How to use LAC's internal menus to map the buttons of your joystick or console game controller to LAC's flight and communication functions:

LAC implements 45 distinct, named, *flight functions* and several *communication functions* as it simulates World-War II aircraft in combat.

You will want easy access to these functions in flight, and the best way to get this easy access is to "map" the most important LAC functions to keyboard keys, joystick buttons, or game controller buttons that are positioned for your easy, logical reach. If you get this right, you'll have a HUGE advantage in combat because you'll develop *muscle memory* for the most important flight and communication functions and you won't lose time trying to remember or look up their locations, regardless of the type of joystick, console game controller, or keyboard in use. *All of your controls will be positioned exactly where YOU want them.*

LAC comes preconfigured with its own "standard" control layout based on a standard keyboard, numeric keypad, and the most popular flight joystick ever made: the "Logitech Extreme 3d Pro". Here is a picture of that joystick that we've labeled with LAC control layouts:

And here is a second picture with one change:

As you can see, those 2 pictures are nearly identical. However, if you look carefully at the joystick base of the *lower* picture, you'll notice that it maps mumble's "PTT" ("Push To Talk") function to one of the buttons that are unused in the *upper* picture.

LAC's ability to map joystick buttons (or console game controller buttons) to LAC's functions *through its menus* is new. We couldn't do this through our menus before we released version 9.93 in June of 2026, and we couldn't map mumble's "Push to Talk" function to a joystick button by any means until LAC version 9.94 appeared in early August 2026, after mumble released their new version 1.5.7 supporting "Push to Talk" through their new "RPC" Application Program Interface.

Of course, we've *always* been able to map LAC's flight and communication functions to keyboard keys and joystick buttons by editing LAC's "LacControls.txt" configuration file, and that power remains, unchanged, in these newer LAC versions. That method is well documented and well understood in the LAC community, so we aren't discussing it further here, even though these newer versions of LAC expand the list of functions that can be mapped.

Let's concentrate on using LAC's menus to map a joystick button to one of LAC's flight or communication functions. Mapping LAC's functions through its menus is easier than editing the LacControls.txt configuration file, and LAC's menus have now been expanded to handle all of the most important functions. You'll only need to edit the LacControls.txt file if you're attempting some kind of control mapping that is really unusual.

As an example, we'll show exactly *how we map a joystick button for use with mumble's "Push-to-Talk" facility through LAC.* (We like this example because it was never possible until mumble improved their "RPC" API in support of "Push-to-Talk", and we are pretty sure LAC is the only flight simulator that has been enhanced to implement it, and our implementation works even in new "Wayland" LINUX environments where the old "keyboard shortcut" Push-to-Talk method fails for everybody else.) *In this example, we'll frequently refer to "joystick buttons". If you are using a Console Game Controller instead of a traditional joystick, you can map your Game Controller buttons using exactly the same sequence that we describe for joysticks.*

From the "SETUP OPTIONS" button of LAC's main menu, click on "GAME". If necessary click on the "CONTROLLER" button until it cycles to your chosen controller hardware. For our purposes in this discussion, we'll assume you cycle to "JOYSTICK". Then click "KBD & JOYBUTTON MAP".

A list of all of LAC's mappable flight and communication functions appears. Each function is labeled with a descriptive name. If any keyboard key is already configured to activate each listed function, it is shown adjacent each name.

Click on one of the listed function names. In our example we'll click on "MUMBLE PTT". That function name (and the name of any keyboard key that has already been associated) is highlighted as soon as you click on it.

Now you must make a decision that isn't fully obvious: You must choose between 2 options:

Option 1 of 2: Map the highlighted function ONLY to a keyboard key. Or
Option 2 of 2: Map the highlighted function to a joystick (or controller) button AND to a keyboard key.

If you decide on "Option 1" because you only want to map the highlighted function to a keyboard key, just tap that keyboard key. You'll hear 2 pairs of quick "beeps". One will be a "happy beep" (higher pitch) and the other will be a "sad beep" (lower pitch). This is LAC's way of telling you that only ONE of the TWO possible mappings was done for you. If the keyboard key you designated is different from whatever was previously configured for the highlighted function, you'll see the name of that keyboard key replace the name of any key that was configured previously. Otherwise, if the same keyboard key was already designated, you won't see any visible change (but you will still hear the "happy beep" followed immediately by the "sad beep" as just described).

If you decide on "Option 2" because you want to map the highlighted function to BOTH a joystick button AND a keyboard key, you must **first** tap the desired joystick button. You'll hear a single quick "happy beep" pair at that point (which is LAC's acknowledgement that it has mapped the indicated joystick button to the highlighted function), but you won't see any displayed change yet. **Then** you must tap the desired keyboard key. You'll hear TWO "happy beep" pairs, confirming that LAC has mapped the highlighted function to TWO controls, exactly as you would expect. You'll see the name of that keyboard key updated in the displayed list, adjacent the name of the highlighted LAC function that it will control.

You can continue this pattern for additional LAC functions until you have everything mapped to your satisfaction. Of course, you'll need to pay attention to ensure that you haven't mapped any single keyboard key or any single joystick button to more than one LAC function.

Any functions you don't explicitly change (as described above) will remain mapped according to LAC's default control mappings and ought to match the diagrams we've already shown above.

When you've mapped all of the controls that you want to customize, there is ONE MORE STEP that isn't perfectly obvious: it is necessary to cycle the "CONTROLLER" button through all of its options until the display matches your control hardware again. For our purposes in this discussion, we will assume you have a joystick, so you'll click "CONTROLLER" until it cycles through all of the other options and ends up displaying "JOYSTICK" again. *It is not sufficient to simply leave LAC's menus while the "CONTROLLER" continues to show "JOYSTICK"*. You must explicitly click on "JOYSTICK" and watch it cycle through all of it's other possible states ("CONSOLE", "MOUSE/KBD" and back to "JOYSTICK"). After the "CONTROLLER" button has cycled out of "JOYSTICK" mode, cycled through all of the other options, and back to "JOYSTICK", you'll be able to click "FLY A MISSION" and resume flight in any of the available missions as you'd expect.

After this mapping exercise you need NOT exit from or restart LAC. Upon entry into any of LAC's missions you should find your Joystick and Keyboard mapped to LAC's functions exactly as you directed. The changes you've made will be retained in the "LacControls.txt" configuration file and will remain in force for future LAC sessions.

If anything goes wrong, you can always reset LAC back to its default mappings by deleting the LacControls.txt file from the hidden ~home/.LAC folder as described elsewhere in our documentation. The next time you start LAC, it will rebuild that LacControls.txt file for you, using its default mappings.