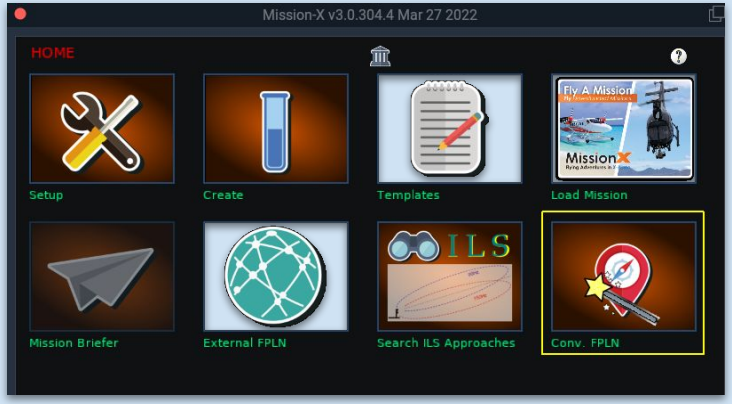
Mission-X Conversion Screen

In a Nutshell

Written by Saar.N

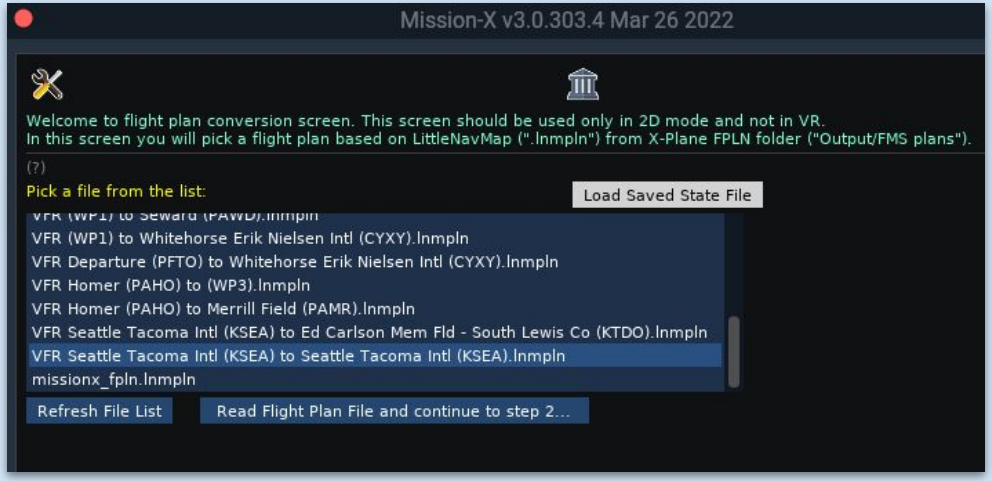
Revision 1.0

Where can I find the conversion screen



You can find the LittleNavMap FPLN conversion screen in the “home” screen of Mission-X.

Pick your Flight Plan to convert



- The first screen you will face is the list of LittleNavMap flight plan files: ".lnmpln" that are located in "*{X-Plane}/Output/FMS plans*" folder.

  You will have to copy the files if not done beforehand.

- You can pick one of the files and press the [*Read Flight Plan...*] button or

- You can load the last saved state file.

Flight Plan as a Table of Waypoints - opt .1

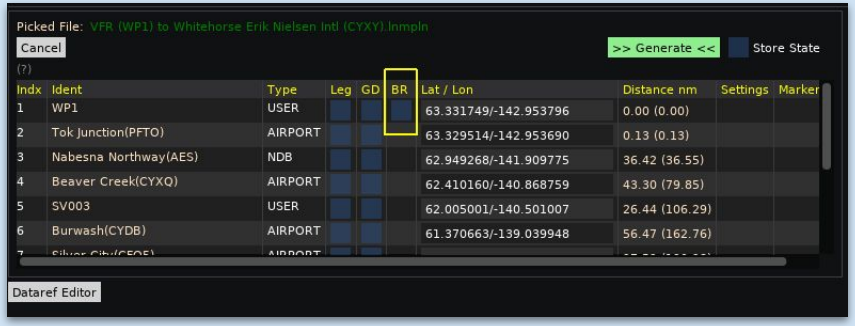
Indx	Ident	Type	Leg	GD	Lat / Lon	Distance nm	Settings	Marker	IG
0	briefer				47.463799/-122.307755	0.00 (0.00)	Briefer		
1	Seattle Tacoma Intl(KSEA)	AIRPORT			47.449890/-122.311775	0.85 (0.85)			
2	WP1	USER			47.453896/-122.313553	0.25 (1.10)			
3	WP2	USER			47.473068/-122.313263	1.15 (2.26)			
4	Seattle Tacoma Intl(KSEA)	AIRPORT			47.449890/-122.311775	1.39 (3.65)			

- When you first load a flight plan, you will see a list of waypoints listed in a table.
- Each row represents one waypoint you defined in LittleNavMap.
- The starting location should be an airport and you will see the name: “briefer” instead.
  The briefer represent your starting location. This is not always ideal, since you might want to start from a specific ramp/terminal location while airport location might imply a runway by LittleNavMap.

Once you define at least one mandatory “leg” you can [***Generate***] a random.xml mission file or even “store” your work for later modification.

Tip: ***The conversion screen is unique due to the fact that it's available during active mission, that way you can modify your work on the fly.***

Flight Plan as a Table of Waypoints - opt .2



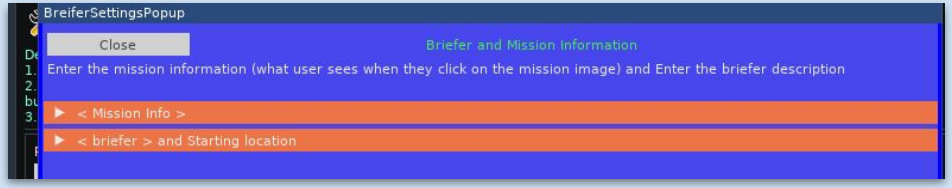
In some cases, you might set the first waypoint in LittleNavMap to be a coordinate based (lat/lon) and not an airport, in such cases the first waypoint will have a [BR] check box field, which you can flag and convert to a briefer.

This means, based on the image on the left, that “WP1” waypoint is your starting position and you can fill in all your briefer text and rules as if it was a “native” briefer (see previous page).

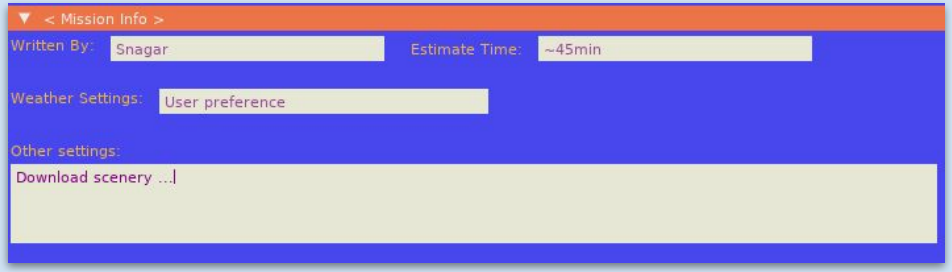
Tip:

- ***A Native briefer row index is always zero (0).***
- ***As of this build (v3.0.304.4), you can’t change waypoint position (lat/lon) from the conversion screen - this should be done inside LittleNavMap application.***

Briefer Screen

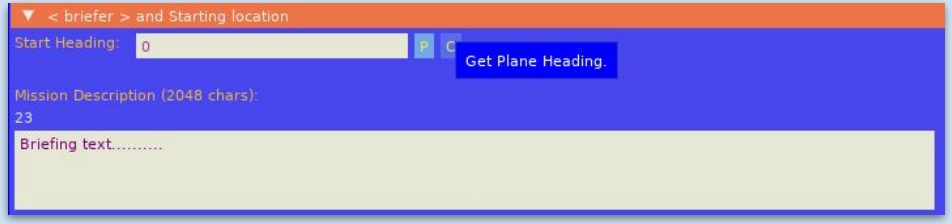


The briefer popup screen lets you fill in data regarding your Mission Information and the Briefer text.



The Mission Information holds statistics and introduces the mission to the end user, data like:

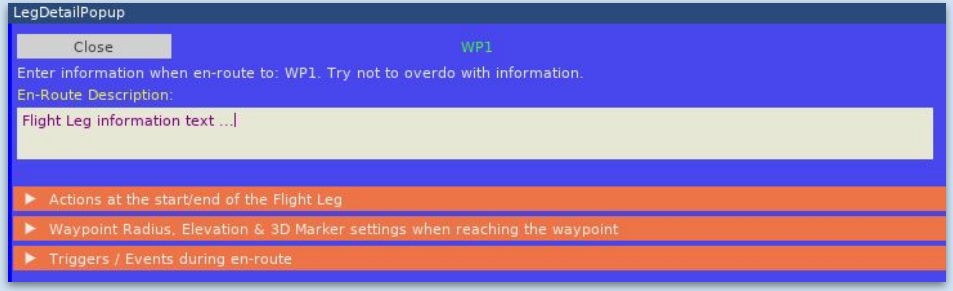
- Who design the mission
- Mission time estimation.
- Weather and
- Other useful information like scenery or plane requirements.



The Briefer section allows you to:

- Pick plane heading - using the [**p**]lane/[**c**]amera heading.
- Write the mission briefing text. Try to make it as detail as you can.

Flight Plan Screens

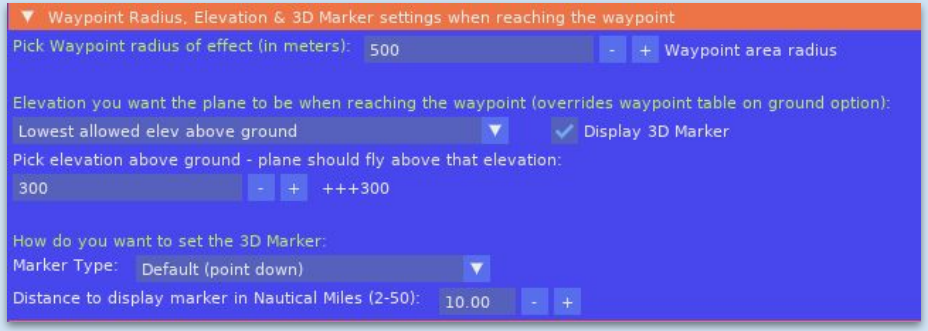


The Flight Plan, popup screen allows you to define the "Enroute flight leg" description text and basic behaviours that will take part during this leg.

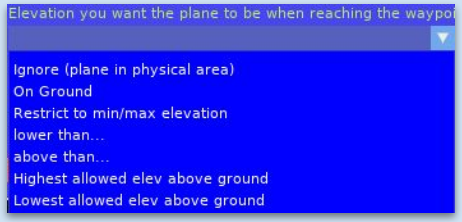
The Action section lets you:

- Define messages at start and end of each flight leg.
  These are optional messages
- Define commands you would like to automatically fire during start and end of the "flight leg"
  These are optional commands
  *Multiple commands must be divided by comma separator "**,**".*

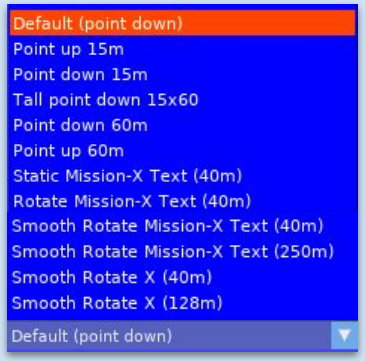
Flight Plan Screens



Elevation Rules List



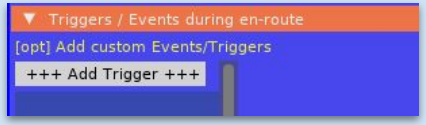
3D Marker List

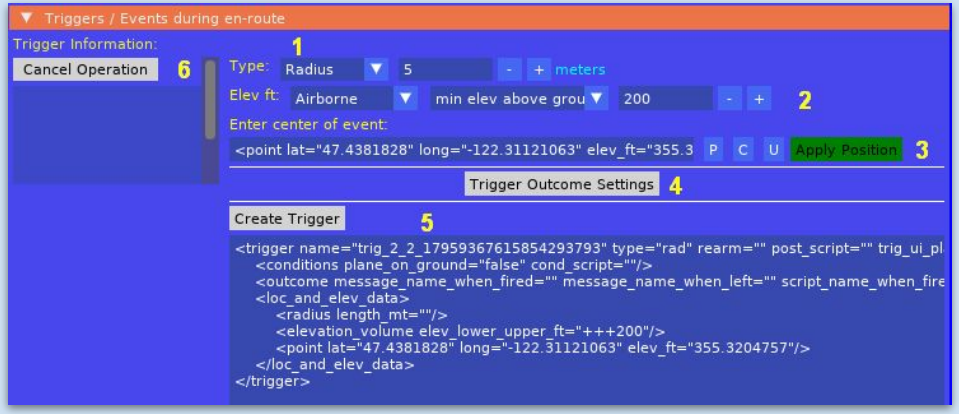


The "Waypoint Radius, Elevation & 3D Marker" section allows you to:

- Define the target waypoint area as a radius in meters.
- Define elevation restrictions - where you want the plane to be when it approaches the area, do you want it to be on the ground, or in specific elevation volume or just above certain elevation rule so the flight leg will be flagged as success.
- Define optional 3D marker as a hint to the simmer and it's visibility range.

Flight Plan Custom Triggers

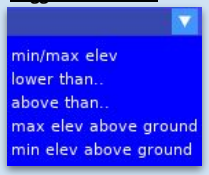




Trigger Types

Elev. Pos

Trigger Elev.Rrules

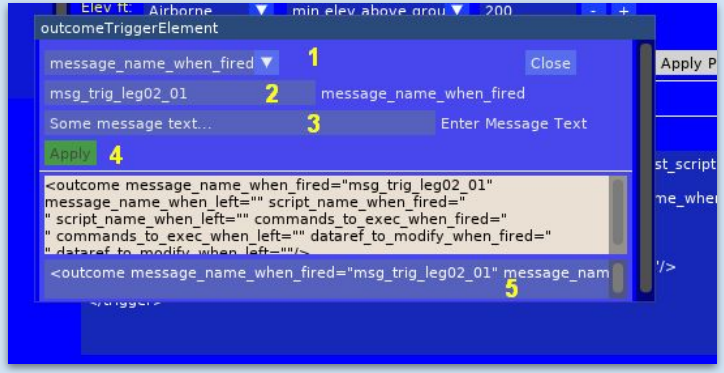


The Custom "Triggers / Event" section allows you to:

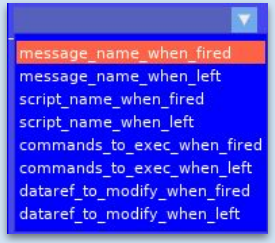
- Add event areas to interact with the simmer in addition to the waypoint.
- There are 3 types of triggers (1), radius, box and script based. For ***radius*** based, you need to define an acceptable area as a radius. For ***Box***(poly) type you will define a box area, and in a ***script*** type you will have to manage the test rules in a script plus setting the "trigger" successful property.
- Another characteristic of an event that is Radius/Box type based is the elevation (2) and its position in the world (3).
  You need to define the position (3) of the trigger and its elevation rules once the plane reach the trigger area (2).
- To make the position valid, you have to pick it using the "[**P**]lane/[**C**]amera or [**U**]ser" buttons and then press the [*Apply Position*] button to store it.
- The [***Trigger Outcome Settings***] button will allow you to define actions once the trigger is fired.

Continue to next page for more information...

Flight Plan Custom Triggers - Outcome Screen



Trigger Outcome Option List

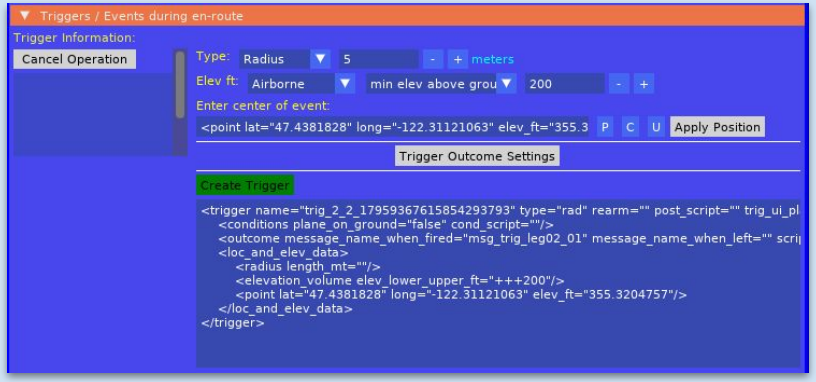


The Outcome popup screen allows you to:

- Define the expected action to send to the user or to run a code for more complex outcome.
- (1)The types of actions you can define are:
  Messages, Commands, Dataref manipulation and scripts.
- The most used action is the "***message name when fired***" that is triggered once the plane enters the trigger area.
  You define the name of the message and a message text.
  To remove a message, you delete the message text content and press the [***Apply***] button.
- The **[*Apply*]** button will save your definition into the <outcome> element and it will be visible in the multiline widget (4,5).
- If you want to copy the content of the <outcome> text, you can do that using the lower input widget (5).

Continue to next page for more information...

Flight Plan Custom Triggers - Creating the trigger



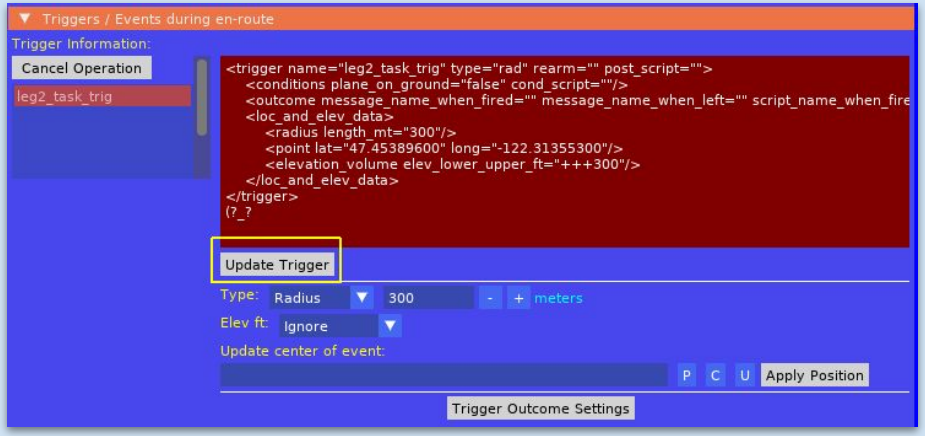
Once you define the

- Type of trigger
- The elevation rules
- Position and Outcome

You can click the [***Create Trigger***] button.

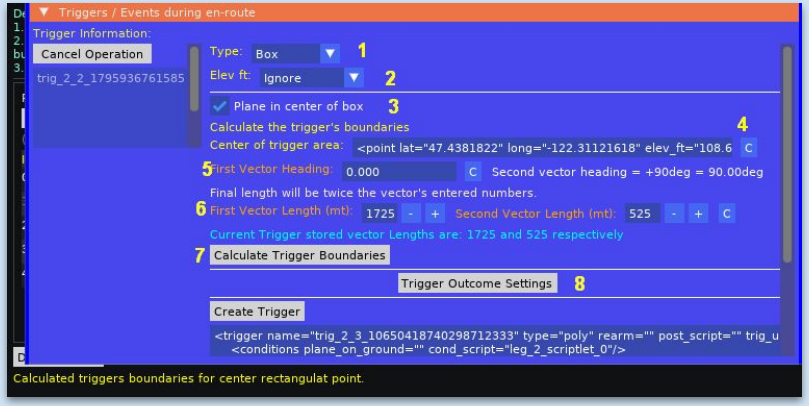
Once that is done, you will see the name of the trigger at the left side of the screen.

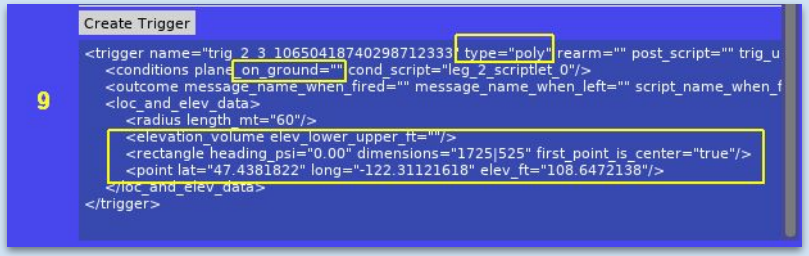
Flight Plan Custom Triggers - Updating a Trigger



- To Update a trigger, just click on it.
- The right side of the screen will display the current trigger XML settings at the top, as oppose to a new trigger that is displayed at the bottom of that screen.
- The screen allow you to update parts of the trigger, so you will have to decide if to redefine the trigger itself or just alter parts of it (*use the xml as a reference to your original values*).
- Once you are done, you will have to press the [***Update Trigger***] button so changes will take effect and will override the XML part.

Flight Plan Custom Triggers - Box/Poly Trigger

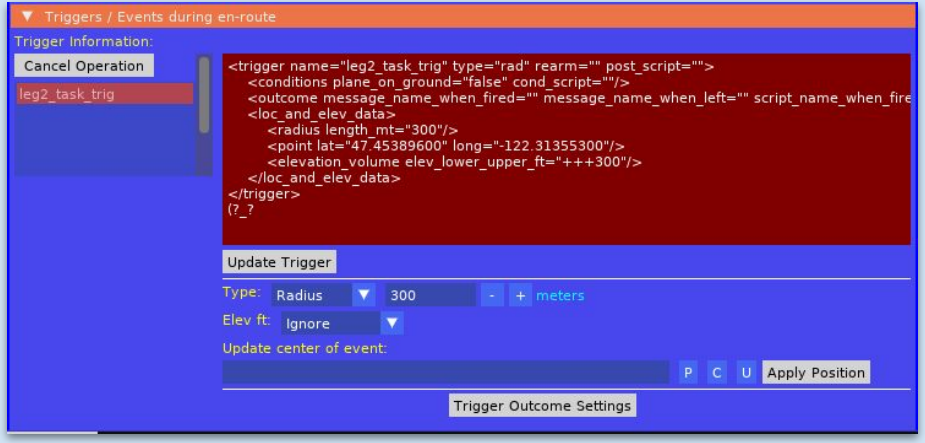




The polygon based trigger - or "Box" type trigger (1) allow you to define a polygonal boxed area using simple rules of location and vector lengths.

- In this example, we ignore the elevation rules (2) and we pick a location (4) which will be the "center" (3) of the polygonal area (box).
  We define the heading of the first vector (5) and then we define the length of each vector (6).
  In this example it is *1725x525* meters.
  Once that is done, we should press the [***Calculate Trigger Boundaries***] to implement the rules into the <trigger> element (9).
- You should also define the "Trigger Outcome" (8) as we saw in previous pages.
- If all is valid, you can press the [***Create Trigger***] button to store the trigger.

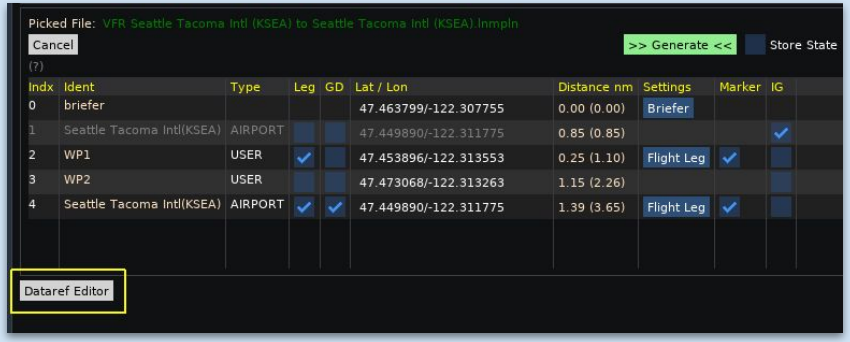
Flight Plan Triggers - Summary

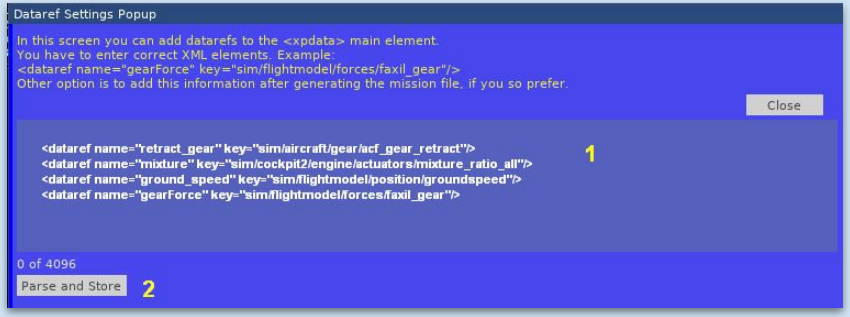


- Custom triggers, or events, allows you to spice the mission for the end user, it can provide explanation messages to point the simmer to the right direction or to encourage them in their flight.
- When you will feel that you better understand the concept you can continue and enhance the outcome of a trigger using commands, datarefs or even scripts.
- The naming of custom triggers starts with "***trig_xxx***" while waypoint triggers are named "***legN_task_trig***". Do not alter those names in the "conversion.sav" save file since the plugin might fail loading it.

- You can build a pretty comprehansive mission without adding custom triggers.
- You can always modify generated mission and make it complex as you see fit.
  My suggestion is to clone the "random" folder, rename it and work on the generated mission as a new mission pack, just remember to: "rename the mission XML file", "provide a new mission image" and "rename the title and name" of the mission inside the XML file as a start.
- In the example above, you can see the "generated waypoint" target trigger that was created by the plugin for the mandatory waypoint. You can alter it at your own discretion, my suggestion is to modify what you can from the other sections and not directly modify the mandatory flight leg trigger from here (unless there is no other option).

Dataref Editor





The Dataref Editor allows you to:

- Add dataref references to use in your scripts.
- You enter "raw" XML syntax into the editor (1) and test it using the [***Parse and Store***] button. Once it is parsed, it will store it in memory.

- The XML syntax for dataref element is:
  `<dataref name="{name}" key="{sim/...}"/>`

  You can find key list from plugins like "Dataref Tools" or "Laminar own Dataref page".

- You can define datarefs that are based on other plugins, but please make sure that you explain the simmer that they have to install the plugin first to be able to fly your mission.

Generating and Storing Conversion State

Once all is done, it is time to test your mission.
To do that, press the [*Generate*] button to create a mission in the: `"{xp}/Custom Scenery/missionx/random"` folder.

- I strongly advise to, at least, "check" the "Store State" checkbox, before generating the mission. This will keep a copy of the conversion data, you prepared, for future modifications. The plugin only saves the last conversion work that is currently in memory to a file.

- To test your work, open the Mission List screen from the home page and pick the "random" mission.

- At any point in time you can open the conversion screen and modify it, based on your tests, it will be kept in memory until you will press the [*Cancel*] button or exit X-Plane.
  This is very handy if you need to tweak "custom triggers" position based during test flight, or you need to modify Flight Leg settings based on what you observed during flight.

Summary

The conversion screen, in a way, can be your first tool to try and create your own adventure. The end result might seem basic at first but it should be good enough for most flights, you would like to share with the community.

To get better results, I encourage you to read the "*Designer Guide*" in order to better understand the parts of the mission file while later on, you could read the "*Designing Templates*" document. It explains how to convert one mission to a template, hence allowing you to provide mission customizations in one mission pack (see "Townsvile to Palm and Back" demo mission in x-plane.org site).

If you have any questions, you can email me directly "snagar.dev@protonmail.com"

Other materials can be found in the following links:

Tutorial - Converting LittleNavMap to mission file - part 1

Tutorial - Converting LittleNavMap to mission file - part 2

Until Then
Blue Skies
Saar.N