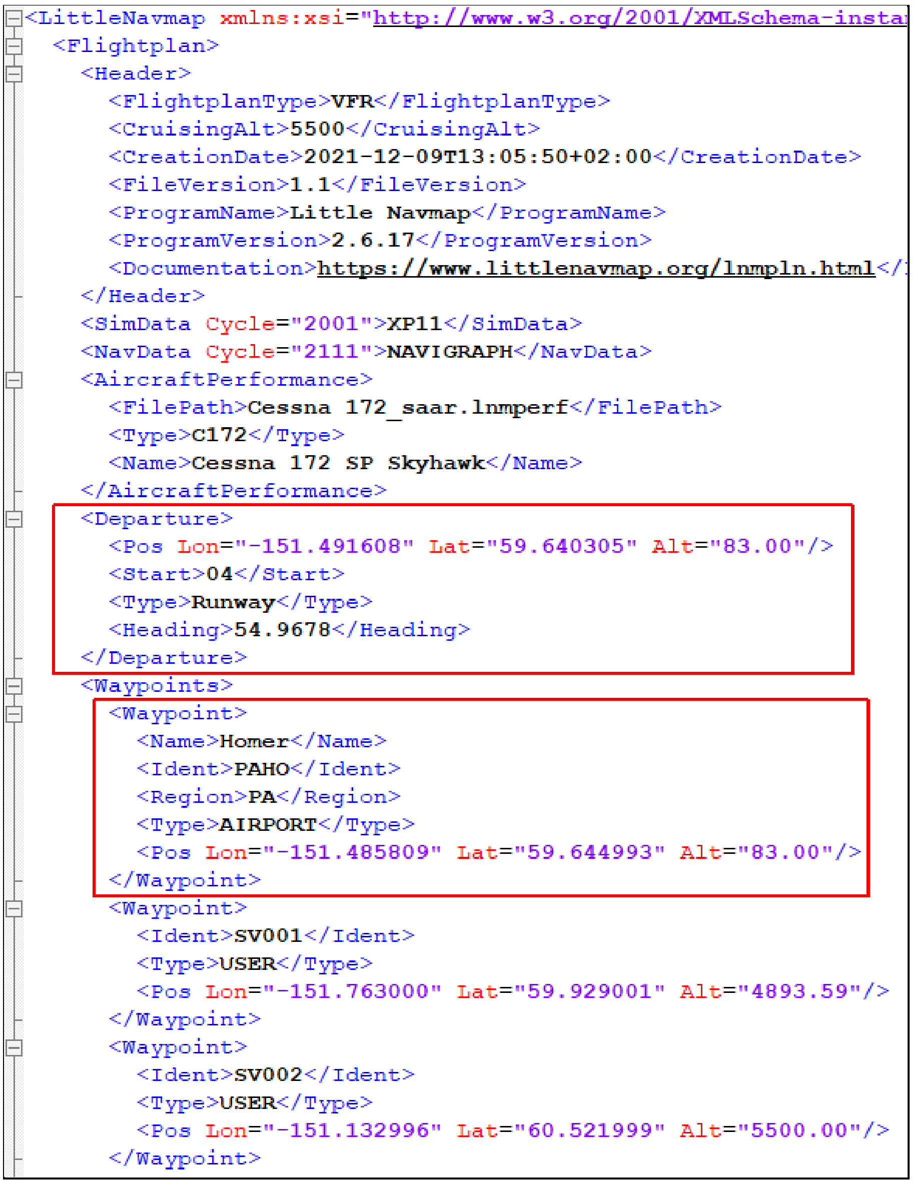
Homer (PAHO) Runway 04 to **Merrill Field (PAMR)**
117 nm, 0 h 57 m

	Ident	Region	Name	Procedure
1	PAHO	PA	Homer	Departure
2	SV001			
3	SV002			
4	SV003			
5	PAMR	PA	Merrill Field	Destination

The first part of this adventure is to fly to PAMR airport from Homer, you can see the route in LittleNavMap.
If you look closely, you can see that there are 5 waypoints, 3 of them are user created waypoints (not navaids). If you so choose you can easily tweak some of them.
Once you are ready, save the flight plan in “.lnmpln” format and copy it to “*{X-Plane}/Output/FMS plans*”

“.lnmpln” file structure

In a nutshell, the “lnmpln” file is a simple XML file that has few main elements that holds the flight plan information and are of interest to us. Let's take a quick pick to better understand it. The following screenshot is a snippet from the flight plan we saw on the previous page.



```
<LittleNavmap xmlns:xsi="http://www.w3.org/2001/XMLSchema-insta
  <Flightplan>
    <Header>
      <FlightplanType>VFR</FlightplanType>
      <CruisingAlt>5500</CruisingAlt>
      <CreationDate>2021-12-09T13:05:50+02:00</CreationDate>
      <FileVersion>1.1</FileVersion>
      <ProgramName>Little Navmap</ProgramName>
      <ProgramVersion>2.6.17</ProgramVersion>
      <Documentation>https://www.littlenavmap.org/lnmpln.html</
    </Header>
    <SimData Cycle="2001">XP11</SimData>
    <NavData Cycle="2111">NAVIGRAPH</NavData>
    <AircraftPerformance>
      <FilePath>Cessna 172_saar.lnmperf</FilePath>
      <Type>C172</Type>
      <Name>Cessna 172 SP Skyhawk</Name>
    </AircraftPerformance>
    <Departure>
      <Pos Lon="-151.491608" Lat="59.640305" Alt="83.00"/>
      <Start>04</Start>
      <Type>Runway</Type>
      <Heading>54.9678</Heading>
    </Departure>
    <Waypoints>
      <Waypoint>
        <Name>Homer</Name>
        <Ident>PAHO</Ident>
        <Region>PA</Region>
        <Type>AIRPORT</Type>
        <Pos Lon="-151.485809" Lat="59.644993" Alt="83.00"/>
      </Waypoint>
      <Waypoint>
        <Ident>SV001</Ident>
        <Type>USER</Type>
        <Pos Lon="-151.763000" Lat="59.929001" Alt="4893.59"/>
      </Waypoint>
      <Waypoint>
        <Ident>SV002</Ident>
        <Type>USER</Type>
        <Pos Lon="-151.132996" Lat="60.521999" Alt="5500.00"/>
      </Waypoint>
```

The information we look for is:

* **[optional] Departure**: In LittleNaMap this must be a navaid. If the first waypoint is user based and not a valid departure location, then it won’t be available in the flight plan file (which is fine, it is not mandatory).

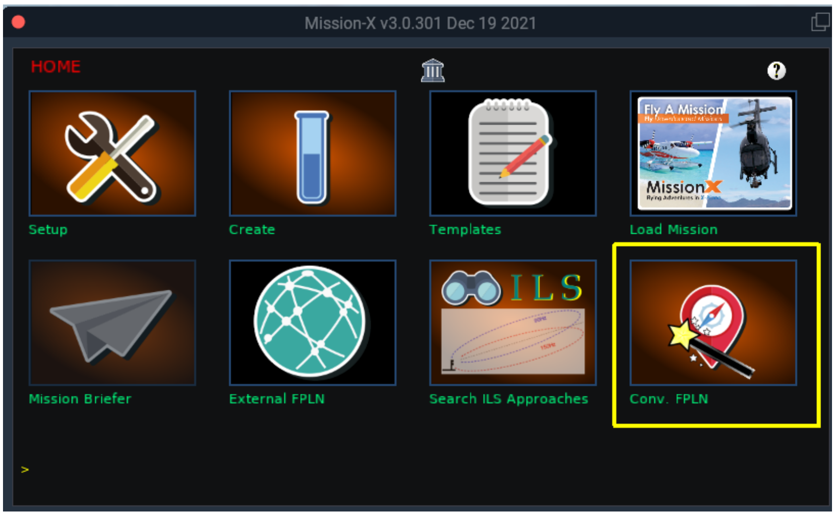
* **Waypoint:** We construct a mission from waypoints.
At least one of them will have to be mandatory, the filtering and decision we will do in the plugin.

Loading the ".lnmpln" flight plan into Mission-X

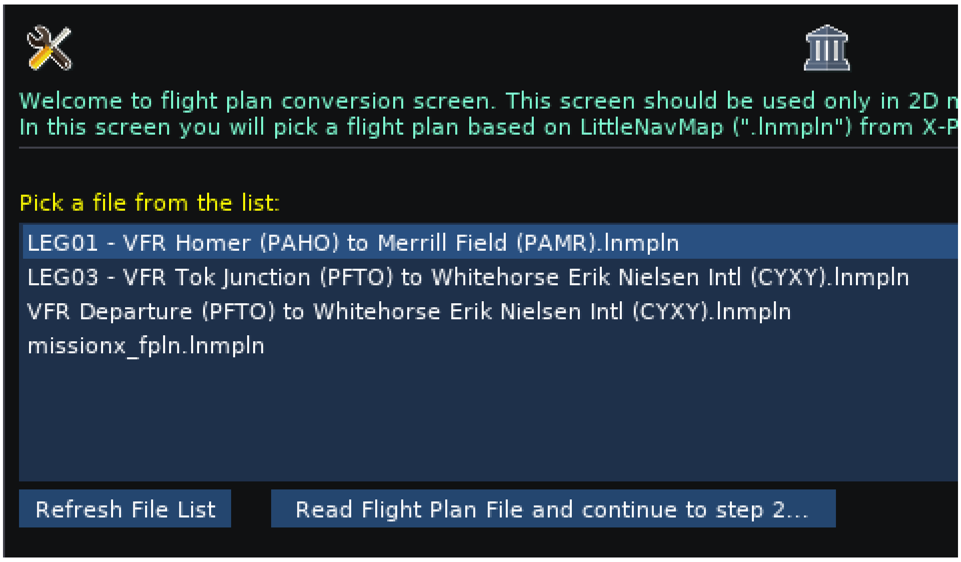
Now that we better understand which part of the LittleNavMap flight plan we read, we can move forward and load it to the mission conversion screen.

Steps to load and convert the mission

- Copy the flight plan to the XP11 "*FMS plans*" folder, so the plugin will be able to load it.
- Start X-Plane and open Mission-X conversion screen



- The landing page will list all flight plans with the extension ".lnmpln".
  You can refresh the page list using the [refresh file list] button.



Welcome to flight plan conversion screen. This screen should be used only in 2D n
In this screen you will pick a flight plan based on LittleNavMap (".lnmpln") from X-P

Pick a file from the list:

LEG01 - VFR Homer (PAHO) to Merrill Field (PAMR).lnmpln
LEG03 - VFR Tok Junction (PFTO) to Whitehorse Erik Nielsen Intl (CYXY).lnmpln
VFR Departure (PFTO) to Whitehorse Erik Nielsen Intl (CYXY).lnmpln
missionx_fpln.lnmpln

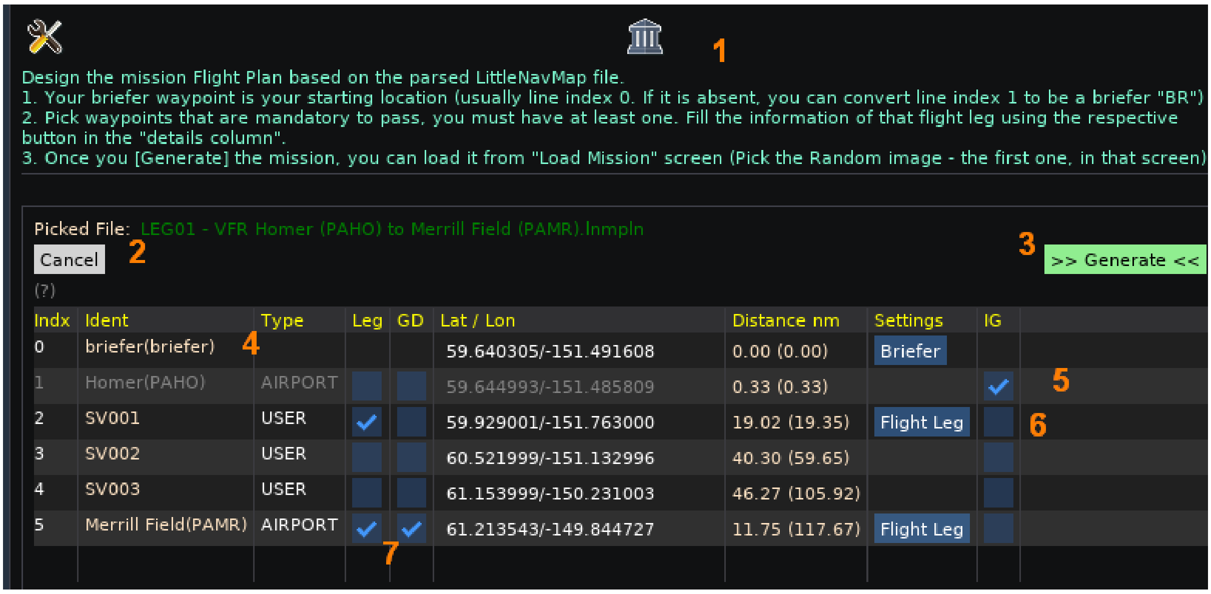
In this example we will pick the first file and click the [... continue to step 2] button.

Main design page

The mission design page will allow you to

- Add briefing details to the mission.
- Choose which waypoints you would like to have as mandatory, which to ignore and which to just have as part of the route.
- For each mandatory waypoint (flight leg) you can add information and few actions.

But first things first, let's have a look at the design page and explain its parts:



Design the mission Flight Plan based on the parsed LittleNavMap file.
1. Your briefer waypoint is your starting location (usually line index 0. If it is absent, you can convert line index 1 to be a briefer "BR")
2. Pick waypoints that are mandatory to pass, you must have at least one. Fill the information of that flight leg using the respective button in the "details column".
3. Once you [Generate] the mission, you can load it from "Load Mission" screen (Pick the Random image - the first one, in that screen)

Picked File: LEG01 - VFR Homer (PAHO) to Merrill Field (PAMR).lnmpln

(?)

Indx	Ident	Type	Leg	GD	Lat / Lon	Distance nm	Settings	IG
0	briefer(briefer)				59.640305/-151.491608	0.00 (0.00)	Briefer	
1	Homer(PAHO)	AIRPORT	□	□	59.644993/-151.485809	0.33 (0.33)		☑
2	SV001	USER	☑	□	59.929001/-151.763000	19.02 (19.35)	Flight Leg	□
3	SV002	USER	□	□	60.521999/-151.132996	40.30 (59.65)		□
4	SV003	USER	□	□	61.153999/-150.231003	46.27 (105.92)		□
5	Merrill Field(PAMR)	AIRPORT	☑	☑	61.213543/-149.844727	11.75 (117.67)	Flight Leg	□

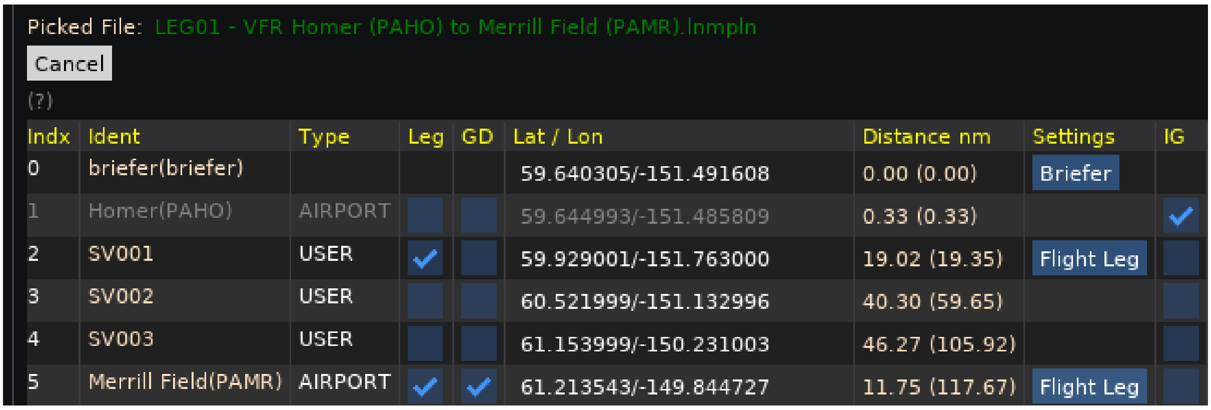
Legend:

1	Short explanation of this screen.
2	The cancel button will clear your work, and you will go back to the file list screen.
3	The generate button creates the mission file. As of this writing it will create a “random.xml” file in the “random/briefer” folder.
4	If you have the <Departure> tag in “.lnmpln” flight plan, then you will have a ***briefer*** waypoint as first row. It also represents your starting location.
5	You can decide to ignore a waypoint (it won’t be present in the GPS).
6	A mandatory waypoint will have a “flight leg” button settings to add more information to this special waypoint.
7	This is an example of another mandatory waypoint that represents the ending location in the flight plan. We can also see that we checked the GD field which means we need to be on ground.

After generating a mission file, you will be able to load it from the “**Load Mission”** screen. Pick the Random mission image and fly the route (this will be addressed in later builds).

The table columns

Table columns explanation:



Picked File: LEG01 - VFR Homer (PAHO) to Merrill Field (PAMR).lnmpln

Cancel

(?)

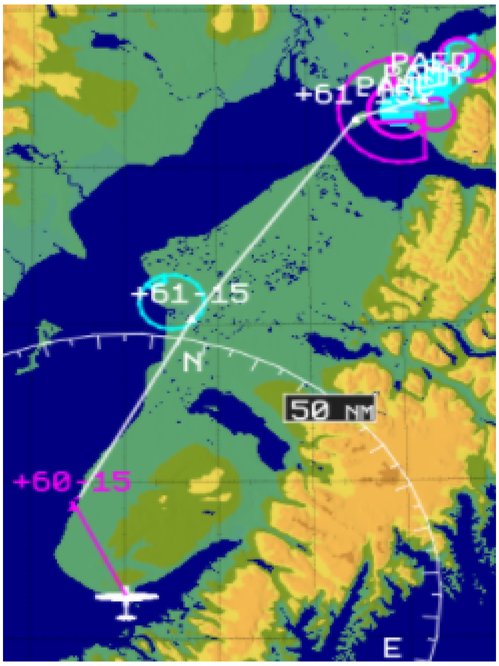
Indx	Ident	Type	Leg	GD	Lat / Lon	Distance nm	Settings	IG
0	briefer(briefer)				59.640305/-151.491608	0.00 (0.00)	Briefer	
1	Homer(PAHO)	AIRPORT	☐	☐	59.644993/-151.485809	0.33 (0.33)		✓
2	SV001	USER	✓	☐	59.929001/-151.763000	19.02 (19.35)	Flight Leg	☐
3	SV002	USER	☐	☐	60.521999/-151.132996	40.30 (59.65)		☐
4	SV003	USER	☐	☐	61.153999/-150.231003	46.27 (105.92)		☐
5	Merrill Field(PAMR)	AIRPORT	✓	✓	61.213543/-149.844727	11.75 (117.67)	Flight Leg	☐

The columns provide the following information:

Indx	Each waypoint receives a unique number. The briefer will always have a zero index number. The briefer represents the <Departure> element. If there is none, then the index will start from number 1.
Ident	The is a mandatory field in the flight plan file, so it can represent a navaid, like airport or NDB or a user base point, like SV001.
Type	Optional: The type of the waypoint based on “lnmpln” info.
Leg	If *checked*, then this waypoint is ***mandatory***, and it will build a flight leg rule in the mission file. Example, “SV001” is the first leg, hence our first leg is: from starting location - “briefer” to “SV001”.
GD	On Ground, This is a quick way to tell the plugin to expect the plane to be on ground when arriving at this *mandatory waypoint*. If you want to check some more elevation rules, open the “flight leg” settings at that same row.
BR	[Briefer] Not present in this screenshot, but allows the first waypoint to act as a briefer - when the briefer line is not present (see special case below)
Lat/Lon	Location of the waypoint.
Distance nm	The distance between previous waypoint and the current one (and cumulative since the start of the flight plan)
Settings	If a waypoint is mandatory, you will have the option to add more information to it.
IG	Ignore this waypoint

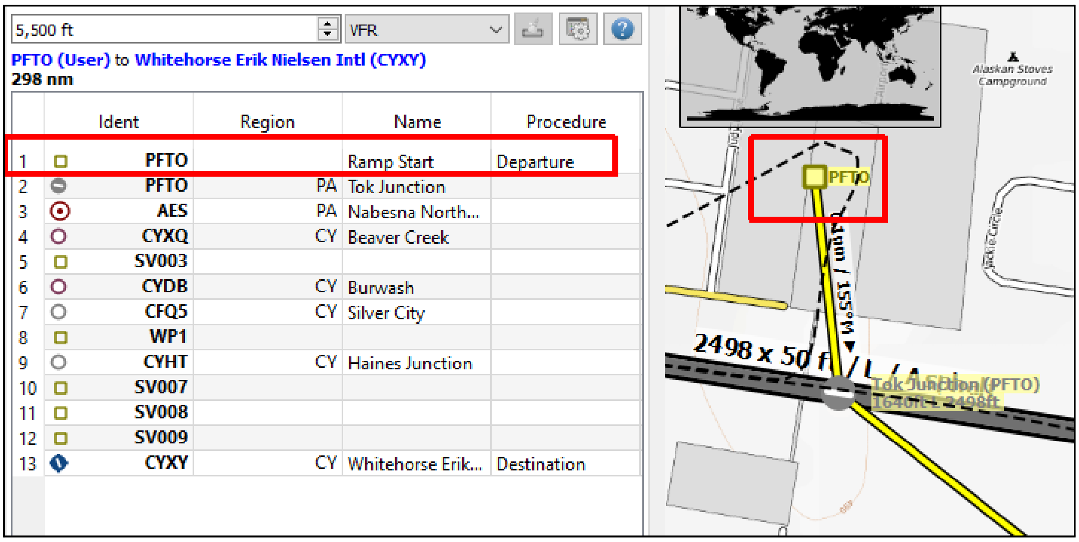
The table provides a high level of the route we will fly. The mandatory waypoints represent flight legs, while the rest of the waypoints, that are not being ignored, will be part of the waypoints in the GPS.

And here is how it looks in the X-Plane G1000:



Special Case

In some cases, the flight plan in LittleNavMap won't have a “valid” departure location, meaning it will be a user defined waypoint or a navaid that is not an airport. In such cases we will only have the <waypoint>s but not the <departure> tag.



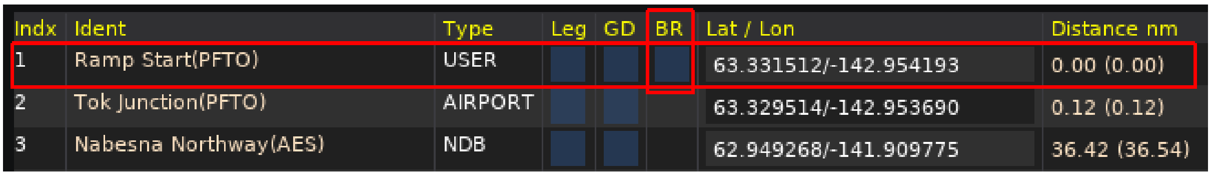
5,500 ft — VFR

PFTO (User) to **Whitehorse Erik Nielsen Intl (CYXY)**
298 nm

		Ident	Region	Name	Procedure
1	□	PFTO		Ramp Start	Departure
2	⊖	PFTO	PA	Tok Junction	
3	⊙	AES	PA	Nabesna North...	
4	○	CYXQ	CY	Beaver Creek	
5	□	SV003			
6	○	CYDB	CY	Burwash	
7	○	CFQ5	CY	Silver City	
8	□	WP1			
9	○	CYHT	CY	Haines Junction	
10	□	SV007			
11	□	SV008			
12	□	SV009			
13	◆	CYXY	CY	Whitehorse Erik...	Destination

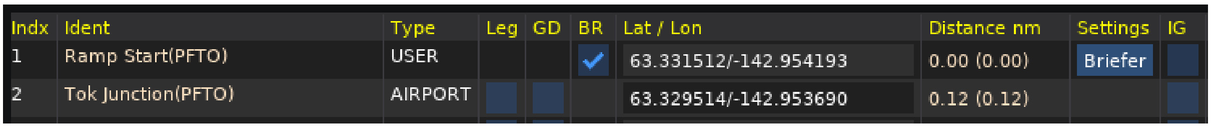
In the screenshot above you can see that the “departure” waypoint is a “ramp start”. In such cases LittleNavMap won’t create a <Departure> tag in the flight plan file.

Here how it will look in the conversion table screen:



Indx	Ident	Type	Leg	GD	BR	Lat / Lon	Distance nm
1	Ramp Start(PFTO)	USER	□	□	□	63.331512/-142.954193	0.00 (0.00)
2	Tok Junction(PFTO)	AIRPORT	□	□		63.329514/-142.953690	0.12 (0.12)
3	Nabesna Northway(AES)	NDB	□	□		62.949268/-141.909775	36.42 (36.54)

You can see that we do not have “index 0” in the table, instead we have “row 1” with a new column “**BR**”, which means: make this row a briefer, and thus you could refer it as the starting location.



Indx	Ident	Type	Leg	GD	BR	Lat / Lon	Distance nm	Settings	IG
1	Ramp Start(PFTO)	USER			☑	63.331512/-142.954193	0.00 (0.00)	Briefer	□
2	Tok Junction(PFTO)	AIRPORT	□	□		63.329514/-142.953690	0.12 (0.12)		□

Once we check “first row” it can no longer be a flight leg, and we get a [briefer] button in the “settings” column.

If you won’t make the first line a briefer, that is also legit, and the plugin will use its coordination as the starting location, but you will have a default briefer information constructed by the plugin.

Are we there yet ?

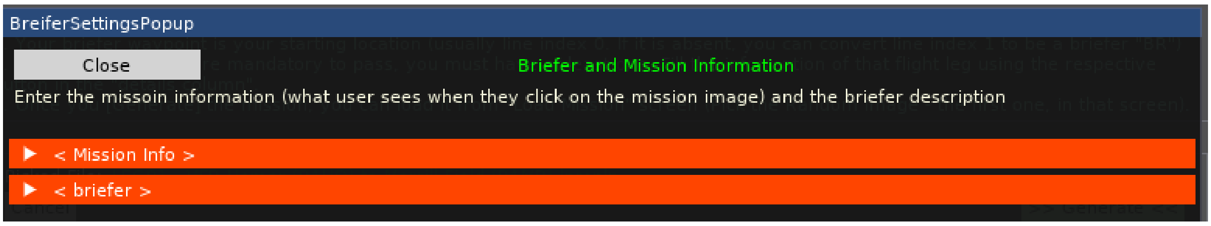
Not quite, if we want to turn this flight plan to a mission we should fill in some more information to add more interest in the mission, like:

- Briefer information
- En-Route description for the mandatory flight legs.
- Messages at the beginning or flight leg and when reaching the waypoint.

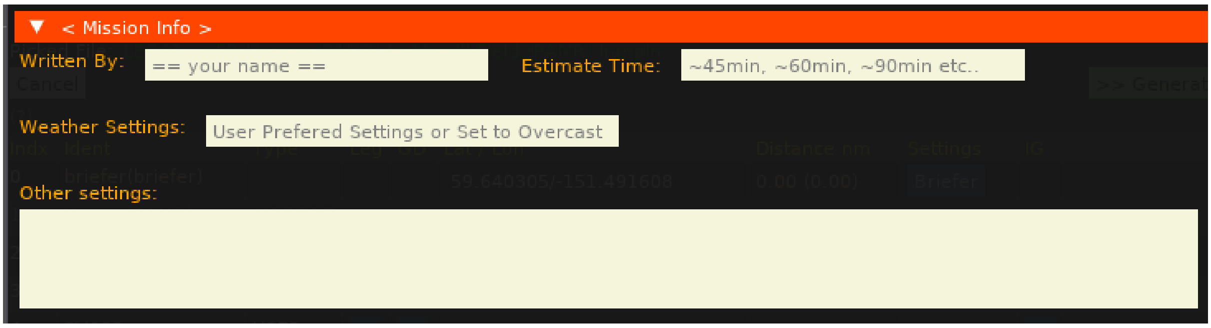
Although all information is optional, the plugin generates far less desirable messages. In the next few pages we will visit the “setting” screen for the briefer and flight leg.

The Briefer Settings Screen

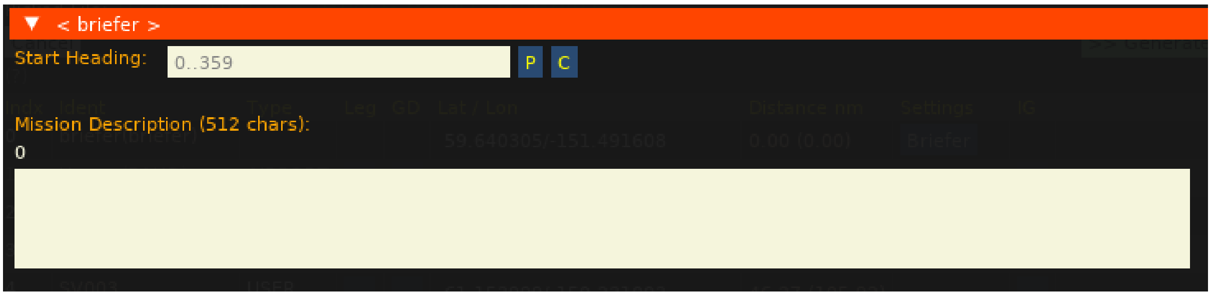
The briefer popup settings has two main topics that you should fill in:
The “**Mission Info**” where you fill in some meta information regarding the mission and the “**Briefer**” where you will fill in the mission details and the start heading information.



Enter the missoin information (what user sees when they click on the mission image) and the briefer description



Written By: == your name ==
Estimate Time: ~45min, ~60min, ~90min etc..
Weather Settings: User Prefered Settings or Set to Overcast
Other settings:



Start Heading: 0..359

Mission Description (512 chars):
0

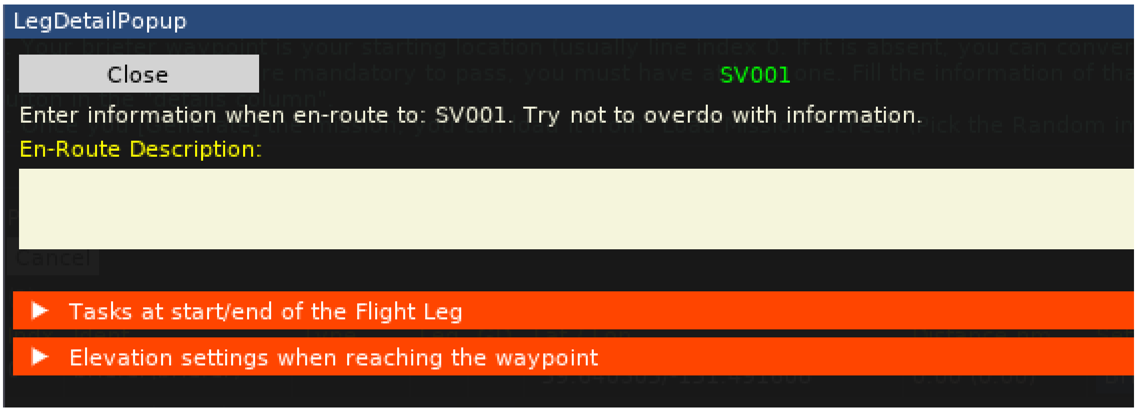
The [P] and [C] buttons stand for [P]lane heading and [C]amera view heading from X-Plane, this will allow for precise positioning.

The Flight Leg Settings Screen

The flight leg settings provide the option to fill in information regarding:

- En-Route description: special insight tasks that you would like to highlight.
- Auto message at the beginning of the flight leg
- Auto message when arriving at the waypoint.
- Fire commands at the beginning and end of the flight leg.

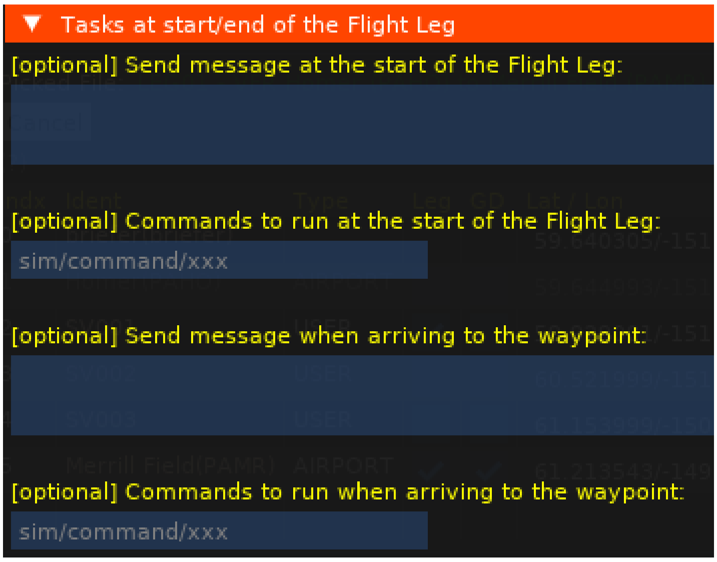
All of this is optional, but let you add more interest to the mission.
Remember, when starting a flight leg, you are, in most cases, en-route to the waypoint, so your description should reflect that.


LegDetailPopup
Close
SV001

Enter information when en-route to: SV001. Try not to overdo with information.

En-Route Description:

▶ Tasks at start/end of the Flight Leg
▶ Elevation settings when reaching the waypoint



[optional] Send message at the start of the Flight Leg:

[optional] Commands to run at the start of the Flight Leg:

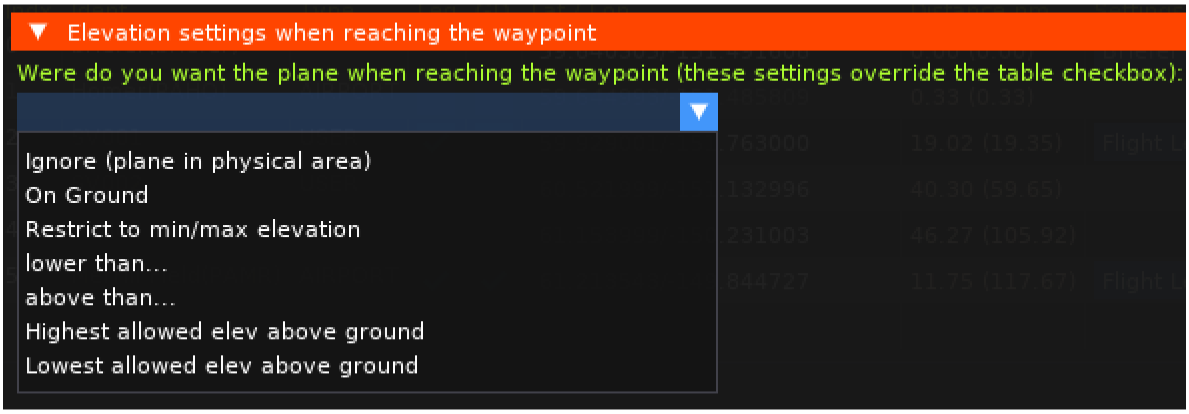
sim/command/xxx

[optional] Send message when arriving to the waypoint:

[optional] Commands to run when arriving to the waypoint:

sim/command/xxx

All the information in the first topic is optional, but they can add to the mission atmosphere.



Elevation settings when reaching the waypoint

Were do you want the plane when reaching the waypoint (these settings override the table checkbox):

Ignore (plane in physical area)
On Ground
Restrict to min/max elevation
lower than…
above than…
Highest allowed elev above ground
Lowest allowed elev above ground

As of this writing, a target waypoint represents an area of 2000m radius where the plane should physically be, and 200m radius when on ground.
This means that the Elevation settings are relevant when the plane is reaching the target waypoint and the plugin needs to evaluate if we are following the rules of the “En-Route description”.

The elevation rules:

Ignore	**Default** - plane just needs to be in the physical area of the waypoint (2000m when airborne and 200m when on ground)
On Ground	On the ground (self explanatory)
Restrict min/max elevation	You want the plane to be in certain volume elevation, for example ***higher than 2000 but lower than 5000***
Lower than…	Define the highest elevation the plane can fly and simmer can fly in any elevation ***below*** it.
Above than	Define the lowest elevation the plane can fly and simmer can fly in any elevation ***above*** it.
Highest allowed elev above ground	This represents a dynamic elevation rule. Example: We don’t know the exact elevation above sea level, but we want the plane to fly ***no more than 3000ft above the ground***,when arriving at this waypoint.
Lowest allowed elev above ground	This represents a dynamic elevation rule. Example: We don’t know the exact elevation above sea level, but we want the plane to fly ***more than 2000ft above the ground***,when arriving at this waypoint.

For each option you will have a slider like input to define the elevation rule at hand.

Remember: Elevation rules are optional and you can ignore them, but it is much more entertaining and challenging when we add them.
Do remember to explain these rules in the “En-Route” description.

Pros and Cons

The current implementation of the flight plan conversion highlight few design decisions that might be addressed in the future (no promises):

Pros

- Easy flight plan conversion() if it is “.lmnpln”.
- Plugin creates a valid mission file you can enhance later on.
- During the same X-Plane session, You can generate the same flight plan as much as you want as long as you do not “cancel” your work or exit X-Plane (You can even fly it and quit the mission and still your conversion settings will be intact).
- Easy UI (imho).

Cons

- Only convert the LittleNavMap flight plan.
- The conversion is only one way street, from “LittleFlightMap” to “Mission-X” and you can’t later on modify the converted file in the plugin, only from an editor.
- Many aspects like “datarefs” and “script/scriptlet” were not implemented but you can add them after generating the mission file.
- Once you press the “generate” button, you override the last random.xml file, unless you manually backup the previous generated file.

The flight plan conversion feature is a way to lay the foundation to the adventure you want to share. It allows basic messaging and flight rules but it is yet far from being a full editor, and it probably won't be.

Blue Skies
Saar.N