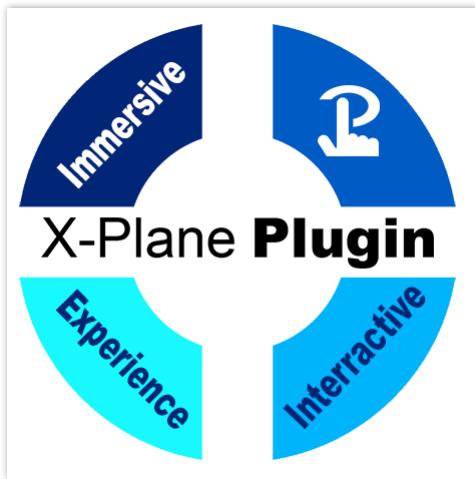


User Guide



1. Introduction

This is your user guide that will help you understanding the features of the X-Plane Plugin for Touch Portal.

In this guide, I will be using the Windows 10 operating system. Touch Portal is also compatible with Linux or Mac OS.

If the Touch Portal application has not been installed yet, please refer to the installation guide in the **README.MD** file.

You will find in the **Help** tab of the [Touch Portal website](#), Guides, FAQs, documentation on actions, and much more.

A [Discord site](#) is also available for Touch Portal to assist you.

It should be noted that many tutorials on Touch Portal are available in English, so it is essential to use the interface in this language. Go to the Touch Portal application. Go to Settings and Language to select English.

The illustrations in this guide will come from the English interface and an iPad.

The Touch Portal application can be used on a smartphone, **but using an iPad is preferable due to the size of its screen and ease of use.**

Touch Portal requires the use of a Windows application and a specific Wi-Fi device application. This can be a tablet or a smartphone.

User Guide

The purpose of this document is to link to the **default.json** file and some of buttons, on the **Cessna 172 G1000_1**, **Cessna 172 G1000_2** and **Cessna 172 G1000_3** pages. Note that I will be using the **Cessna 172 with the Garmin 1000**. This is an aircraft contained within X-Plane 12.

The use of the [**DatarefTool**](#) is essential to identify which of these Datarefs can be associated with a button on a Touch Portal page.

The three pages on the Cessna 172 provide a summary of all the options offered by the X-Plane plugin for Touch Portal.

N.B.: Decimal values (float) coming from X-Plane 12 Datarefs will be rounded to two decimals on the X-Plane plugin side (Touch Portal). I consider two decimals to be sufficient in terms of precision or display.

After finishing reading this document, you will be able to understand the mechanics of the X-Plane plugin for Touch Portal and create your own pages.

Examine this document carefully.

User Guide

2. Overview and Important Files

❖ The import file

To import essential elements for using the X-Plane plugin for Touch Portal, I use the **xplane_plugin_for_touch_portal.tpp** file, which is a dedicated format for importing plugins into Touch Portal. If you rename this file with the **Zip** extension, you will be able to see its contents.

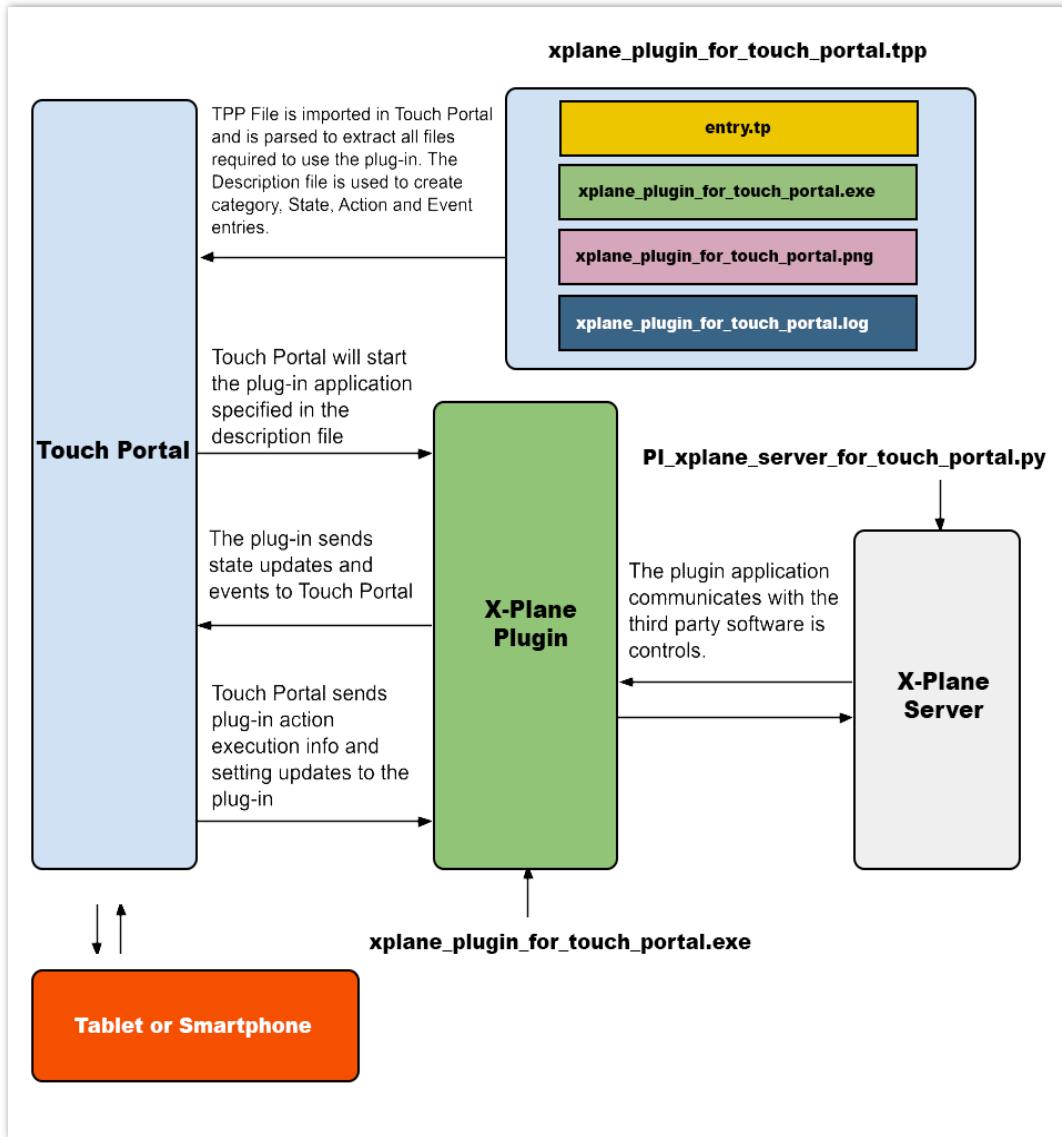


Figure 2.1: Import Format

User Guide

In this **TPP** file, there are 4 important elements:

1. entry.tp

This is a description file that serves as a framework or guide for Touch Portal regarding the X-Plane plugin. It is in JSON format and should be located at the root of the plugin file. **Never modify this file.**

2. xplane_plugin_for_touch_portal.exe

This is the X-Plane plugin for Touch Portal. It allows interaction with Touch Portal and the server (**PI_xplane_server_for_touch_portal.py**). This server is in the **PythonScripts** directory of **X-Plane 12**.

3. xplane_plugin_for_touch_portal.png

This is an image for the plugin menus in Touch Portal.

4. xplane_plugin_for_touch_portal.log

This is a log file that records what happens in the X-Plane plugin. You will find important messages and elements to understand errors.

User Guide

❖ The Importance of the default.json File

I will provide explanations on how to associate what you observe here with what can be read in the **default.json** file. It is somewhat a language to communicate with Touch Portal and X-Plane 12.

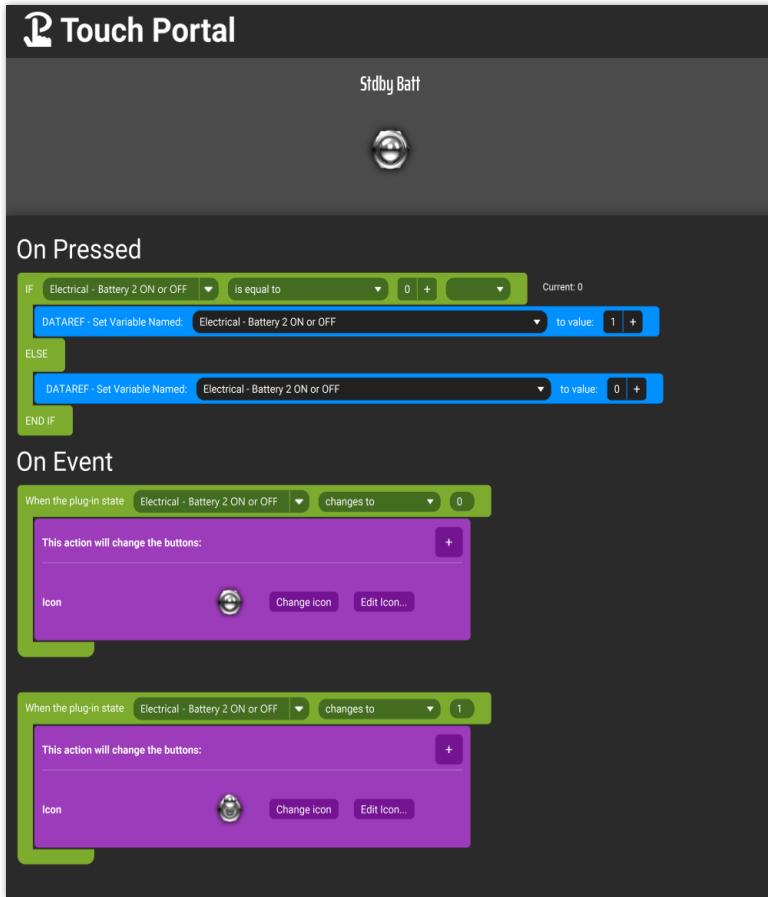


Figure 2.2 : Stdby Batt

➤ The Fields that are required by default.json File

It is important to emphasize the significance of the following fields in the **default.json** file. It is imperative that the following fields be assigned a value:

id, desc, group, dataref, and comment.

The **Electrical – Battery 2 ON or OFF** word has the identifier (**ID**):
xplane_plugin.sim/cockpit2/electrical/battery_on[1].

Search for this in the default.json file.

It is necessary that the **ID** in the default.json file be unique. Frequently, I will use the name of the **DATAREF** as the reference **ID** in this file with the prefix "**xplane_plugin.**" to ensure uniqueness.

User Guide

The word **Electrical** designates the **GROUP**.

Very important: If the **GROUP** field contains one of these values: **Slider**, **Command**, or **Command_3**, then the descriptions of the Dataref corresponding to these groups will be placed in dropdown lists for the **Slider**, **Command**, or **Command_3** usage respectively (These elements will be discussed in more detail later).

In the **DESC** field, we have the value **Battery 2 ON or OFF**.

The **COMMENT** field is useful for remembering the value domain of the Dataref related to this **ID**. For this Dataref, we have the values 0 and 1.

➤ The DATAREF field, and its very particular syntax

Here is how the **DATAREF** field is composed:

1. The name of the Dataref
2. Its **index** (if applicable) in brackets.

We will use a particular syntax to clearly indicate to the plugin that we have a Dataref with an index.

Thus, for our example, the Dataref **sim/cockpit2/electrical/battery_on** will have the syntax **sim/cockpit2/electrical/battery_on[1]**. The **index** is inside brackets.

It is essential to use this syntax in this field.

Note that the first index of a Dataref always starts at **0**.

➤ The optional fields values of the default.json file

It is important to emphasize the significance of the following fields in the **default.json** file. It is not necessary for these fields to be assigned a value:

touch_portal_format, **xplane_update_min_value**, **xplane_update_max_value**, and **accelerated_control**.

The **TOUCH_PORTAL_FORMAT** field might contain a format code (**chapter 10**).

The **XPLANE_UPDATE_MIN_VALUE** and **XPLANE_UPDATE_MAX_VALUE** fields might contain values corresponding to the limit of a particular Dataref's value. Often, we will find 0 and 1 as the minimum and maximum values, but there are others (**chapter 9**).

The **ACCELERATED_CONTROL** field might contain the value altitude or degree (**chapter 5**).

User Guide

➤ **A Tool to achieve perfect syntax**

Note that the github page for X-Plane plugin has a special HTML page (**See the link into the README.MD file**) that allows you to create a JSON file such as default.json, correctly.

This way, you can create a file for a specific aircraft, for example, the Toliss A320 (A320Toliss.json).

➤ **Where Can the default.json File Be Found?**

The default.json file provides a template for any page designer and concerns basic elements.

As you know, at the root of the X-Plane plugin in Touch Portal, there is a **misc** folder. Thus, the JSON files containing the Datarefs and their crucial information for the X-Plane plugin are under this directory.

TouchPortal/misc/xplane/default.json.

User Guide

3. An Overview of Buttons

❖ Editing Tabs Regarding a Button

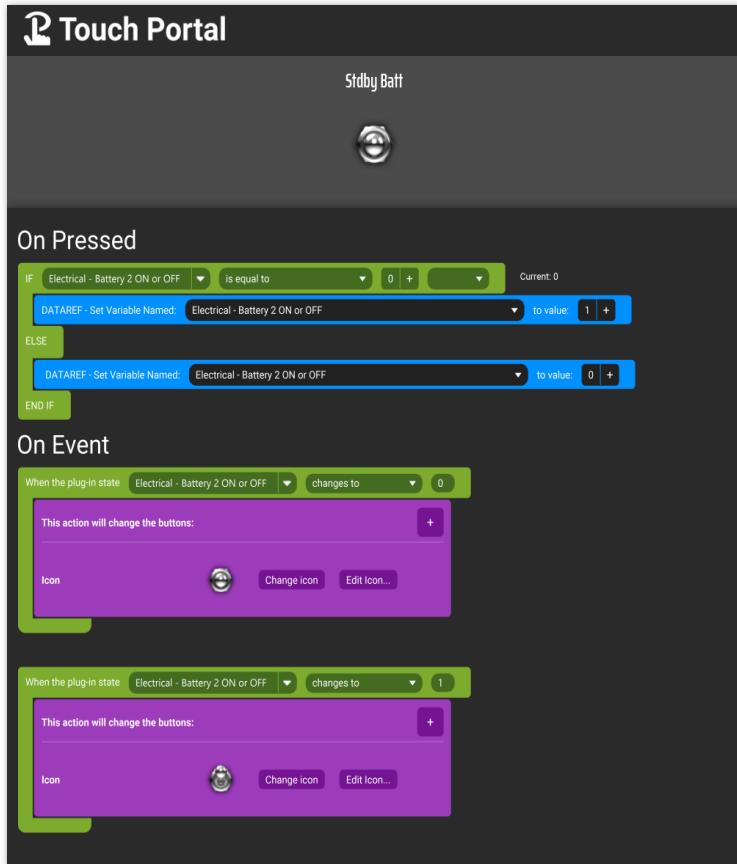


Figure 3.1 : Tabs and containers (STATE)

For a button to be functional, you need to use the **On Pressed** and/or **On Event**.

There is also an **On Hold** tab, which will be discussed later.

The **On Pressed** tab concerns what should be done when a button is pressed. This is referred to as the action to be performed here.

The **On Event** tab refers to what should be displayed based on the **container** values. We're talking about an event here. An event is triggered when a **container** value changes.

To the right of the IF condition, there is written **Current: 0**. This represents the current value of the **container**.

In Touch Portal terms, the **container** evaluated in this IF is a **STATE**. So we'll have a **STATE** on the Touch Portal side and a **Dataref** for X-Plane.

There are different containers (**STATE**) for this plugin. I'll explain the differences here.

User Guide

- Containers (STATE) for the personal processing of the plugin

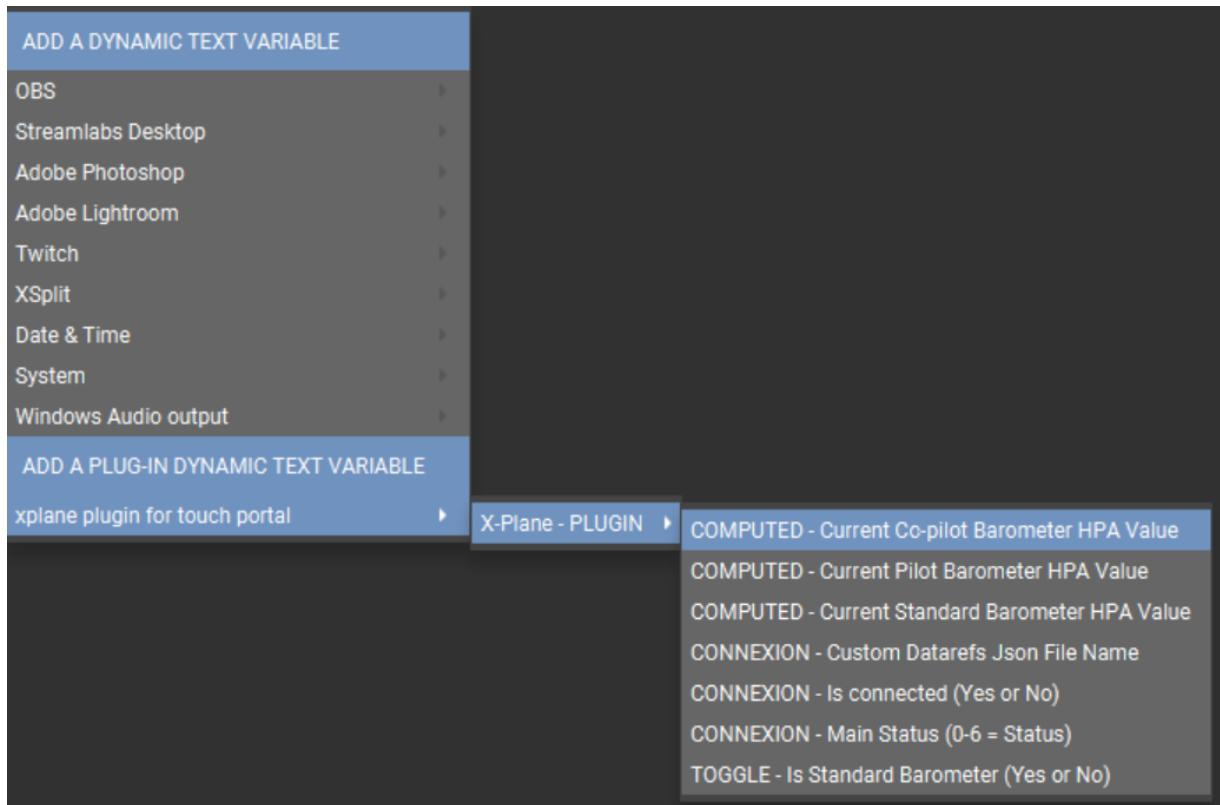


Figure 3.2: Container Menu (STATE) for the plugin

This plugin uses containers (**STATE**) for personal processing.

Thus, in Touch Portal menus, they will be categorized under:

xplane_plugin_for_touch_portal / X Plane-PLUGIN.

We have them for keeping calculated values (**COMPUTED**), for plug-in connection (**CONNECT**) and for processing switch buttons (**TOGGLE**).

User Guide

➤ Containers (STATE) for dataref and X-Plane controls

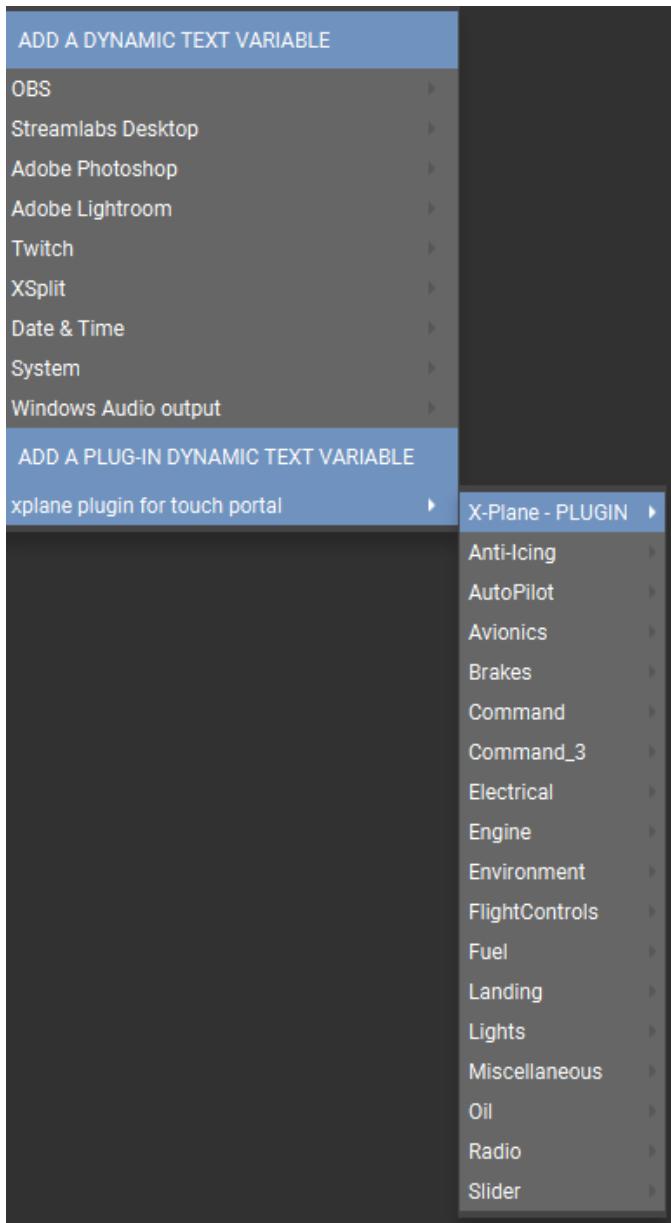


Figure 3.3: Container Menu (STATE) for Dataref and X-Plane Commands

This plugin uses containers (**STATE**) for Dataref and X-Plane commands. What we see here are the groups, and in these groups we have the containers (**STATE**) that keep the Dataref values.

All these containers (**STATE**) are dynamically generated and powered once at the first connection.

User Guide

➤ Containers (STATE) under the Command menu

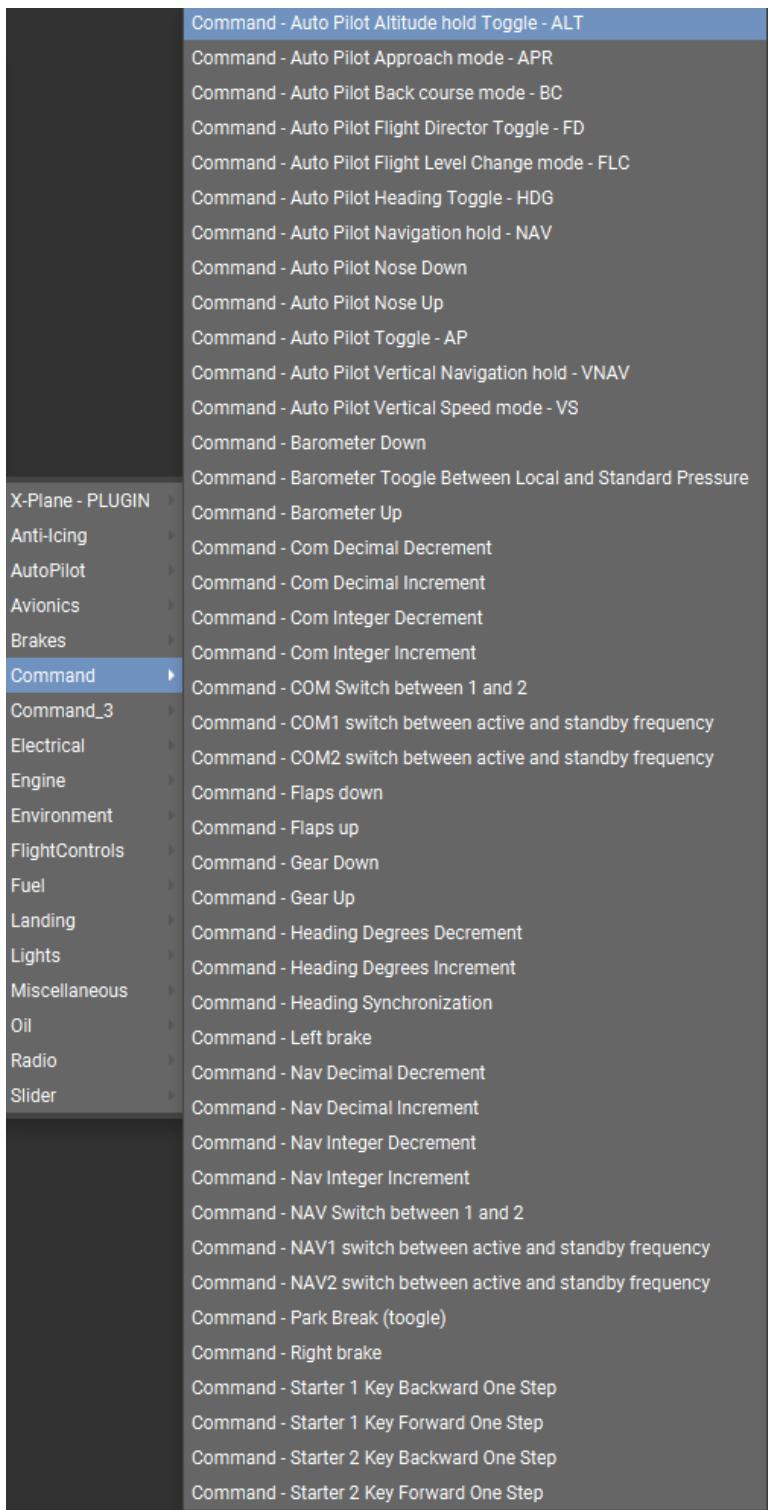


Figure 3.4: List of containers (STATE) under the Command menu

User Guide

➤ Containers (STATE) under the Command_3 menu

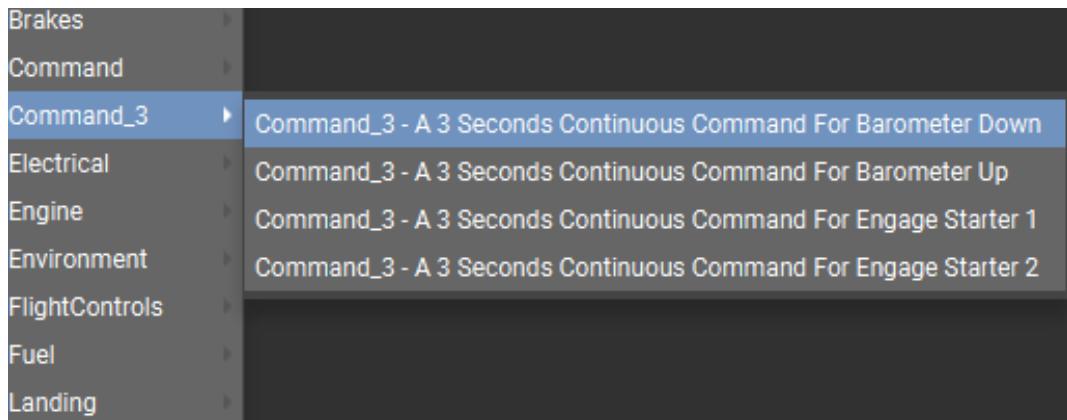


Figure 3.5: Container List (STATE) under the Command_3 menu

➤ Containers (STATE) under the menu Slider

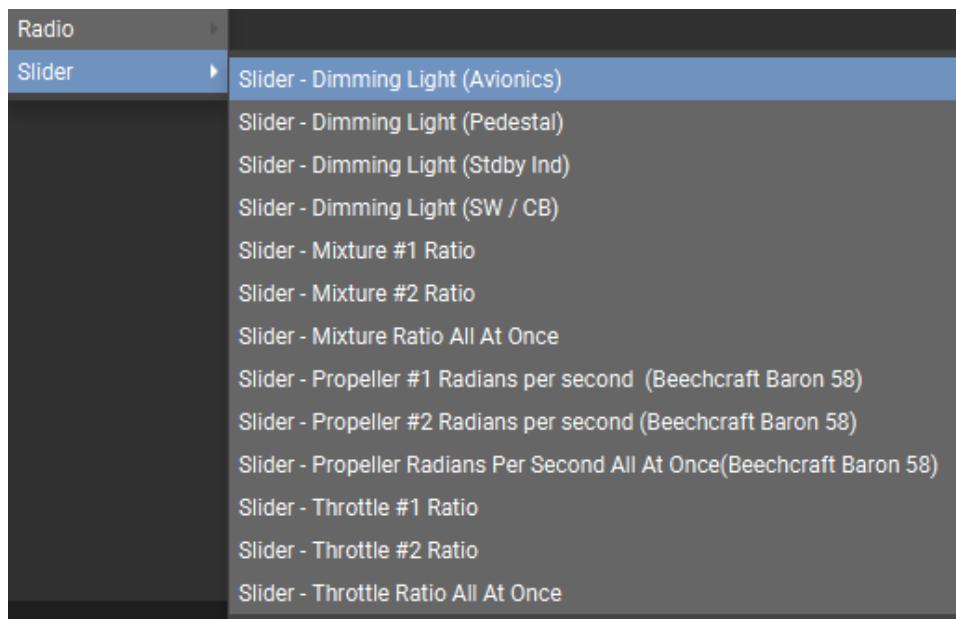


Figure 3.6: List of containers (STATE) under the Slider menu

User Guide

➤ Containers (STATE) under the Engine menu

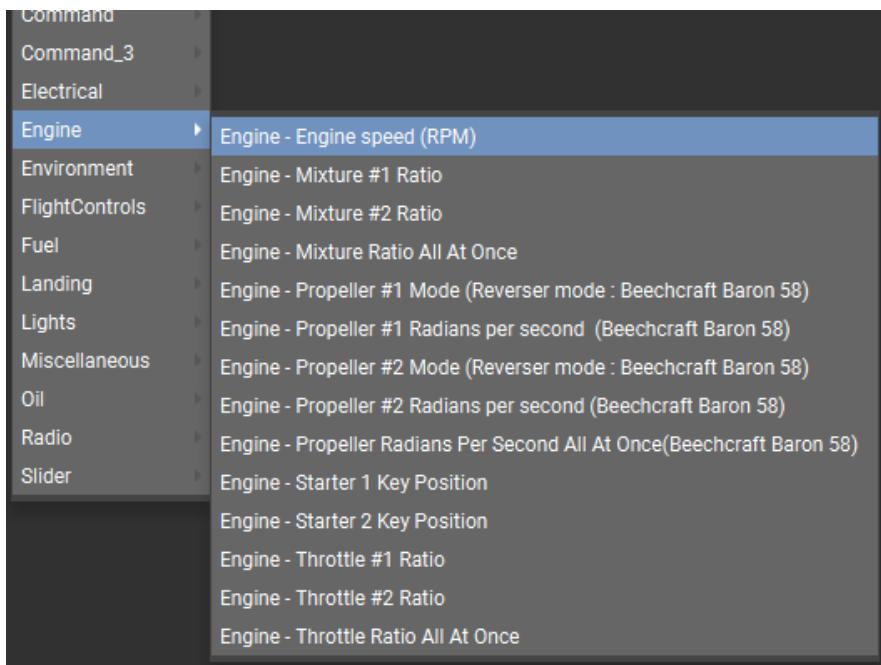


Figure 3.7: List of containers (STATE) under the Engine menu

As you can see, this plugin uses several (STATE) for Dataref and X-Plane commands. We see containers (STATE) that are grouped under **Engine**. It's up to you to find out about the others.

User Guide

❖ A Button Without the On Pressed Tab

Some buttons, such as the TEST button, do not utilize the On Pressed tab. Throughout this document, I will discuss buttons. Any box on a Touch Portal page can react when clicked. Some are there to inform only.

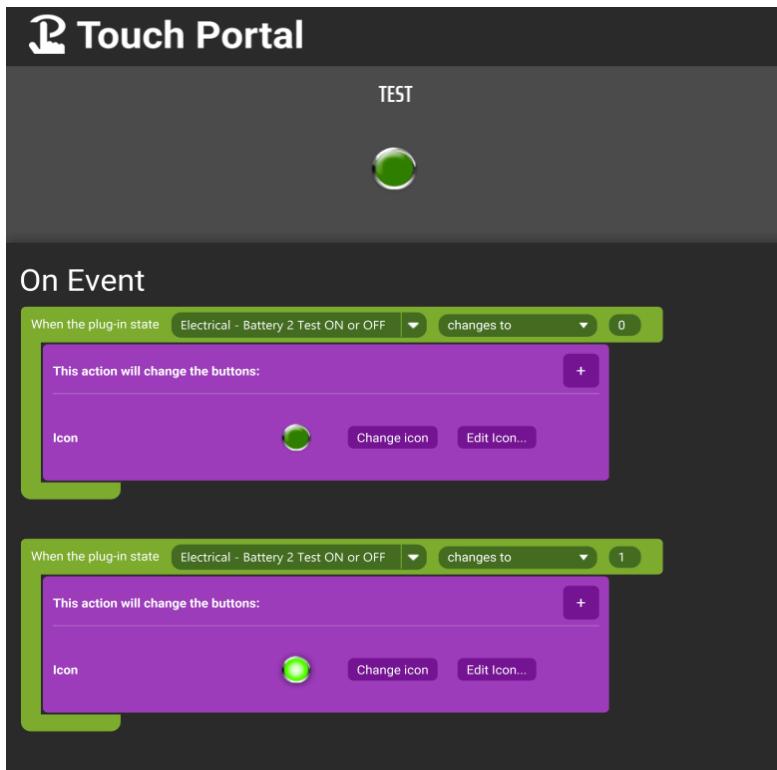


Figure 3.8 : TEST

This button is simply there to display a result based on the values of a Dataref.

The Dataref associated with this button is not array type. Therefore, you will not find any brackets at the end of the name of this Dataref. It's important to mention that.

Its reference **ID** is as follows: **xplane_plugin.sim/cockpit/warnings/annunciator_test_pressed**.

As you can see, there are conditions on the values 0 and 1.

Important: once again, know that on the Touch Portal side, the term **STATE** is frequently used. Here, we have (**When the plugin-in state**).

Our conditions concern **Electrical – Battery 2 Test ON or OFF** and its values 0 or 1.

I will explain later what concerns the content of the purple box labeled "This action will change...".

User Guide

❖ Other buttons are used as displays.

To display Airspeed, we use the On Event tab.

The green thing is a **Dynamic Text Updater**, and it allows displaying a Dataref based on its description. When you click on the "+" in this green box, you choose a description related to what you want to display.

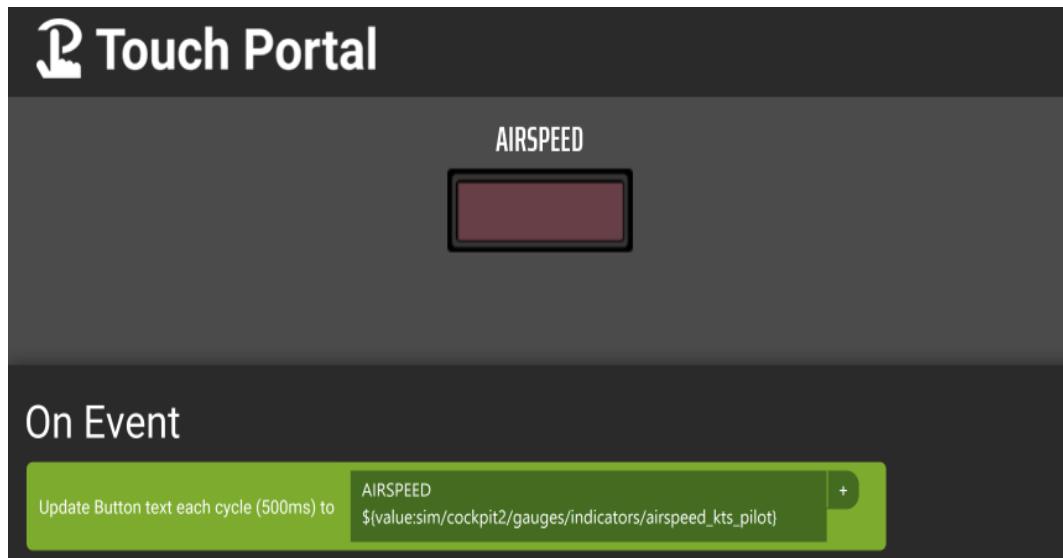


Figure 3.9 : AIRSPEED

I will elaborate on this **Dynamic Text Updater** later in this document.

The **ID** of Airspeed is: **xplane_plugin.sim/cockpit2/gauges/indicators/airspeed_kts_pilot**.

Avionics is the group and Airspeed in knots (pilot) is the description.

You can add a title to this display. This title is essential to know what we want to display. Thus, we have the **AIRSPEED** title preceding the value of the Dataref.

❖ Context Menu for Buttons in the X-Plane Plugin

By clicking on the "+", as I mentioned, and to find the description of Airspeed, you have to navigate through a context menu.

The root of this menu (for this plugin) is under **X-Plane plugin for touch portal**. Then, you choose its group, which is **Avionics**. Thus, you find the description **Avionics – Airspeeds in knots (pilot)**.

Note that groups are essential for grouping Datarefs.

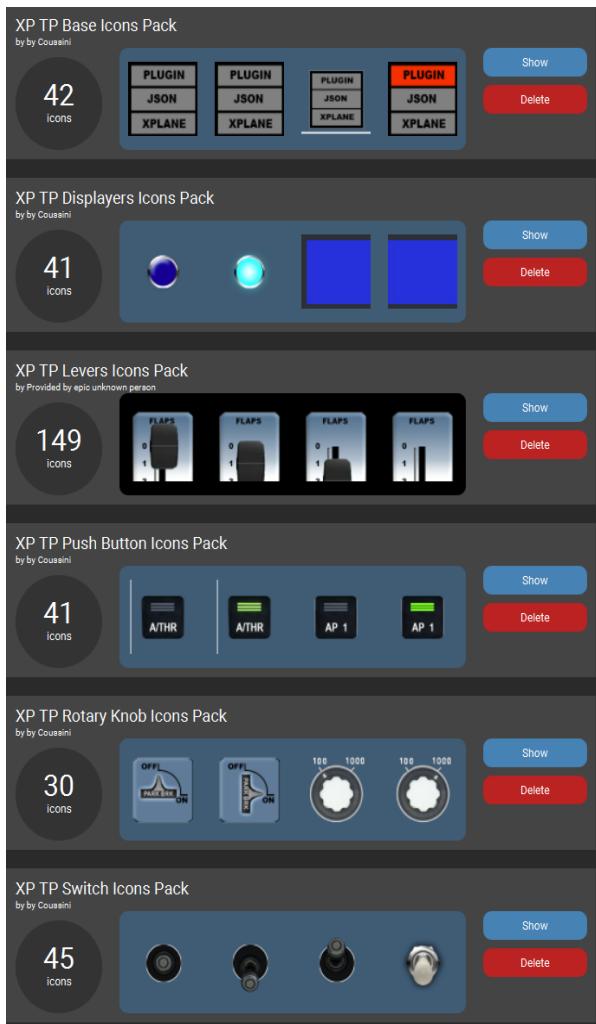
This is an efficient way to navigate through a menu that contains multiple elements.

User Guide

❖ Button Icons for the X-Plane Plugin

A picture is worth a thousand words. That's why a button uses an icon capable of symbolizing a state. It's possible to have multiple states for a single button.

The X-plane plugin for Touch Portal offers libraries for icons.



XP TP Base Icons Pack are the basic icons for the plugin.

XP TP Displayers Icons Pack are dedicated to displays.

XP TP Levers Icons Pack are dedicated to throttle levers... etc.

XP TP Push Button Icons Pack are dedicated to push buttons.

XP TP Rotary Knob Icons Pack are dedicated to rotary knobs.

Finally, **XP TP Switch Icons Pack** are dedicated to switch buttons.

Figure 3.10 : The different icons for the X-Plane plugin

User Guide

❖ Connection buttons

Special connection buttons are important (in fact, on the page **Cessna 172 G1000_1**, we will have three navigation buttons followed by the button representing a JSON file).

Starting from the left, we have:



Figure 3.11 : Navigation button

➤ Navigation button

Allows you returning to the "main" page of Touch Portal. However, you can use this same icon if you have a second page for an aircraft. This button should then allow you to go to the first page of your aircraft.

➤ Load JSON File and Start/Stop Plugin button

Allows you to start or stop the X-Plane plugin and load the JSON file. It is in this button that you must specify the file you want to load. In our case, it will be the default.json file.

When the JSON file is loaded, the third button will appear and represent that of X-Plane.

During the stopping process, it's important to wait until the status shows that it's done.

➤ Start Communication button

Will start communication between the plugin and X-Plane 12 server (`PI_xplane_server_for_touch_portal.py`). We will find this server in the **PythonScripts** directory under X-Plane 12.

➤ Status Indicator button

This is a status window that shows if there are any special errors or if everything is OK regarding the plugin, JSON file, or communication with the server.

➤ Loaded JSON File Name Display button

This is a display button that shows the JSON file name if the contains is valid. It will display the file name without the `.json` extension.

User Guide

❖ Using Connection buttons

The initial state of Touch Portal is the one explaining that the JSON file is not loaded and the plugin is not connected (**arrow 1**).

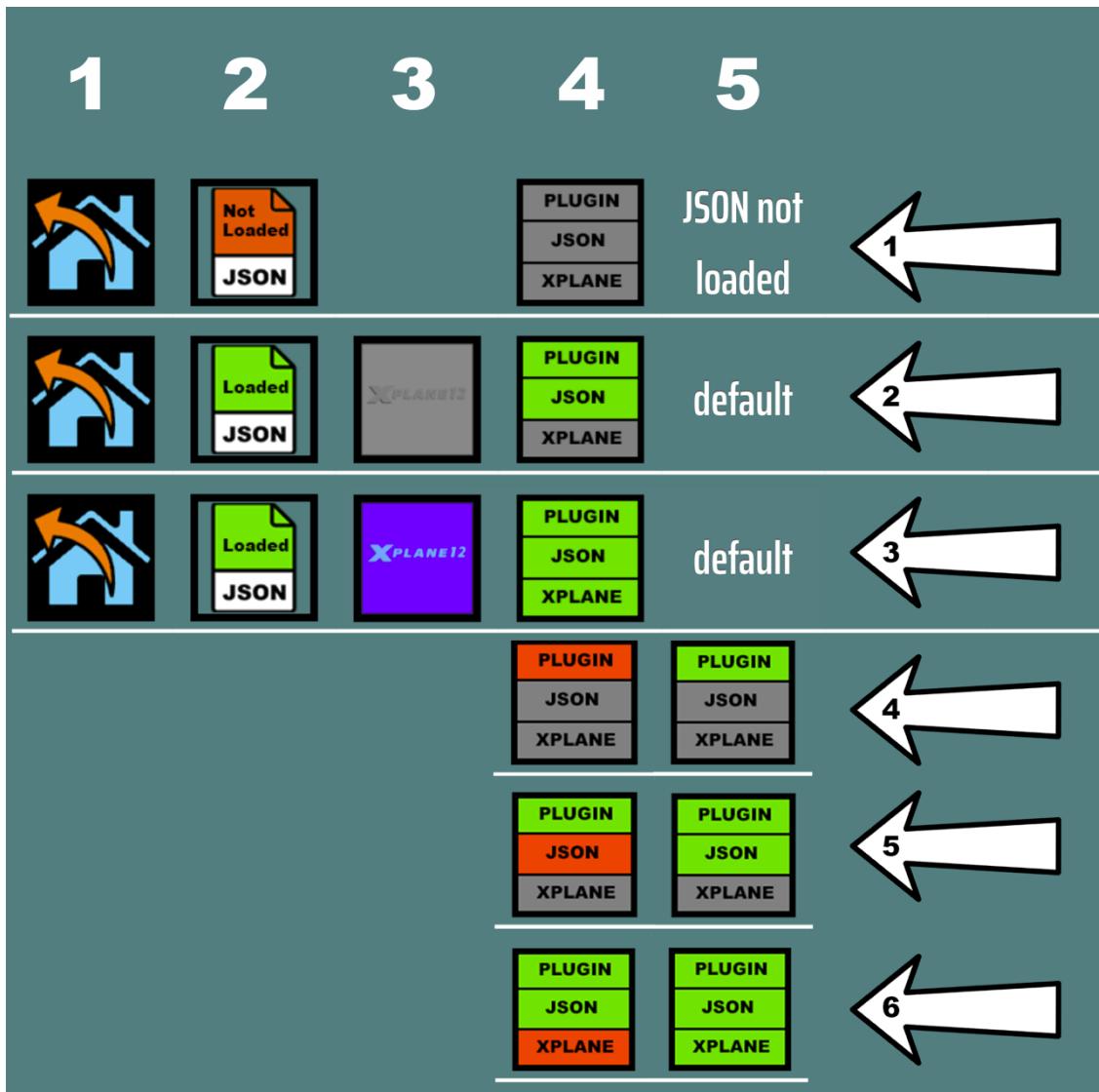


Figure 3.12 : Connection buttons states

By pressing the button representing a JSON file, the plugin will attempt to connect and load the JSON file.

If everything is OK, the **PLUGIN** and **JSON** indicators will turn green (**arrow 2**) and indicates that the plugin is operational. This state explain that the JSON file is valid and loaded. Consequently, the JSON file name will appear on button 5 (**JSON not loaded** will be replaced by **default**).

If the status button stays gray (**arrow 1**), please click the JSON button again until you get a good status (**arrow 2**).

User Guide

➤ State indicating that we can edit buttons

Very important: When the **PLUGIN** and **JSON** indicators are green, it means we can now use the data from the JSON file in our Touch Portal pages and buttons. Since the JSON file is our container for the Datarefs, this status indicates that the Datarefs from this file are available for editing in our Touch Portal pages.

Imagine you forgot to insert a Dataref in the JSON file. You should disconnect the plugin using the button representing a JSON file until all statuses are gray (**arrow 1**).

Then, add or edit your Datarefs to this file: **TouchPortal/misc/xplane/default.json**.

Be precise with your changes. However, the plugin can detect major errors in the JSON file during loading.

Once your modifications are complete in this file, restart the plugin to get the green indicators on **PLUGIN** and **JSON**.

➤ Why I don't see my Datarefs in the button editing menus?

Issue: Sometimes, the data from the JSON file are not visible or available in the editing menu. Attempting a **Reset** in Touch Portal or an **Exit followed by a start** can resolve this type of issue.

Sometimes, it's good to do a **Clear Data** in the Settings tab of Touch Portal.

➤ State indicating that we are communicating with X-Plane 12

Button 3 (**arrow 2**) explains that communication with X-Plane 12 is not currently running. Simply start a session of the X-Plane simulator, wait until the aircraft cockpit is loaded, then click this button.

Once the link between the X-Plane plugin and the X-Plane 12 simulator is established, the button representing X-Plane 12 will turn blue.

➤ States in all its colors

There are different statutes offered by this plugin.

If there is an error in the **PLUGIN**, it will be red, otherwise, it will be green (**arrow 4**).

If there is an error in the **JSON** file, it will be red, otherwise, it will be green (**arrow 5**).

If there is an error in the communication between the server and the plugin, the **XPLANE** label will be red, otherwise, it will be green (**arrow 6**).

User Guide

➤ Other buttons at the top of the first page

In the **Cessna 172 G1000_1 page**, the **AUTO PILOT** button will allow you to display the **CESSNA 172 G 1000_2** page concerning the necessary controls in the autopilot navigation. Note that the layout of some display items is similar to the Cessna 172 primary display. Note the blue area and the brown area representing the sky versus the ground. See **chapter 11** for more information.

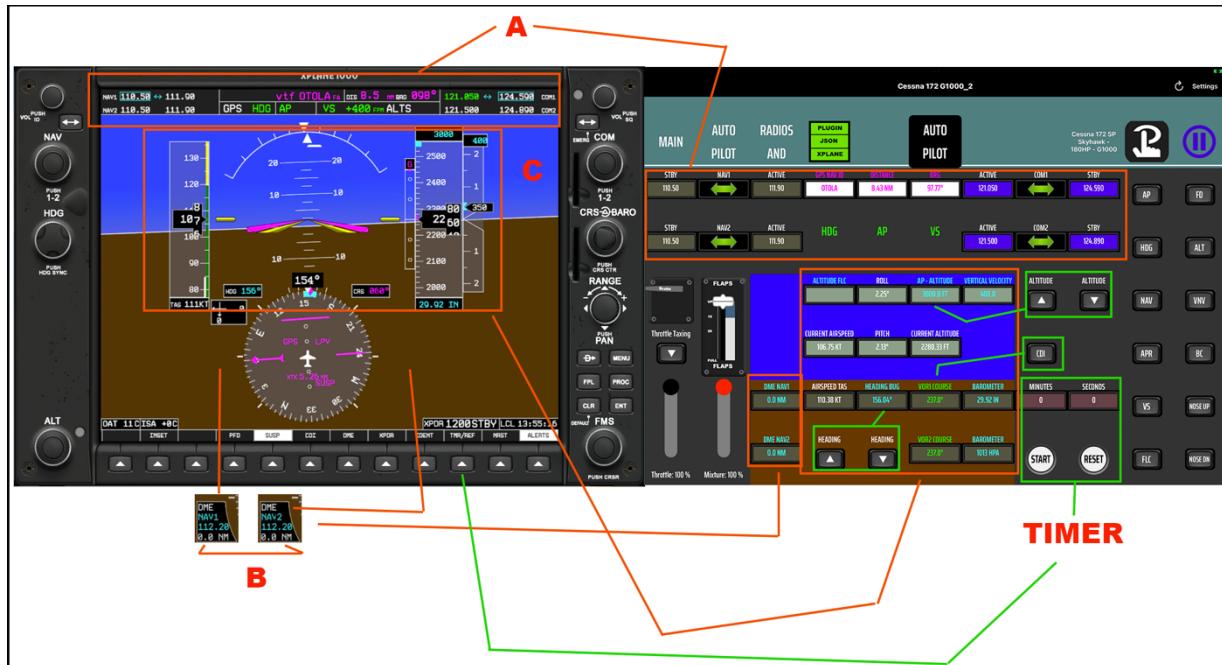


Figure 3.13: Autopilot page

The **RADIOS AND OTHERS** button will display the **Cessna 172 G1000_3** page to control the NAV and COM radios, the VOR1 and VOR2 courses, the autopilot altitude and the heading bug.

The **Touch Portal Icon** button will show the official website of Touch Portal.

The **Blue Pause button** allows you to switch between a pause or a resume simulation.

User Guide

❖ Editing Page Preview

When clicking on a page button, you can see the values of the Datarefs.

Main Status has the value 4 (**Current: 4**).

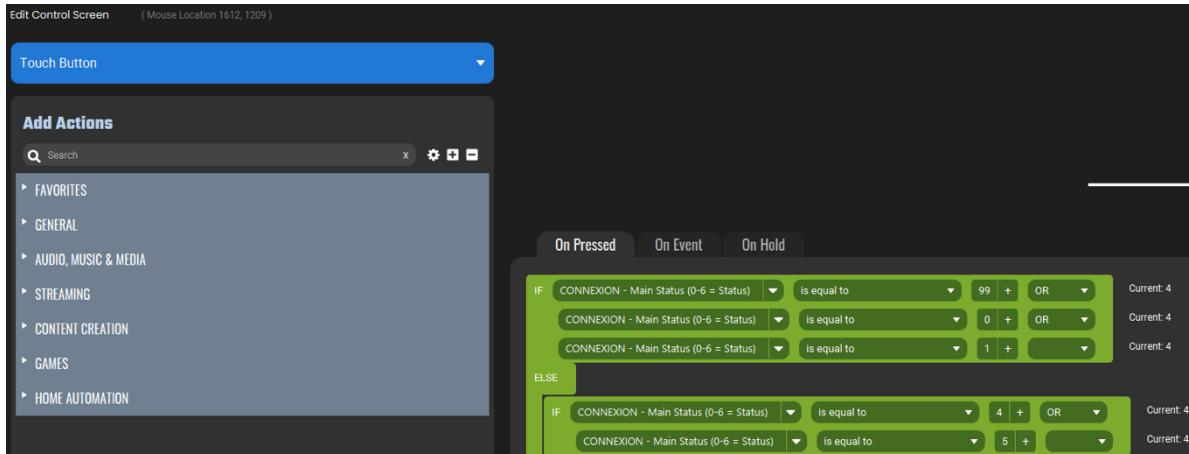


Figure 3.13 : Editing the X-Plane connection button

On the left, under **Add Actions**, we have different categories.

The control for the X-Plane plugin is under the **GAMES** category.

Important: Above the **Add Actions** label, there is a large bleu button, **Touch Button**. This allows you to choose between the **Touch Button** or **Slider** menu.

Explore the options under **Add Actions**, and you will see that you can filter categories, remove categories, or create favorites.

User Guide

4. Editing (Overview of categories in the X-Plane plugin)

❖ Overview of main categories under GAMES for X-Plane

Under the **GAMES** categorie, you will find the essential actions to build your X-Plane 12 aircraft pages. Further details will be provided in the following pages.

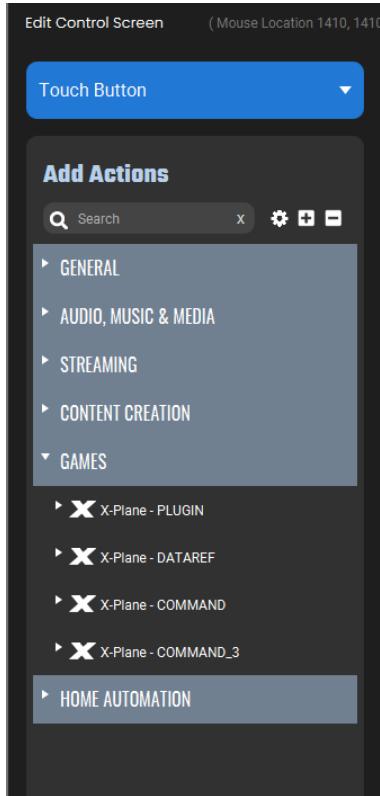


Figure 4.1 : Categories under GAMES for X-Plane 12

➤ X-Plane – PLUGIN

This is a category of essential actions for the X-Plane plugin and its communication.

➤ X-Plane – Dataref

This is a category of actions related to X-Plane Datarefs.

➤ X-Plane – COMMAND

This is a category of actions related to X-Plane commands.

➤ X-Plane – COMMAND_3

This is a category of actions related to X-Plane commands. However, it allows executing a command for 3 seconds (I will explain this in the following pages).

User Guide

❖ Overview of the X-Plane – PLUGIN category

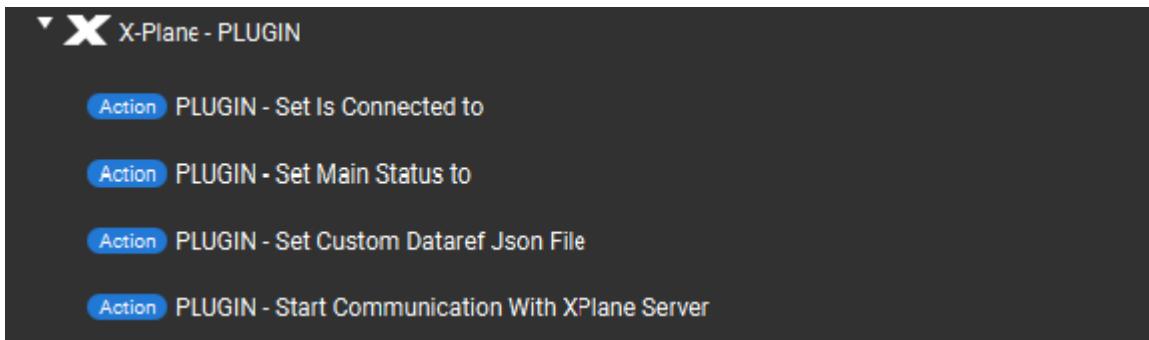


Figure 4.2 : X-Plane - PLUGIN category

➤ PLUGIN – Set Is Connected to

This control allows you to specify whether the plugin is connected or not.

➤ PLUGIN – Set Custom Dataref Json File

With this control, you can specify the name of the JSON file for your aircraft pages. This control is important on one of your pages. Just like in the button representing a JSON file on the **Cessna 172 G1000_1** page.

➤ PLUGIN – Start Communication With XPlane Server

This control is important on one of your pages and allows linking the X-Plane plugin with the X-Plane server, that runs when starting the X-Plane simulator.

Enables communication between the Plugin and X-Plane. See the blue or gray **XPLANE 12** button on the **Cessna 172 G1000_1** page.

User Guide

➤ **PLUGIN – Set Main Status to**

This control is important on one of your pages and allows specifying the status of the X-Plane plugin.

It can be found in a rectangle button (**PLUGIN, JSON and XPLANE**).

Typically, we set this data to 0, because the plugin monitors the other statuses.

The following statuses are available:

- 0: Initial status.
- 1: Error in the plugin.
- 2: Connected to Touch Portal.
- 3: Error regarding the JSON file.
- 4: The JSON file is loaded and valid.
- 5: Error regarding the X-Plane server.
- 6: The plugin is connected to the X-Plane server.

User Guide

❖ Preview of the X-Plane - Dataref category

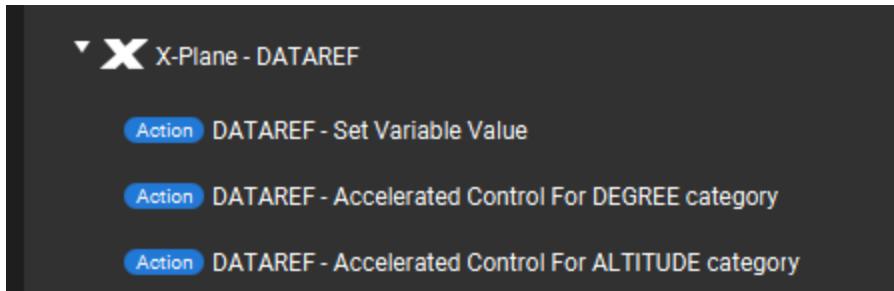


Figure 4.3 : X-Plane - Dataref category overview

➤ Dataref – Set Variable Value

This control allows specifying a value and assigning it to a Dataref. It is widely used in the **Cessna 172 G1000_1** page, like in the **AVIONICS** button.

➤ Dataref – Accelerated Control For DEGREE category

Similar to a digital clock adjustment, this control enables holding a button for adjustment. The longer you hold the button, the faster the values increase.

It provides fine adjustment initially and, with continued button press, accelerates the adjustment.

When reaching 0 or 360 degrees, the value wraps around to 0.01 or vice versa to 359.99.

Used in the **Cessna 172 G1000_2** and **Cessna 172 G1000_3** pages, for instance, in the **HEADING BUG**.

➤ Dataref – Accelerated Control For ALTITUDE category

Similar to the previous control, but it is for altitudes. For example, **AP-ALTITUDE** into **Cessna 172 G1000_2** and **Cessna 172 G1000_3** pages.

User Guide

❖ X-Plane - COMMAND category overview

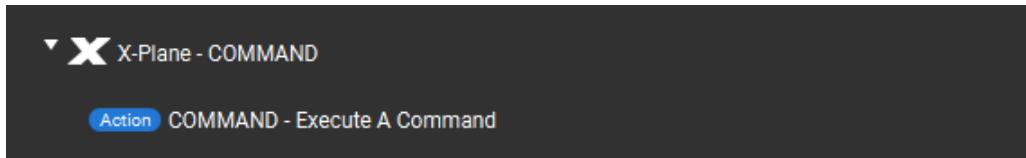


Figure 4.4 : X-Plane - COMMAND category overview

➤ COMMAND – Execute A Command

This control initiates a command. Further details will be provided in subsequent pages. When a button utilizes this control, pressing the button triggers the specified command in this control. It is widely used on the **Cessna 172 G1000_2** page, such as on the **VNV** button.

❖ X-Plane - COMMAND_3 category overview

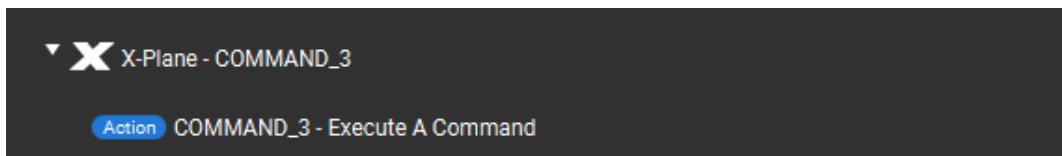


Figure 4.5 : X-Plane - COMMAND_3 category overview

➤ COMMAND_3 – Execute A command

When a button utilizes this control, pressing the button triggers the specified command in this control for approximately 3 seconds.

It is used on the **Cessna 172 G1000_1** page for adjusting the barometers. Look for the mauve area on this page. Check under the label: **Click Once 3-Seconds Execution**.

User Guide

5. Editing (Using categories of the X-Plane Plugin)

❖ Utilizing the X-Plane - PLUGIN category

➤ PLUGIN – Set Is Connected to

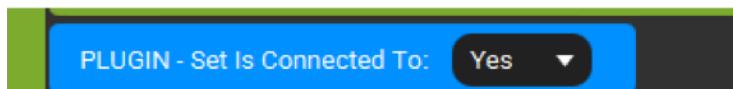


Figure 5.1: Details of the item PLUGIN – Set Is Connected to

- The previous control, allow to specify, whether or not, that the plugin is connected. The values are **Yes** or **No**.

🔍 See this action in the button representing a JSON file in the **Cessna 172 G1000_1** page.

➤ PLUGIN – Set Custom Dataref Json File



Figure 5.2 : Details of the PLUGIN – Set Custom Dataref Json File category

- There must be a value in this field. The JSON filename must exactly match the one in the directory: **TouchPortal/misc/xplane/default.json**.

If the JSON filename contains uppercase letters, then it must be written in the same way.

🔍 See this action in the button representing a JSON file on the **Cessna 172 G1000_1** page.

➤ PLUGIN – Start Communication With XPlane Server

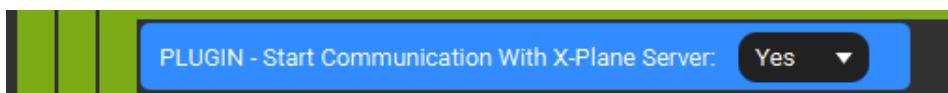


Figure 5.3 : Details of the PLUGIN – Start Communication With XPlane Server category

- Possible values for this control are **YES** or **NO**.

🔍 See this action in the X-Plane button (gray or blue).

This button may not be visible on the **Cessna 172 G1000_1** page. However, know that it is immediately to the right of the button representing a JSON file .

User Guide

➤ PLUGIN – Set Main Status to

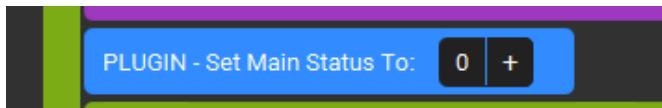


Figure 5.4 : Details of the PLUGIN – Set Main Status to category

- This allows resetting the status to zero (initial status).

See this action in the button representing a JSON file on the **Cessna 172 G1000_1** page.

❖ Utilization of the X-Plane – Dataref category

➤ DATAREF – Set Variable Value

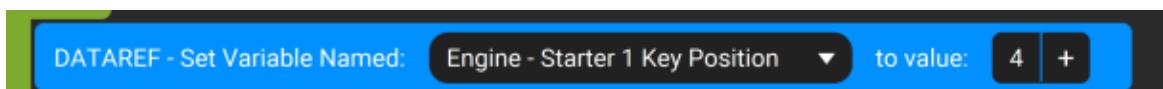


Figure 5.5 : DATAREF – Set Variable Value

"The most useful category in your pages."

- The first input field corresponds to the Dataref description. We will find this data through a contextual menu.
- The second input field is the value we want to assign to this Dataref.

See this action in the right button representing the starter of the Cessna 172 on page **Cessna 172 G1000_1**.

User Guide

➤ DATAREF – Set Accelerated Control For DEGREE category

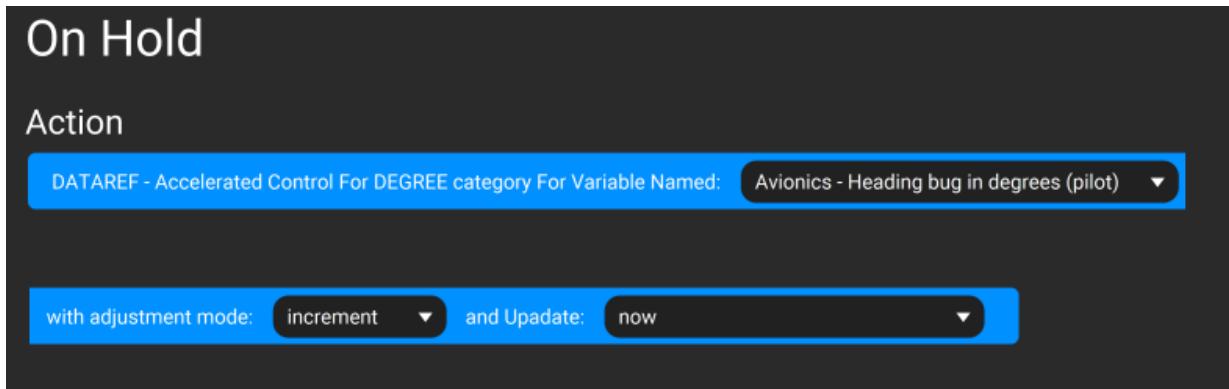


Figure 5.8 : DATAREF – Set Accelerated Control For DEGREE category

The input box for this action is so long that I present it in two segments.

- The first input field corresponds to the description of the Dataref.
- The second input field offers the choice between **increment** or **decrement**.
- The third input field offers the choice "**now**" or "**as soon as you release the button**". The choice "**now**" means that the values will be immediately written to the X-Plane 12 Dataref. The other choice is special, as Touch Portal has internal **STATE** values that allow displaying. By using this update type, holding the button will display the values and prevent multiple updates to X-Plane 12. When you release the button, the last value is sent to the server to update the Dataref.

Note that this action must be placed in the **On Hold** tab because it is linked to holding down a button. For this function, you must hold down the corresponding button.

Also note that in the default.json file, we have **degree** in the **accelerated_control** field according to the description **Avionics - Heading bug in degrees (pilot)**.

The **ID** for this description is: " **xplane_plugin.sim/cockpit/autopilot/heading_mag**".

This **accelerated_control** field allows grouping and is used in the DEGREE category list for the dropdown box. There are several Datarefs using degree value. It's up to you to decide which is best for this particular action.

 See this action in the triangle-shaped button with an **HEADING** label, in **Cessna 172 G1000_2** and **Cessna 172 G1000_3** pages.

User Guide

➤ DATAREF – Set Accelerated Control For ALTITUDE category

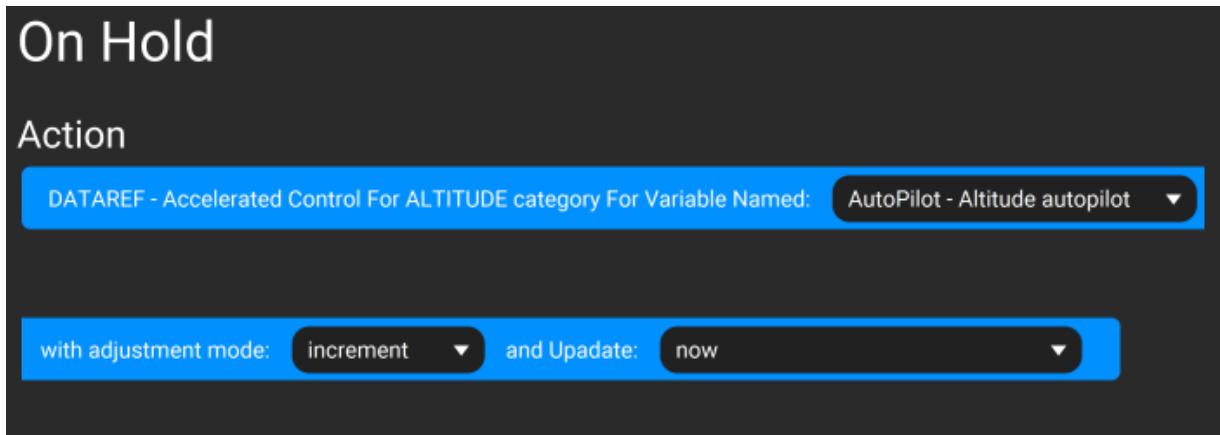


Figure 5.9 : DATAREF – Set Accelerated Control For ALTITUDE category

The input box for this action is so long that I present it in two segments.

- The first input field corresponds to the description of the Dataref.
- The second input field offers the choice between **increment** or **decrement**.
- The third input field offers the choice "**now**" or "**as soon as you release the button**". The choice "**now**" means that the values will be immediately written to the X-Plane 12 Dataref. The other choice is special, as Touch Portal has internal **STATE** values that allow displaying. By using this update type, holding the button will display the values and prevent multiple updates to X-Plane 12. When you release the button, the last value is sent to the server to update the Dataref.

Note that this action must be placed in the **On Hold** tab because it is linked to holding down a button. For this function, you must hold down the corresponding button.

Also note that in the default.json file, we have **altitude** in the **accelerated_control** field according to the description **Autopilot - Altitude autopilot**.

The **ID** for this description is: "**xplane_plugin.sim/cockpit/autopilot/altitude**".

This **accelerated_control** field allows the control to categorize Datarefs of the same type and limit the values in the dropdown box to this type. There are indeed Datarefs with an altitude value. It's up to you to choose the one that suits you for this action.

See this action in the triangle-shaped button labeled **ALTITUDE** in **Cessna 172 G1000_2** and **Cessna 172 G1000_3** pages.

User Guide

❖ Utilisation de la rubrique X-Plane - COMMAND

➤ COMMAND – Execute A Command

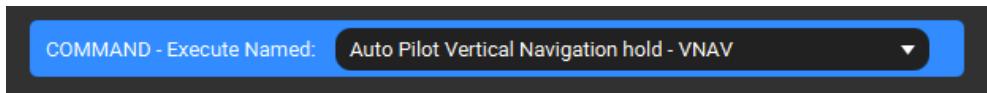


Figure 5.10 : COMMAND – Execute A Command

- The input field corresponds to the description of the X-Plane command.

See this action in the button labeled VNV in the black area of page **Cessna 172 G1000_2**.

➤ The Dataref field for an X-Plane command and its very specific syntax

Here's how the **DATAREF** field in the default.json file is composed when we want to use it for an X-Plane 12 command.

- The name of the Command
- The **CMD** label enclosed in brackets.

We use a specific syntax to inform to the plugin that we have a command.

The **ID** of this command is: " **xplane_plugin.sim/autopilot/vnav[CMD]**"

Thus, regarding our example, the Dataref **sim/autopilot/vnav[CMD]** will be the X-Plane 12 command **sim/autopilot/vnav**.

Note that we also use brackets for array indices in this file. We discussed this earlier in this document. This time, we use the word **CMD** to indicate that it is a command.

It is essential to use this syntax in this field regarding a command.

User Guide

❖ Using the X-Plane - COMMAND_3 category

➤ COMMAND_3 – Execute A Command

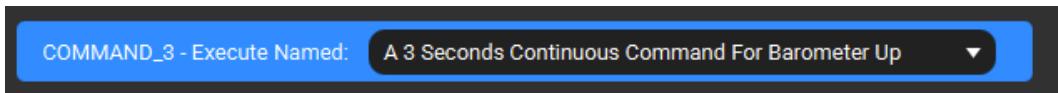


Figure 5.11 : COMMAND_3 – Execute A Command

- The input field corresponding to the description of the X-Plane command.

🔍 See this action in the button labeled **BARO ACCEL** in the mauve area of the **Cessna 172 G1000_1** page.

🔍 Also, see this action in the right button of the starter on the **Cessna 172 G1000_1** page. This allows simulating a 3-second starter operation.

➤ The Dataref field for an X-Plane command and its very particular syntax

Here's how the **DATAREF** field in the default.json file is composed when you want to use it for an X-Plane 12 command.

- The name of the Command
- The **CMD_3** label in square brackets.

We use a specific syntax to inform the plugin that we have a command that will be executed for 3 seconds.

The **ID** of this command is: "**xplane_plugin.sim/GPS/g1000n1_baro_up[CMD_3]**"

So, in our example, the Dataref **sim/GPS/g1000n1_baro_up[CMD_3]** will be the X-Plane 12 command **sim/autopilot/vnav**.

Note that we also use square brackets for array indices in this file. We discussed this earlier in this document. This time, we use the term **CMD_3** to indicate that it's a command.

It is essential to use this syntax in this field for a command that executes for 3 seconds.

User Guide

- **Another Dataref uses the same command as explained above!**

Note that you will also find a **sim/GPS/g1000n1_baro_up** command under the following **ID**:

xplane_plugin.sim/GPS/g1000n1_baro_up[CMD]

This will allow for fine adjustment for barometer. This action is similar to the one in **Figure 5.11**. However, you should not hold down this button; you should press once only. Additionally, this command will not be in the **On Hold** tab but in the **On Pressed** tab.

 See this action in the button labeled **BARO** in the red area of the **Cessna 172 G1000_1** page.

User Guide

6. Edition (categories under Slider menu)

❖ Overview of the main titles under GAMES for X-Plane

Important: Above the **Add Actions** label, there is a large bleu button, **Touch Button**. This allows you to choose between the menu related to **Touch Button** or **Slider**.

Under the GAMES section, you will see the essential actions for the **slider** and to build your airplane pages under X-Plane 12.

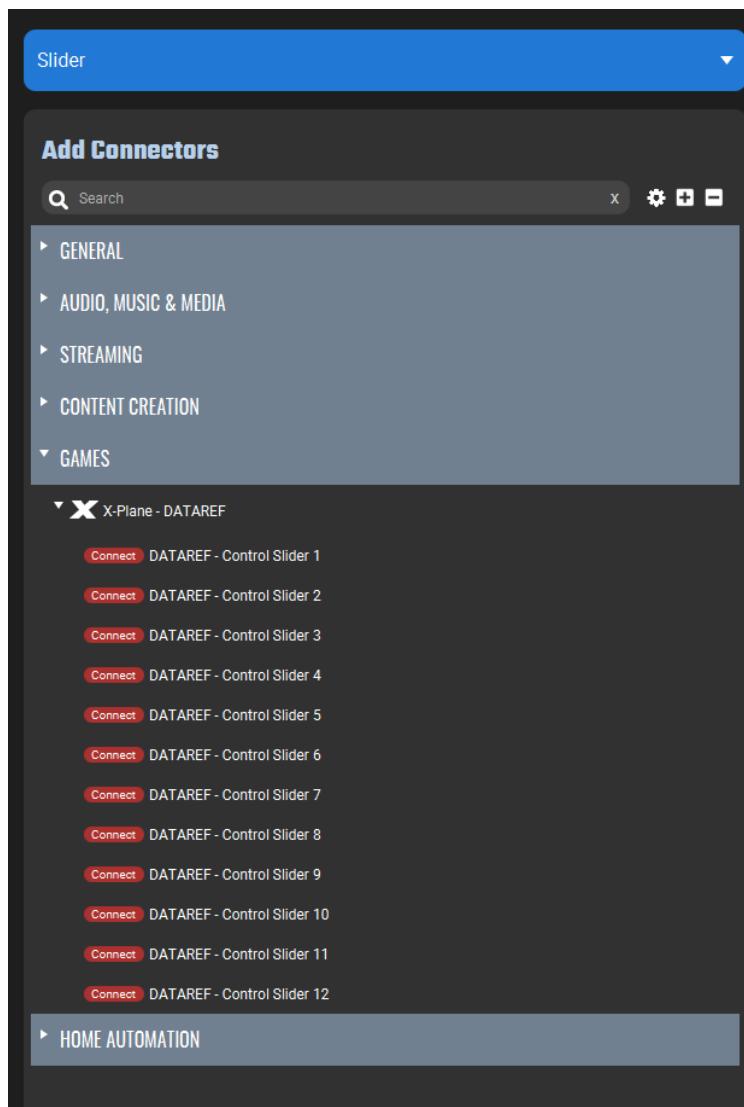


Figure 6.1: Slider menu and containers (STATE) for sliders

User Guide

❖ A Very Special Button I Call Slider

Here we have **Slider** or **Connector** type controls.

These controls are simple but require a certain rigor.

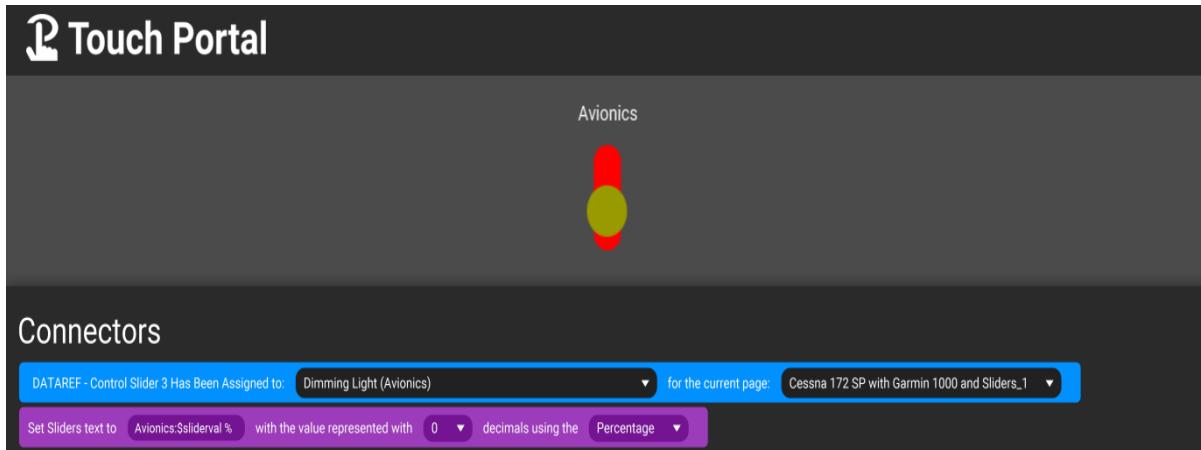


Figure 6.2: Avionics

The X-Plane plugin for Touch Portal has **12 Slider controls (Figure 6.1)**.

Important: Each number must be used only once per page or per aircraft.

```
=====
Slider list by page, slider number and description
=====
Page = ??? ; Slider num = 2 ; Description = Mixture Ratio All At Once
Page = ??? ; Slider num = 2 ; Description = Throttle Ratio All At Once
Page = Cessna 172 SP with Garmin 1000 and Sliders_1 ; Slider num = 1 ; Description = Throttle #1 Ratio
Page = Cessna 172 SP with Garmin 1000 and Sliders_1 ; Slider num = 2 ; Description = Mixture #1 Ratio
Page = Cessna 172 SP with Garmin 1000 and Sliders_1 ; Slider num = 3 ; Description = Dimming Light (Avionics)
Page = Cessna 172 SP with Garmin 1000 and Sliders_1 ; Slider num = 4 ; Description = Dimming Light (SW / CB)
Page = Cessna 172 SP with Garmin 1000 and Sliders_1 ; Slider num = 5 ; Description = Dimming Light (Pedestal)
Page = Cessna 172 SP with Garmin 1000 and Sliders_1 ; Slider num = 6 ; Description = Dimming Light (Stdby Ind)
Page = Cessna 172 SP with Garmin 1000_1 ; Slider num = 1 ; Description = Throttle #1 Ratio
Page = Cessna 172 SP with Garmin 1000_1 ; Slider num = 2 ; Description = Mixture #1 Ratio
Page = Cessna 172 SP with Garmin 1000_1 ; Slider num = 3 ; Description = Dimming Light (Avionics)
Page = Cessna 172 SP with Garmin 1000_1 ; Slider num = 4 ; Description = Dimming Light (SW / CB)
Page = Cessna 172 SP with Garmin 1000_1 ; Slider num = 5 ; Description = Dimming Light (Pedestal)
Page = Cessna 172 SP with Garmin 1000_1 ; Slider num = 6 ; Description = Dimming Light (Stdby Ind)
Page = Cessna_barron_Show case ; Slider num = 1 ; Description = Throttle Ratio All At Once
```

Figure 6.3: Logs Regarding the List of Sliders

To have a complete list of Sliders, use the **Logs tab** of the Touch Portal application.

User Guide

Check the first two lines (**see Page = ??? in Figure 6.3**). This indicates that these Sliders are not properly configured.

Therefore, you must go through all pages using sliders, then check the "**for the current page**" input field to see if it is correctly filled out (to resolve this issue).

The **Logs tab** in Touch Portal is important. It is used a lot and ensure that everything is OK.

Note: If you plan to create more than one page per aircraft, the main page should always have the special connection buttons.

The principle of the plugin is to connect on the first page, and each page using Sliders must use the same page value (name of the main page).

If the main page is named **Cessna 172 G1000_1** and a second page is named **Cessna 172 G1000_2**, the Sliders on the second page must be linked to page 1 with the name **Cessna 172 G1000_1**.

Thus, if an aircraft has 4 pages, there will only be possible to create 12 Sliders for these pages.

It is important that an airplane only use 12 sliders.

Therefore, the **Logs tab** can inform us about this.

In summary, these buttons are simple but require a certain rigor.

User Guide

7. Edition (Touch Portal Topics Overview)

Here, I explain only the actions used in my pages. You will see these actions listed in my favorites.

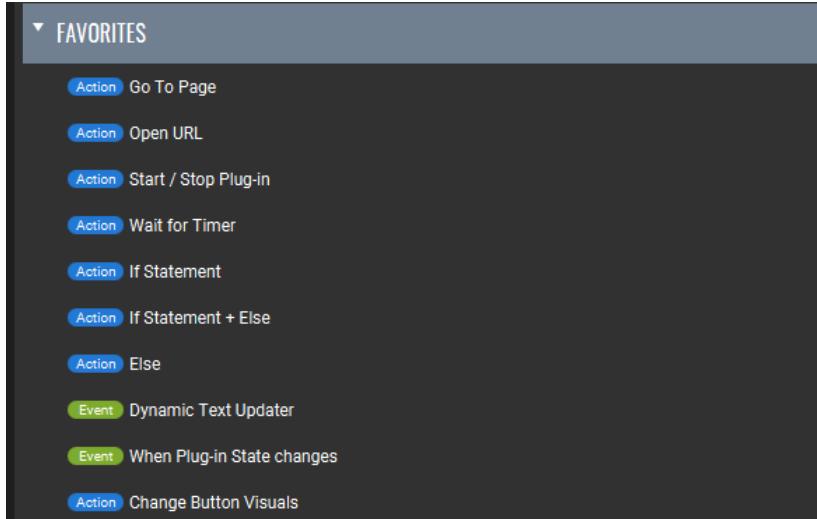


Figure 7.1 : Favorites for actions in the Touch Portal menu

❖ Overview of the main categories under GENERAL for Touch Portal

Under the **GENERAL** category, you'll find other essential actions for building your X-Plane 12 aircraft pages. Further details will be provided in the following pages.

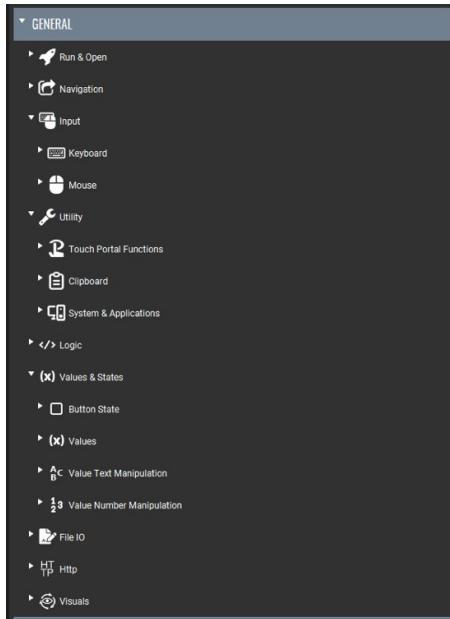


Figure 7.2 : GENERAL category for X-Plane 12

User Guide

❖ The Navigation category

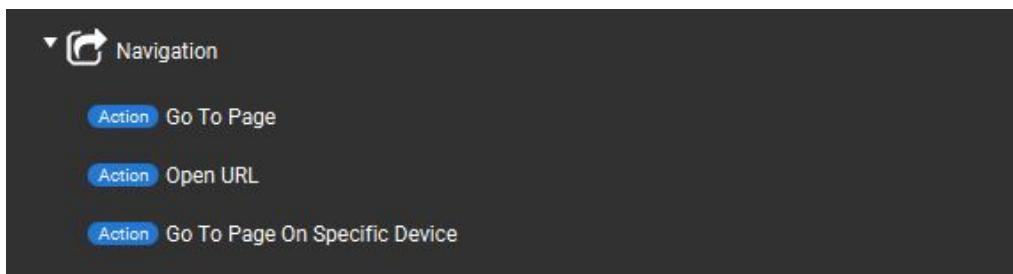


Figure 7.3 : The Navigation category

➤ Go to Page

It's an action that allows you to open a Touch Portal page.

➤ Open URL

It's an action that allows you to open an internet page.

❖ The Utility category

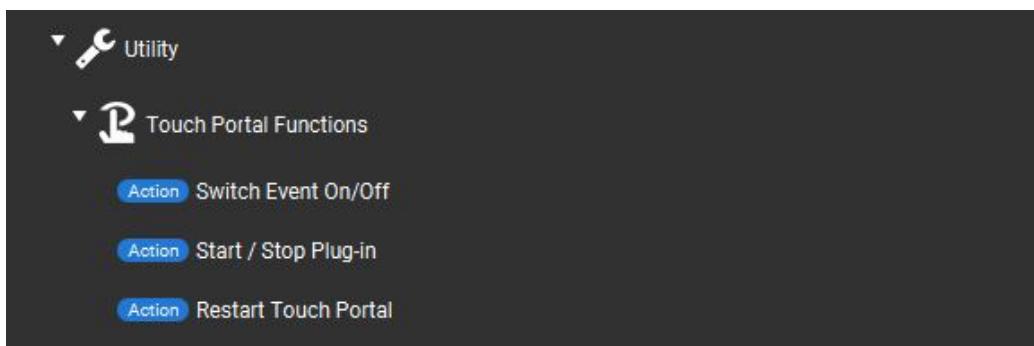


Figure 7.4 : The Utility Category

➤ Start / Stop Plug-In

It's an action to start or stop a plugin.

User Guide

❖ The Logic category

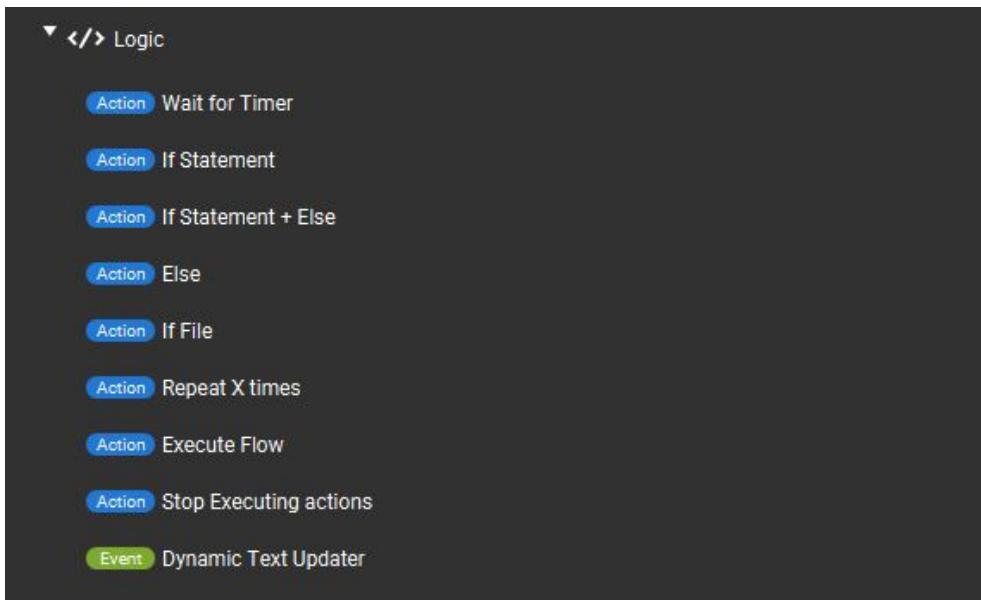


Figure 7.5 : The Logic category

➤ **Wait for Timer**

It's an action to perform a wait.

➤ **If Statement**

It's an action to add a condition to a button.

➤ **If Statement + Else**

It's an action to add a complete condition to a button. If the condition is not met, the Else part will handle the rest.

➤ **Else**

It's an action to add an **Else** to an existing condition.

➤ **Dynamic Text Updater**

It's an action to automatically update the button by displaying a value every half-second. Very useful for displays.

User Guide

❖ The Values & States category

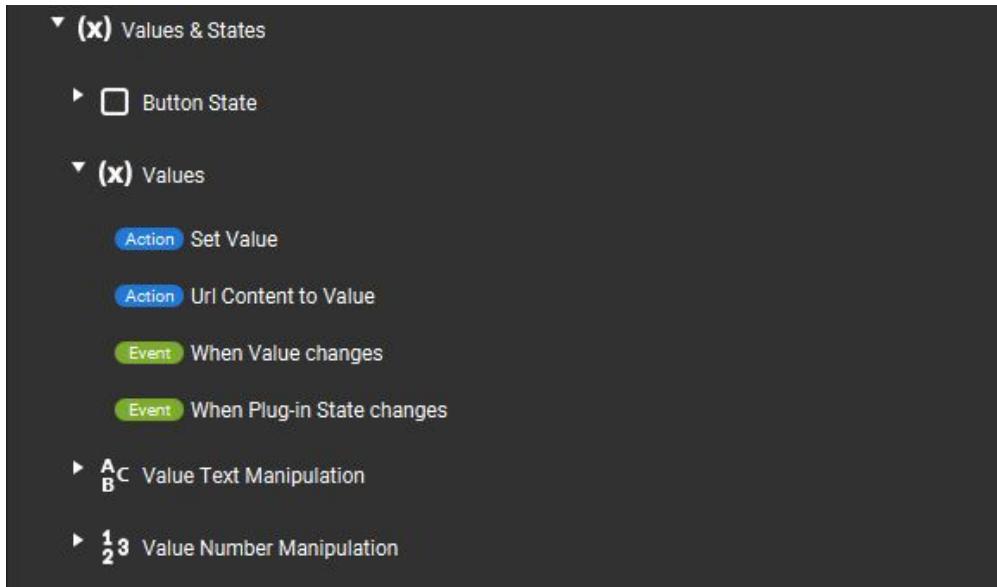


Figure 7.6 : The Values & States category / Values

➤ When Plug-in States changes

This is an action that allows you to add a condition to a button in the **On Event** tab.

❖ The Visuals category

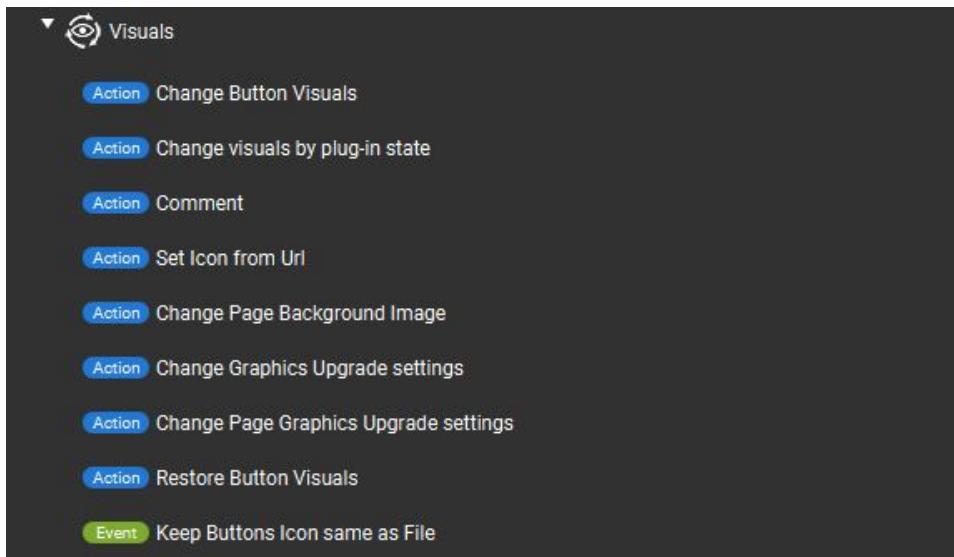


Figure 7.7 : Visuals category

➤ Change Button Visuals

This is an action that allows you to modify the appearance of a button.

User Guide

8. Edition (Using Touch Portal categories)

❖ Using the Navigation category

➤ Go to Page



Figure 8.1 : Go to Page

The dropdown of the previous control allows you to choose a Touch Portal page.

See this action in the button represented by a house and an orange arrow on the **Cessna 172 G1000_1** page.

➤ Open URL

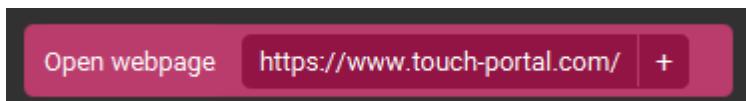


Figure 8.2 : Open URL

- The value of the previous control must be strictly entered, as it represents the address of the page we want to refer to. Here, it is **https://www.touch-portal.com/**

See this action representing a Touch Portal logo on the **Cessna 172 G1000_1**, **Cessna 172 G1000_2** and **Cessna 172 G1000_3** pages.

User Guide

❖ Using the Utility category

➤ Start / Stop Plug-In

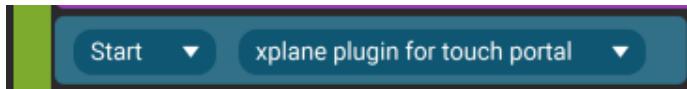


Figure 8.3 : Start / Stop Plug-In

- The first input field offers the **Start** or **Stop** choices.
- The second input field allows selecting a plugin. In our example, we are using the **xplane plugin for Touch Portal**.

See this action in the button representing a JSON file on the **Cessna 172 G1000_1** page.

User Guide

❖ Using the Logic category

This category is complex and very important.

➤ Wait for Timer



Figure 8.4 : Wait for Timer

- The first input field offers a numerical choice of time.
- The second input field offers the choices **milliseconds**, **seconds**, and **minutes**.
- The last input field offers the choices **Default System** and **Experimental System**.

🔍 See this action in the button representing a JSON file on the **Cessna 172 G1000_1** page. It allows waiting between different actions.

➤ If Statement



Figure 8.5 : If Statement

- The first input field allows specifying a Dataref or a another STATE.
- The second input field allows specifying a comparator.
 - **Is equal to**, **is NOT equal to**, **is greater than**, **contains**, etc.
- The third input field allows specifying a value for the comparison.
- The last input field is optional and allows adding other conditions under the **IF**, such as
 - **AND**, **OR**, etc.

🔍 See this action in the button representing an X-Plane icon on the **Cessna 172 G1000_1** page.

User Guide

➤ If Statement + Else

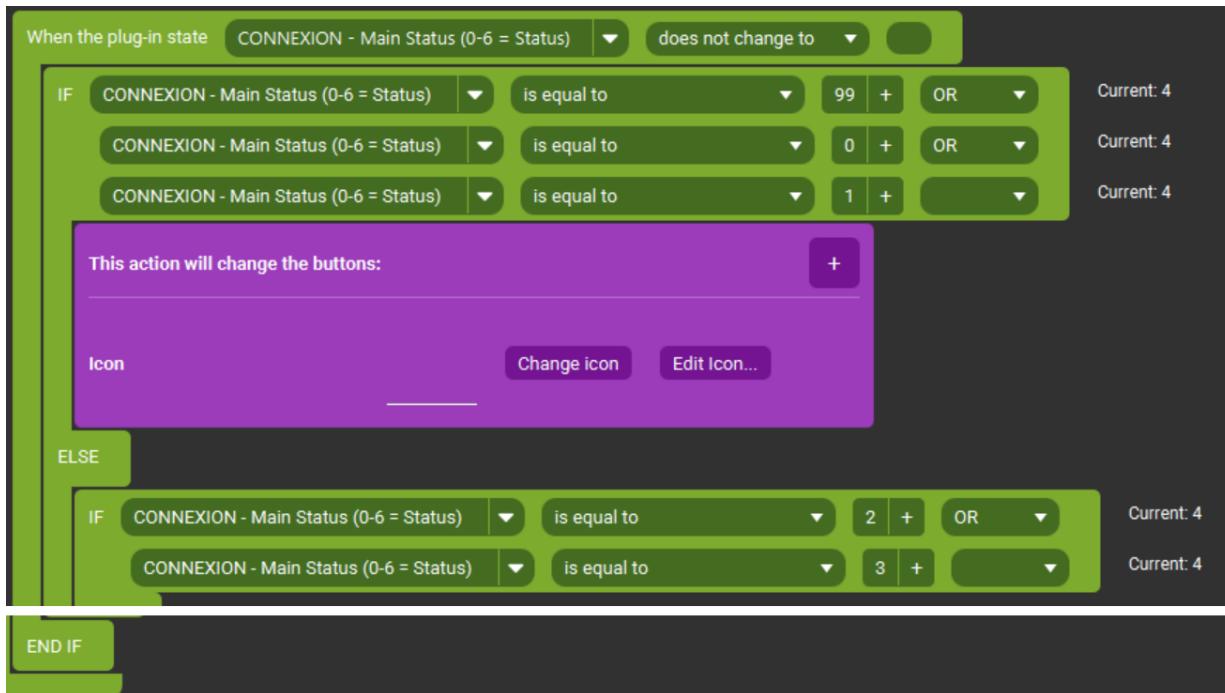


Figure 8.6 : If Statement + Else

This action is identical to the **If Statement** action except that's include an **Else** part.

If the condition is not met, then what's under **ELSE** will be executed. Note that an IF condition always ends with an **END IF** statement.

💡 See this action in the button representing an X-Plane icon on the **Cessna 172 G1000_1** page.

💡 It is possible to nest conditions. See this action in the status button, under the On **Event tab**, and on the **Cessna 172 G1000_1** page.

➤ Else

This action is identical to the **If Statement** action except that it will be used when a condition is not met.

User Guide

➤ Dynamic Text Updater

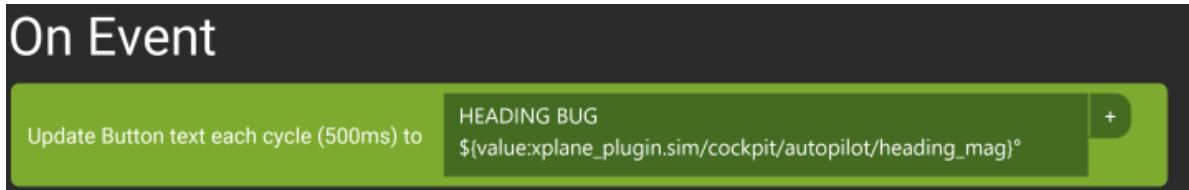


Figure 8.7 : Dynamic Text Updater

This action automatically updates a button by displaying a value every half second. When you click on the "+" in this green box, you choose a description regarding what you want to display.

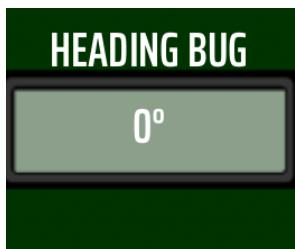


Figure 8.8 The button corresponding to the Dynamic Text Updater in Figure 8.7.

One can put a description (title) to this display. This description is essential to know what we want to display. Thus, we have the description **HEADING BUG**, and in the display box, the value of the Dataref followed by the degree symbol "°".

See this action in the **HEADING BUG** display button in a burgundy area, on the **Cessna 172 G1000_3** page.

User Guide

❖ Usage of the Values & States category

➤ When Plug-in States changes (no values)

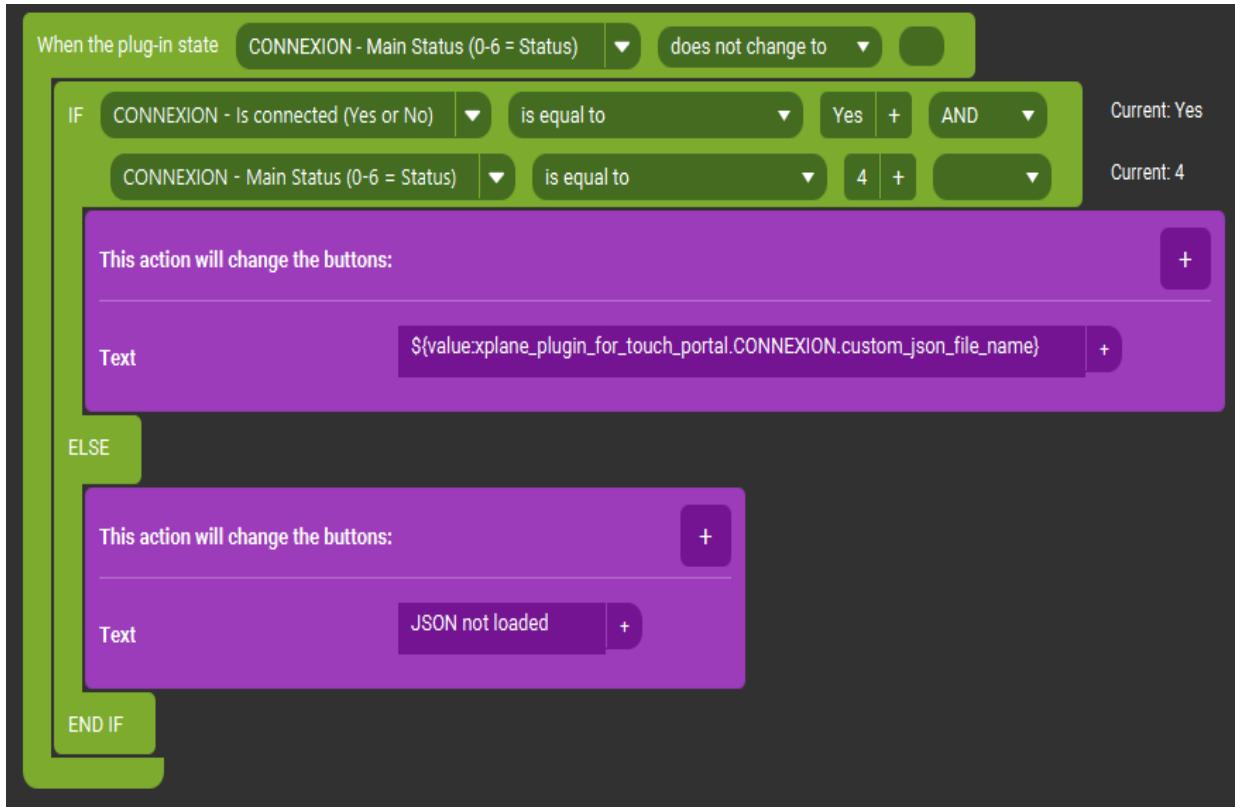


Figure 8.9 : When Plug-in States changes (Rubrique Values & States / Values)

This action is similar to the **If Statement** action, except that it must be placed in the **On Event** tab.

There's no **IF Statements** statement without a **When...**statement. This is a of Touch Portal constrains.

A **When...** action does not allow **OR** or **AND** in its condition, as an **IF Statement**.

This action does not allow showing a value to the end, as an **IF Statement**. We see that the current value of the **CONNEXION Main Status** is **4**.

🔍 See this action on the button representing an X-Plane icon on the **Cessna 172 G1000_1** page.

User Guide

➤ When Plug-in States changes (avec valeur)

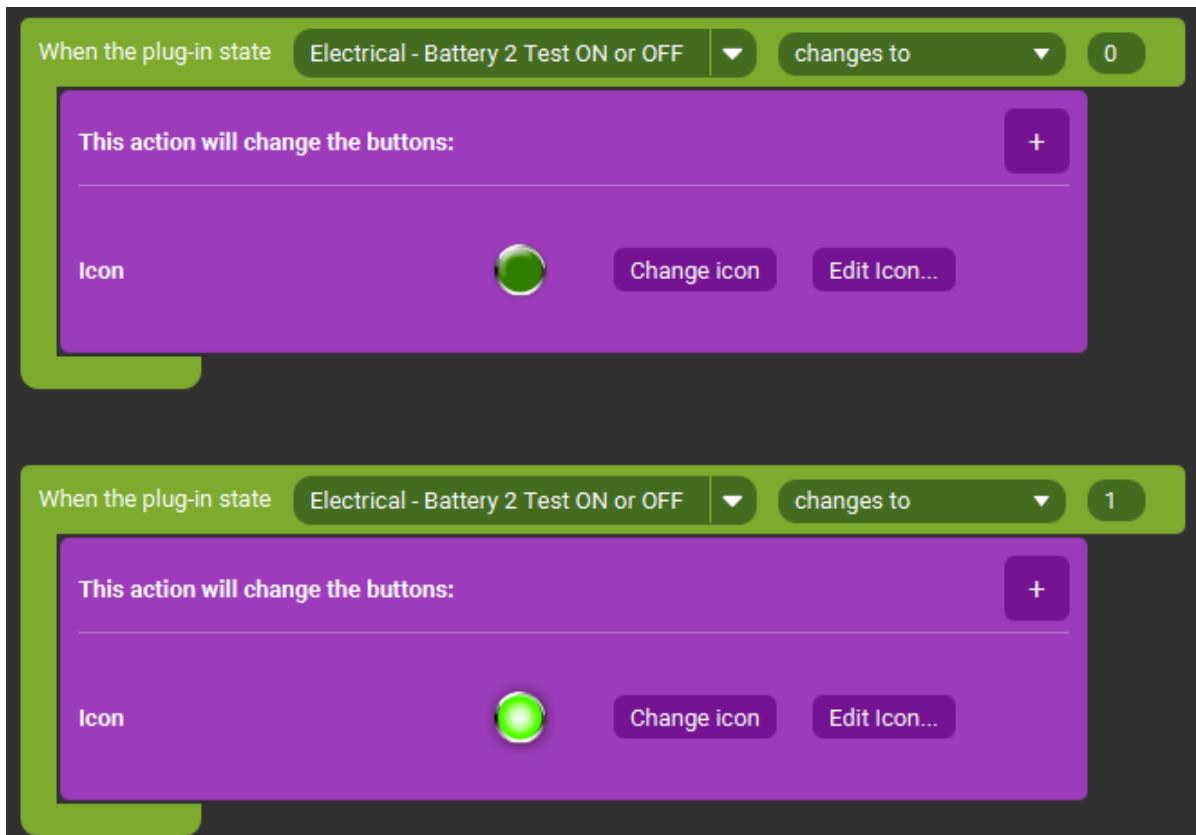


Figure 8.10 : When Plug-in States changes (category Values & States / Values)

We could enter a value in the **When** action, in the third input field. However, this will not allow displaying the current value at the end of the **When** line.

The trick with this **When** action that allows displaying the current value is to use "**does not change to**" in the second input field and to **leave the last input field empty**. This way, below, we can put **IF, ELSE, END-IF** actions that will be able to display current values (*see Figure 8.9*).

🔍 See this action in the green light button labeled **TEST** in **Cessna 172 G1000_2** page.

User Guide

❖ Utilization of the Visuals category

➤ Change Button Visuals

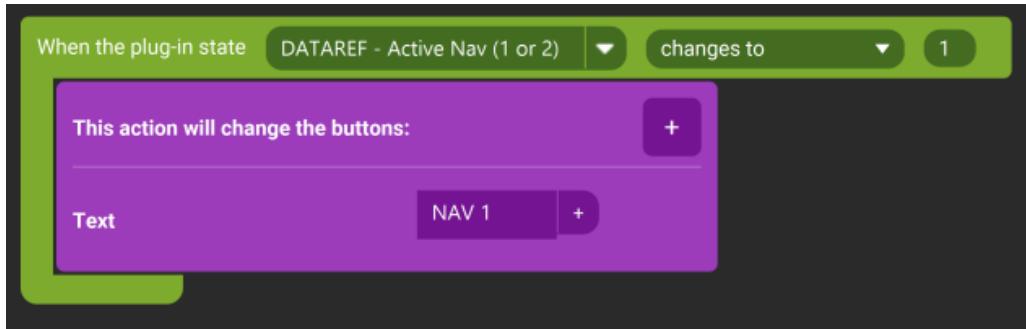


Figure 8.11 : Change Button Visuals

This is an action that is widely used and very important. It allows changing the appearance of a button and thus animating a button using conditions. It can be used in the **On Pressed** and **On Event** tabs.

When clicking on the "+" in this purple box, we have different choices available to us to change the appearance of a button. We can add multiple choices per action.

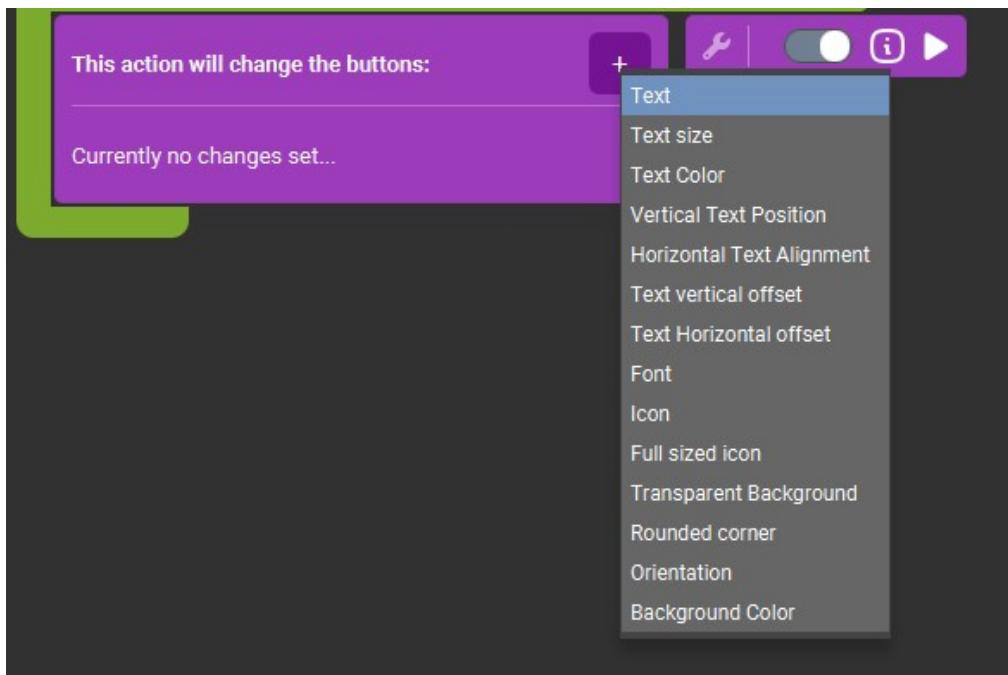


Figure 8.12 : Change Button Visuals (choix)

We can add text, change the text size, change text position, and change its font. We can also add an icon (widely used in my pages) and change its dimension. Try out these several options.

User Guide

9. The Importance of the Visuals Category (with icons)

❖ Vertically juxtaposed buttons

➤ The two Throttle buttons

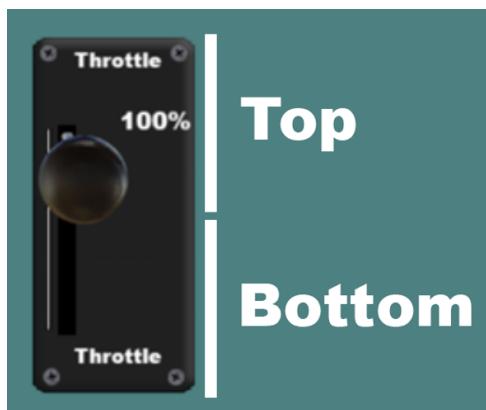


Figure 9.1 : Throttle into Two Buttons

I've created several icons to be used on two buttons. We have an icon for the top part (**Top**) and another icon for the bottom part (**Bottom**).

In the "XP TP Levers Icons" pack, we will find **10** icons for these two buttons.

The naming convention for the icons is that **_1_** represents the top part and **_2_** represents the bottom part. Thus, we will find these icons in this pack with the names **CessnaThrottle_1_** and **CessnaThrottle_2_**.

Furthermore, each icon represents a specific value of a Dataref. The possible values for the **Throttle** are "**0.0, 0.25, 0.5, 0.75, and 1.0**".

At the end of the names of these icons, we have a values. Note that the number **1.0** representing a value will be substituted by the number **099** (for example, **CessnaThrottle_1_099.png**).

The icons are displayed in a specific order in their directory, allowing to see the state progression into "XP TP Levers Icons" library.

See **Engine - Throttle #1 Ratio** and the following **ID**:

sim/cockpit2/engine/actuators/throttle_ratio[0] in the default.json file.

Important: When clicking the top button, the **Throttle lever** moves up, and when clicking the bottom button, it moves down.

See this in **Cessna 172 G1000_1** page.

User Guide

➤ Inside the two Throttle buttons



Figure 9.2 : The conditions allowing the use icons in Throttle buttons

Each condition of these buttons allows displaying an icon representing a state.

Depending on the obtained values, the button displays a specific icon. This is the basis of the animation.

Thus, on Touch Portal, throttle levers, flaps, etc., are simulated.

Some icons are dedicated to Airbus (their blue colors). Others are usable in aircraft such as the Baron.

User Guide

➤ Why use two Throttle buttons instead of a slider?

Know that, if we take the example of our **Cessna 172 G1000_1** page, it uses a mirrored effect in the display compared to X-Plane 12.

If we press the **BEACON** button on our page, this button will be toggled in the simulator's cockpit side.

If we press the **BEACON** button on the simulator side, the button on the Touch Portal side will be toggled.

So, for practical reasons, to simplify processing and to offer acceptable limits, I will use fixed values for the **Throttle, which are 1/4** between the values **0.0** and **1.0**.

Regarding the description **Engine - Throttle #1 Ratio** of these buttons, we have the following **ID: xplane_plugin.sim/cockpit2/engine/actuators/throttle_ratio[0]** in the default.json file.

Two important fields are assigned to this reference and are used for any **XP TP Levers control**. Therefore, we have:

```
"xplane_update_min_value": "0",  
"xplane_update_max_value": "1",
```

When the user presses the **Throttle** on the simulator side, X-Plane monitors the value. If this value changes compared to its previous value, X-Plane will send only quarters value, namely **1.0, 0.75, 0.5, 0.25**, and **0.0**. Thus, these values are represented on the Touch Portal side page.

Important: The plugin is able to figure out **1/4** values between a minimum and maximum value, for greater flexibility of use.

If we have the following:

```
"xplane_update_min_value": "1.0",  
"xplane_update_max_value": "10.0",
```

the returned values will be **1.0, 3.25, 5.5, 7.75**, and **10.0**.

User Guide

❖ Horizontally juxtaposed buttons

➤ The two Starter (Magneton) buttons



Figure 9.3 : Magnetos in Two-Button configuration

I have created additional icons intended for use on two buttons. Here we have an icon for the left part (**Left**) and another for the right part (**Right**).

In the "XP TP Rotary Knobs Icons" pack, we will find **10** icons for the use of these two buttons.

The naming convention for the icons is as follows: **_1_** represents the left part and **_2_** the right part.

Thus, in this pack, we will find these icons with the names **Magnetos_1_** and **Magnetos_2_**.

Additionally, each icon represents a specific value of a Dataref. The possible values for the **Magnetos** are "**0, 1, 2, 3, and 4**".

Regarding the description **Engine - Starter 1 Key Position** for these buttons, we have the following **ID: xplane_plugin.sim/cockpit2/engine/actuators/ignition_key[0]** in the default.json file.

Important: When clicking the left button, the **Magnetos key** moves to the left, and when clicking the right button, it moves to the right until a 3-second start is performed.

See this action in the **Magnetos** buttons on the **Cessna 172 G1000_1** page.

User Guide

➤ Inside the two Magnetos buttons



Figure 9.4 : Conditions for using the icons in these two Magnetos buttons

Each of these conditions, for these buttons, allows displaying an icon representing a state.

Based on the obtained values, the button displays a specific icon.

User Guide

10. Using format codes in the default.json file

❖ List of possible format codes in the JSON file

The plugin offers some format codes for some situation.

- **D1** : Removes the last decimal (99.35 becomes 99.3).
- **D2** : Converts the integer value to a real number with two decimals (11705 becomes 117.05).
- **D3** : Converts the integer value to a real number with three decimals (126400 becomes 126.400).
- **ND** : Removes the decimal point (2312.99 becomes 231299).

❖ Why use format codes

➤ NAV frequencies

Datarefs can contain specific data. We find them on our **Cessna 172 G1000_3** page, in the category concerning **NAV**. Everything around **NAV** is into an olive-colored area.

If you look below the **ACTIVE** display boxes, you will find the following **IDs** in the default.json file:

`xplane_plugin.sim/cockpit2/radios/actuators/nav1_frequency_hz` and
`xplane_plugin.sim/cockpit2/radios/actuators/nav2_frequency_hz`.

In X-Plane 12, both of these references are in integer format. If we have the integer value of **11225** in this Dataref, we must tell our plugin to treat this field with two decimals (**NAV** frequencies).

Therefore, in the default.json file, we have the "**touch_portal_format**" : "**D2**" field.

This field tell our plugin to treat this Dataref with a specific format. So the result will be **112.25** (**NAV** frequencies).

Please note that we can display more than one information in a dialog box. We will also have `xplane_plugin.sim/cockpit2/radios/indicators/nav1_nav_id` representing the NAV1 frequency id.

User Guide

➤ COM frequencies

We find them on our **Cessna 172 G1000_3** page, in the category concerning **COM**. Everything around **COM** is into an blue-colored area.

If you look below the **ACTIVE** display boxes, you will find the following **IDs** in the default.json file:

`xplane_plugin.sim/cockpit2/radios/actuators/com1_frequency_hz_833` and
`xplane_plugin.sim/cockpit2/radios/actuators/com2_frequency_hz_833`.

In X-Plane 12, both of these references are in integer format. If we have the integer value of **126400** in this Dataref, we must tell our plugin to treat this field with two decimals (**COM frequencies**).

Therefore, in the default.json file, we have the "**touch_portal_format**" : "**D3**" field.

This field tell our plugin to treat this Dataref with a specific format. So the result will be **126.400** (**NAV frequencies**).

User Guide

11. Specificities of some buttons in demo pages

❖ Cessna 172 G1000_1 page

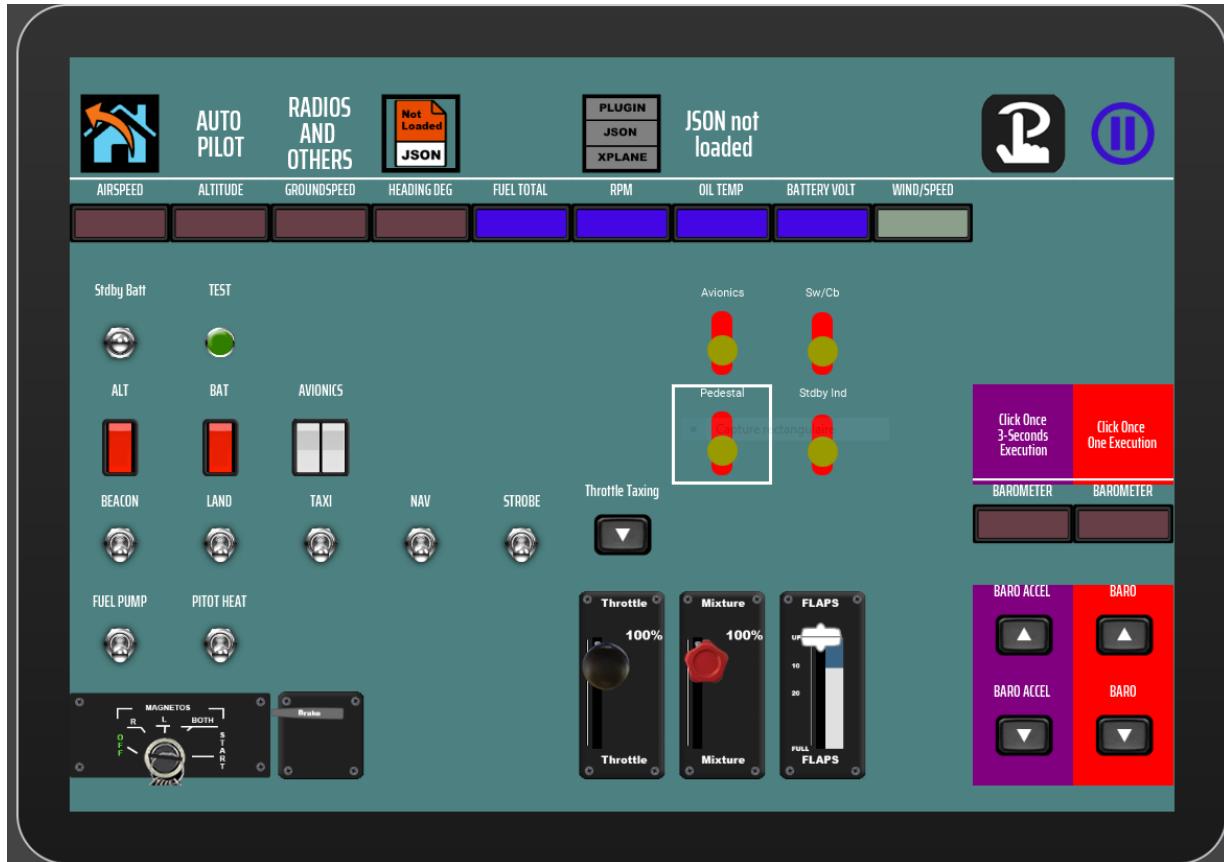


Figure 11.1 : First page (main page)

➤ Throttle

We know that the values of the **Throttle** are **1.0, 0.75, 0.5, 0.25, and 0.0**.

Therefore, for taxiing, **25%** of the **Throttle** value is too fast for this operation.

I have created a button above the **Throttle**, named **Throttle Taxiing**, allowing to decrease the **Throttle** value to 10%. You can adjust this value according to your preferences.

➤ BAROMETER

Clicking on the **barometer** display box will provide the standard barometric value or toggle this value with the local value.

One window displays the barometric values in **InHG** and the other in **HPA**.

It is necessary to press the barometer adjustment buttons without holding a button.

User Guide

❖ Cessna 172 G1000_2 page



Figure 11.2 : Second page (auto pilot page)

➤ NAV and COM

In the category concerning **NAV** and **COM**, we have buttons with double green arrows. These buttons allow toggling between the **Active** and **Standby** values. (also included on page 3 and the values must be changed only on page 3).

➤ HEADING BUG

Clicking on the display box regarding **HEADING BUG** will synchronize the Heading bug with the current Heading. It is important to hold the adjustment buttons of the **HEADING BUG** to adjust it. (represented on page 3 as well).

➤ AP-ALTITUDE

Clicking on the display box regarding **AP-ALTITUDE** will initialize the altitude value to zero.

It is important to hold the adjustment buttons of the **AP-ALTITUDE** to adjust it. (represented on page 3 as well).

User Guide

❖ Cessna 172 G1000_3 page

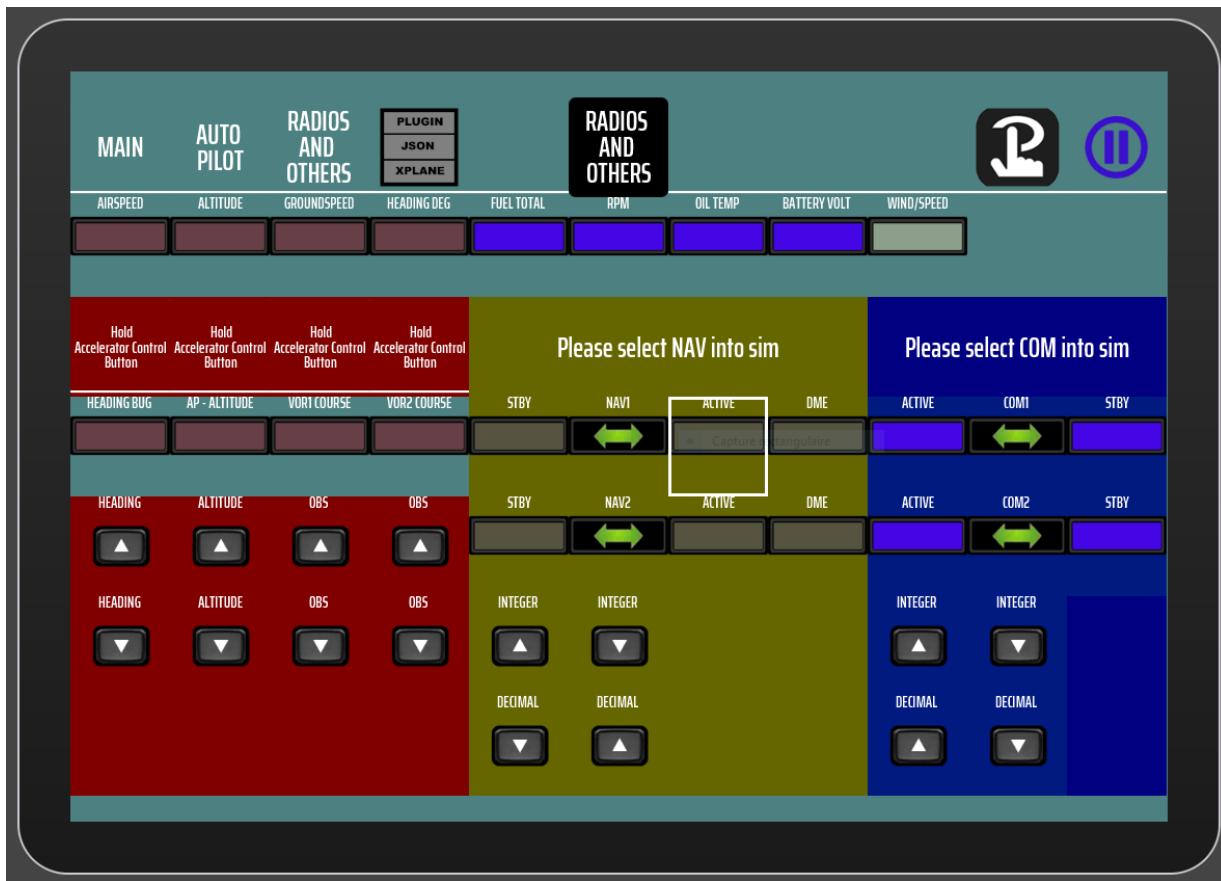


Figure 11.3 : Third page (radios and other controls)

➤ HEADING BUG

Clicking on the display box regarding HEADING BUG will synchronize the Heading bug with the current Heading. It is important to hold the adjustment buttons of the **HEADING BUG** to adjust it.

➤ AP-ALTITUDE

Clicking on the display box regarding **AP-ALTITUDE** will initialize the altitude value to zero.

It is important to hold the adjustment buttons of the **AP-ALTITUDE** to adjust it.

User Guide

➤ VOR1 and VOR2 COURSE

The adjustment of the NAV1 and NAV2 courses is done by holding down the buttons.

If you click on the **VOR1 COURSE** or **VOR2 COURSE** display box, this will synchronize the course.

➤ NAV1, NAV2, COM1 and COM2

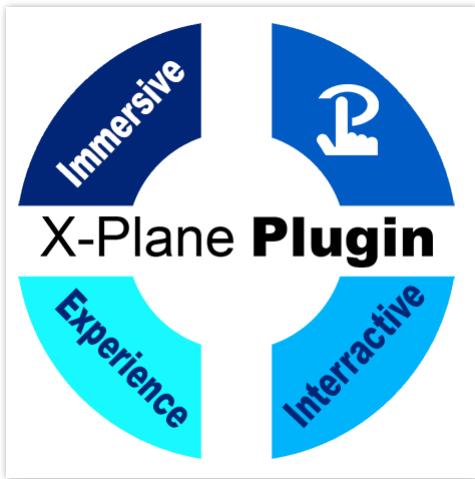
The **STDBY** frequencies are adjusted using the **INTEGER** buttons (the whole part is changed) and **DECIMAL** (the decimal part is modified).

In the section about **NAVs** and **COMs**, we have buttons with double and green arrows. These are for switching the Active and Standby values.

As you can see, there are several ways to offer interesting options without overloading pages.

User Guide

12. Recommendations for use



❖ First recommendation:

Here's a recommended method for building your aircraft pages. You should copy button from the **Cessna 172 G1000_1**, **Cessna 172 G1000_2** or **Cessna 172 G1000_3** to avoid errors.

Personally, copy the connection buttons to the first page of your aircraft.

Remember that I've provided a specific layout for my pages. It's up to you to create pages according to your preferences.

There are also connection icons with different formats in the library.

As you create pages, you'll understand the plugins and touch portal better.

❖ Second recommendation:

Like any editor, it's important to save your changes frequently while editing.

Additionally, Touch Portal, in its Settings menu, allows you to save pages to a special backup library.

Visit the Touch Portal website for further details.

User Guide

❖ Last recommendation:

Pay close attention to the modifications you make. There are many tutorials available on Touch Portal.

Visit the Touch Portal website under the Help category for further details.

➤ Learning programming with Touch Portal:

If you're not familiar with programming and with conditions such as **When**, **If**, **Else**, and **End-IF**, know that there are learning websites explaining this in pseudo-code format:

Touch Portal: [Touch Portal Logic Functions](#)

In French: [Site d'apprentissage](#)

In English: [Learning Site](#)

Happy flying and enjoy using my Touch Portal Plugin!