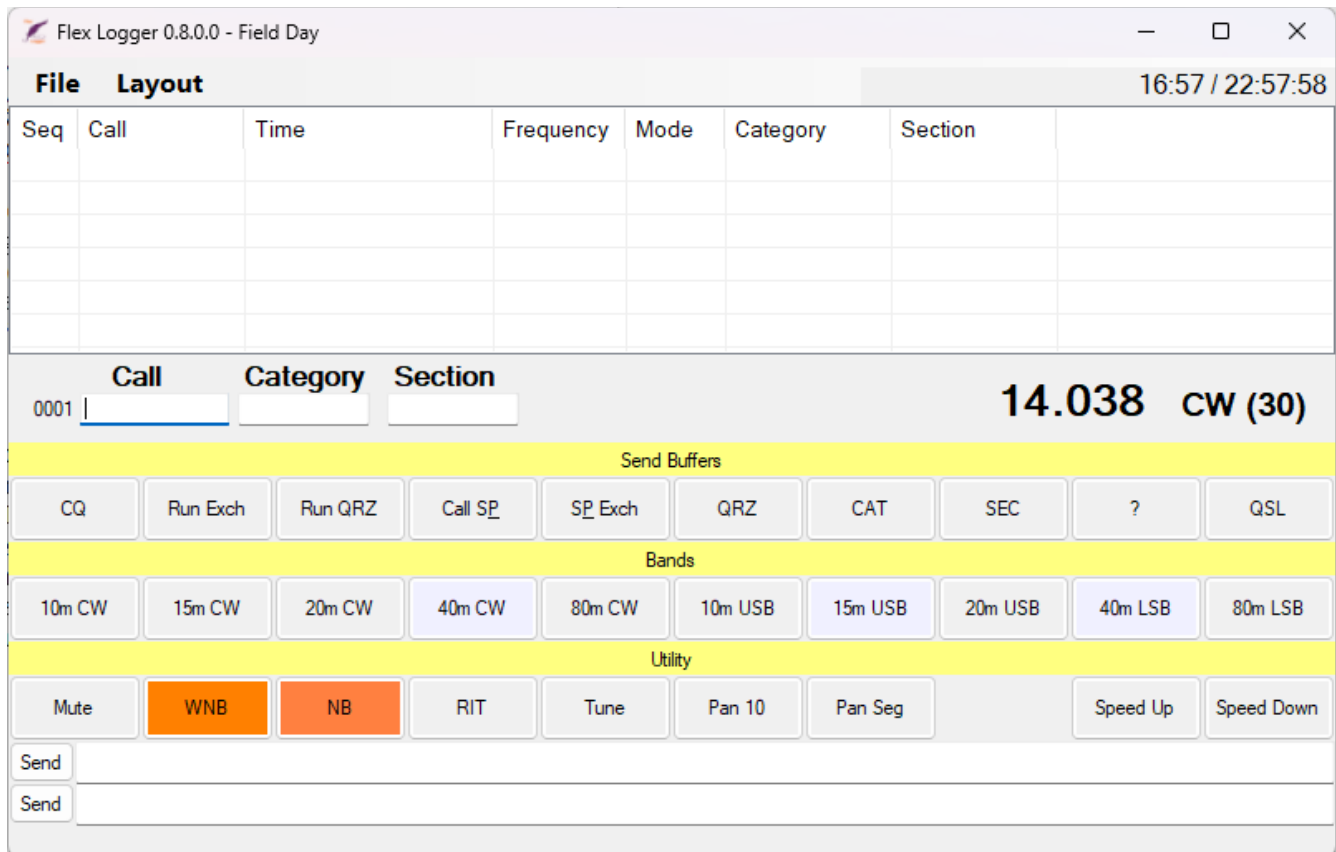


Flex Logger (V0.8)

Keith Regli K7KAR
keith@kregli.com

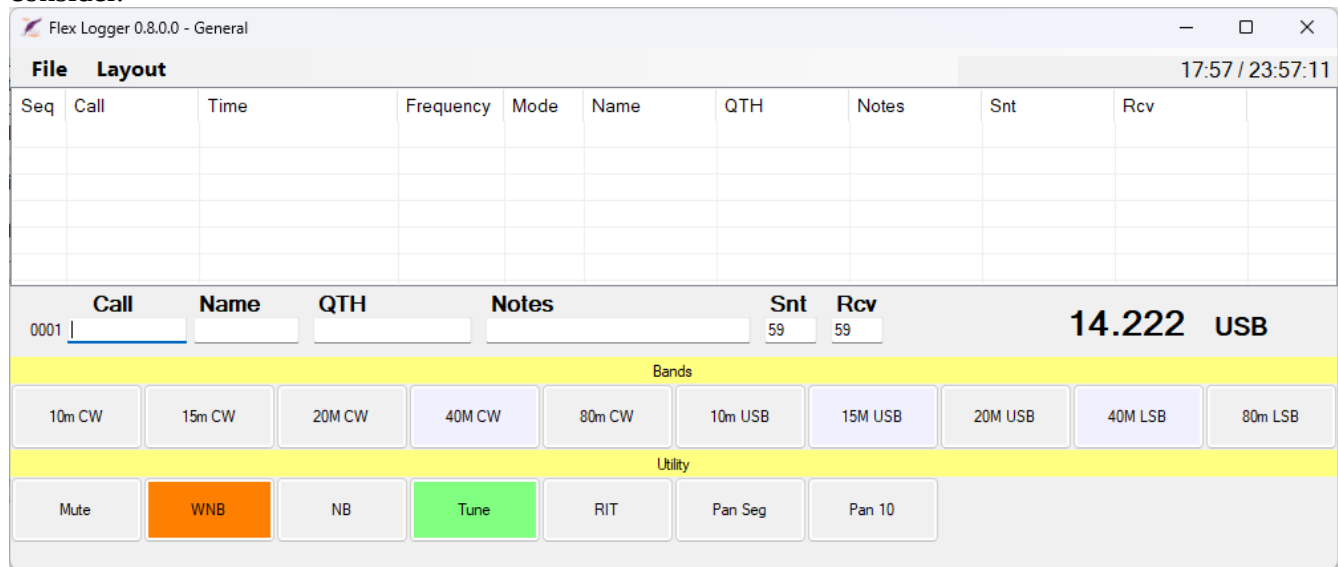
Flex Logger is an early stage logging program that is based off of Flex Buttons. It may not get past the general logger stage as there are several very good logging programs out there and to get to their level of capability is a huge job. For me it is an interesting project and if nobody else ever uses it that's fine. I personally use WriteLog when I'm contesting. The current version of Flex Logger is OK for general logging on phone and CW and has enough features to work on Field Day for those modes. The screen below shows what may be my layout for Field Day. It can be used in conjunction with the CWTY Decoder application ([WD6CNF](#)) – highlight something in the CWTY decoder then double click in the logging field (say CALL) and the highlighted text will transfer.



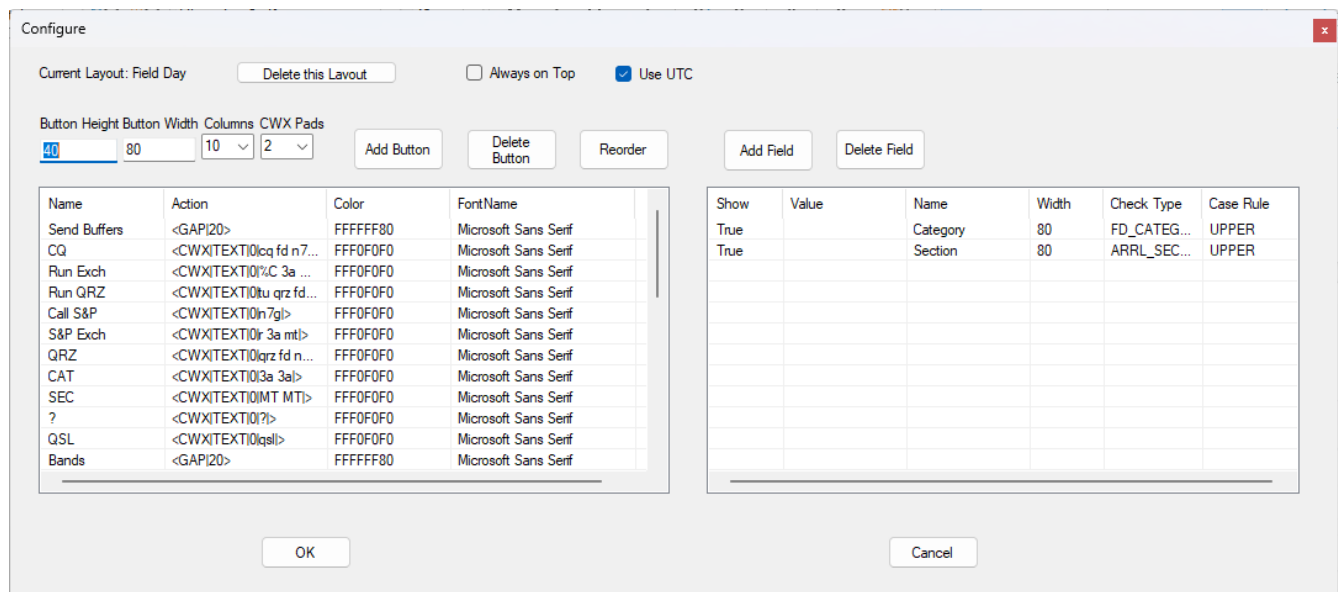
The log is at the top and the entry fields are next. It shows the frequency and mode. In the case of CW it shows the internal keyer speed. The first row of buttons will send buffers using the internal keyer. The buffers can use the entered call (%C), sequence number (%S) and first RST field (%R). The second row of buttons just selects bands and modes. The third row of buttons turns several functions on and off as well as allowing you to increase/decrease the keyer speed. Then there are two fields that allow ad hoc entry of text to send. If you don't like this layout you can change almost everything.

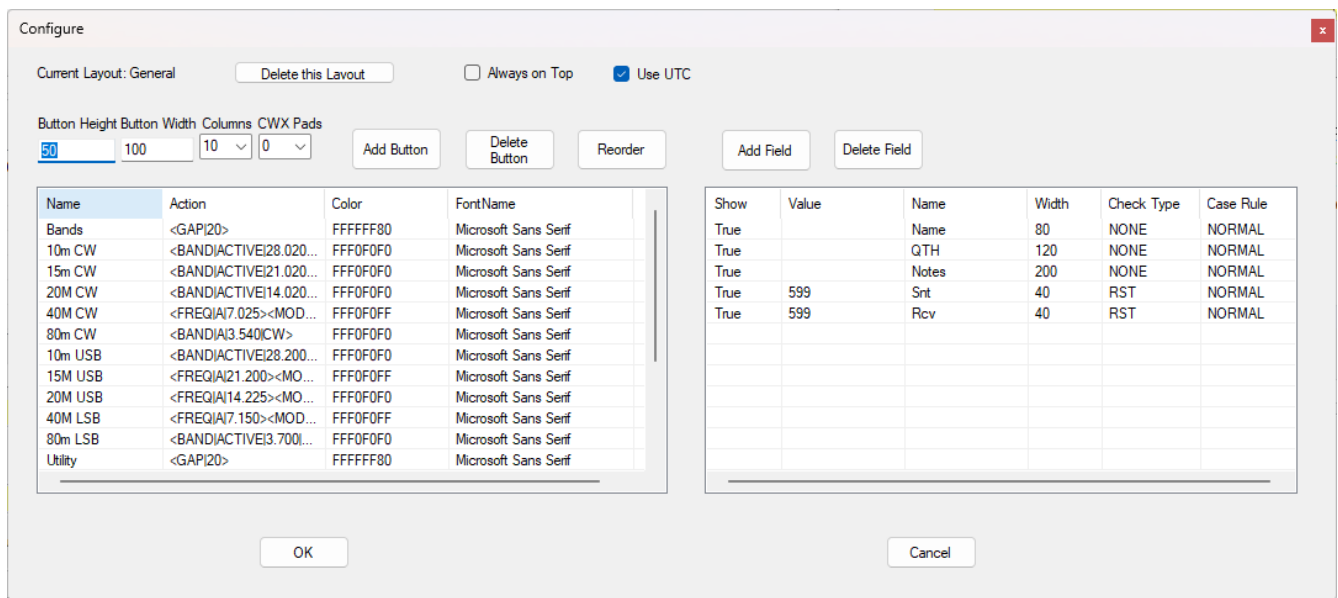
Pretty much everything can be configured and the configuration screens and available buttons are very similar to Flex Buttons (I used cut and paste a lot). I haven't tried to make it work with multiple slices – it will get information for the frequency and mode from the “active radio”.

The screen below is a general logger layout to give an idea of what kind of alternatives you might consider.

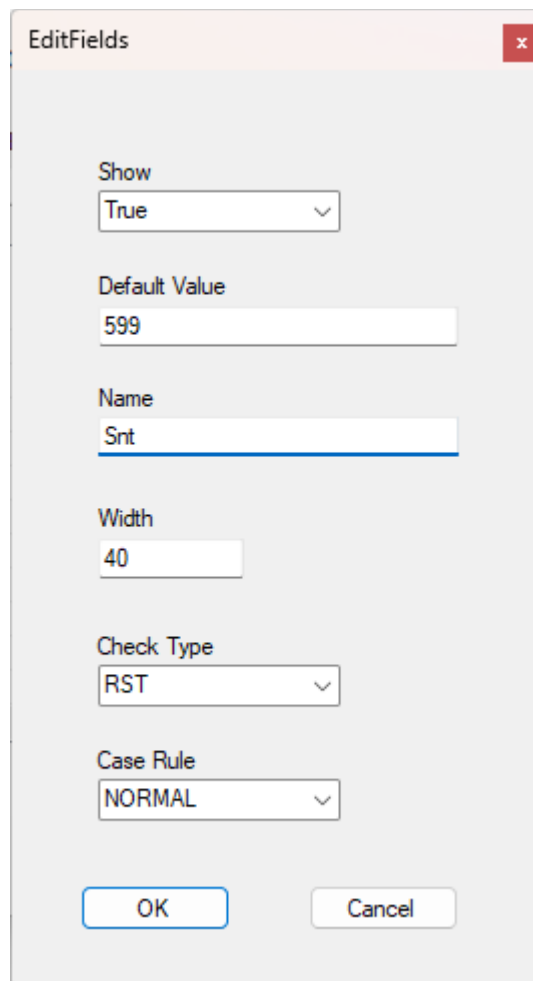


Since this is very preliminary, I'll just include the Flex Buttons documentation below to show how to set up the buttons. Before that I'll show the screens used to set up the logging fields. The first shows the configuration screen with the Field Day layout, while the second shows the general logger. The left side of the screen is pretty much the same as for Flex Buttons. The right side is used in a similar fashion to select the logging fields. There is also a checkbox to select local or UTC time. You can add or delete logging fields in a manner similar to Flex Buttons. To edit a field double click of the field name.





Double clicking on one of the fields shows the following screen.



The default value will be inserted automatically if set. If you select “False” for “Show” the field will not appear on the screen but will be filled by the default value. There are a very small number of checks that you can choose. You can also force the field to upper or lower case.

The installation package has registry files to add Field Day and General layouts in addition to the default that is installed automatically. The ArrlSections.dat file is just a text file with a list of sections.

That’s about it for now. For anything else take a look in the Flex Buttons documentation below.

Flex Buttons documentation.

Simplicity of operation and screen real estate are things I care about when I’m operating. This little utility simplifies my screen layout and reduces the number of clicks I need to accomplish some common functions. I no longer have PGXL and TGXL utilities running all the time and don’t need to run SmartSDR for mouse access to functions. The pictures below show three different layouts that I set up to demonstrate various possibilities. You can have as many or as few of these buttons as you like and you can arrange them as you like. **Note:** It is best to start your radios, PGXL and TGXL *before* starting FlexButtons. It will only pick up a TGXL at startup.



ONE CLICK!

You can combine two or more functions on one button, such as: on slice A go to 14.329Mhz, USB with a 2700 Hz filter, the panadapter zoomed to 10 kHz, set PoWeR to 25 W, BYPASS the ATU, set the TGXL to TUNE and the PGXL to OPERATE. And now you’re ready for the Flex Net.

The image below shows a “full boat”: two radios, multiFlex, a PGXL and a TGXL in use at the same time. We’ll get to all of this as we go on, but for now let’s stick to a single radio, no multiFlex and one or two slices. I haven’t looked into remote operations yet, so that may or may not work. Notice that the title bar indicates the nickname, callsign and station (from multiFlex).



The current set of exposed functions includes:

ANF: Automatic Notch Filter on/off/toggle
 ANT: set the RX, TX or RX/TX antenna
 ATU: TUNE or BYPASS
 BAND: Set a frequency and mode
 BLANK: Insert a blank space for a button
 CWX: Send a pre-programmed buffer or text
 DAX: Set the DAX channel
 DIV: Diversity on/off/toggle
 FDX: Full duplex on/off/toggle
 FIL: Set the filter bounds
 FREQ: Set the frequency in MHz
 GAP: Insert a blank row for a title
 MEM: Select a memory profile to use
 MODE: Set the operating mode
 MOX: Turn MOX on/off/toggle
 MUTE: Set the mute state on/off/toggle
 NB: Turn the Noise Blanker on/off/toggle
 NR: Noise Reduction on/off/toggle
 PAN: Set the panadapter zoom to Band, Segment or a specific width in MHz
 PGXL: Switch the active PGXL to operate or standby
 PWR: Set the transmit power level
 RIT: Set the offset in Hz for the RIT
 RUN: Run a program with arguments
 SLICE: Set the active transmit slice
 SCAN: Set timer and buttons to cycle through
 SPEED: Increase or decrease the CW speed
 SPOTS: Clear all spots
 STEP: Set the dial step rate in Hz
 TGXL; Switch the active TGXL to operate or standby
 TNF: All Tracking Notch Filters on/off/toggle
 TUNE: Radio TUNE on/off/toggle
 VOX: VOX on/off/toggle
 WIDTH: Set the filter to a width in Hz
 WNB: Turn the Wideband Noise Blanker on and off
 XIT: Set the offset in Hz for the XIT

The functions that turn things on and off generally allow you to “toggle” as well. I’ve get the functions I’d like with a few that were suggested by others. If you don’t see something that you use often, let me know. I don’t want to include things like changing the nickname, but frequent actions are fair game.

The image on the top left (Default) is what you get when you first install FlexButtons. The image below that (Generic) allows you to select several frequencies and modes, manage a TGXL and a PGXL, zoom the panadapter, zero RIT and toggle audio mute. The image on the right (Optional) throws in some heading (GAP) buttons along with some CWX buffers. You can setup any of these as a template to work from when making your configuration for FlexButtons. I suggest starting with just two or three buttons and use it a bit, then make another layout with things that will be helpful to you. Starting with all bands, all modes is tedious and painful when you decide you want to make a change.

Installation

You can download this documentation and the software from my [website \(http://www.kregli.com\)](http://www.kregli.com). It’s free – if you feel the need to pay for it, donate to some good charity.

Once you get the zip file (FlexButtons.zip) create a directory in any convenient location (e.g. C:\Programs\FlexButtons). Then extract the files from FlexButtons.zip to that directory (right click FlexButtons.zip and select Extract All...). Depending on the privileges you have, you could run into your security system. You’ll have to work your way around that. Choosing a directory in your documents folder may make that easier.

You’ll see FlexButtons.exe, FlexButtons.pdf (this document), FlexButtonsRepair.exe, six dlls from Flex, and some “.reg” files. Ignore the *.reg and *.dll files for now.

Double click on FlexButtons.exe and you should see FlexButtons appear in the upper left of your screen with the default set of buttons (be patient, it takes a few seconds to search for radios). The information about the buttons is stored in the Windows Registry and the program will setup the necessary keys when it runs the first time (for those who know about the Registry it’s [HKEY_CURRENT_USER\Software\K7KAR\FlexButtons]). If your account is short on privileges, you may have to use “Run as Administrator...” the first time you run the program. Windows will tell you. It may also object (the first time) with a warning that the program is unsigned – this is true. You’re allowed to run it anyway (I don’t feel the need to register it in the Microsoft Store). See the description of FlexButtonsRepair.exe below.

If your radio is on and connected to the same LAN as your computer the buttons should work. With this base you can change the configuration to meet your needs.

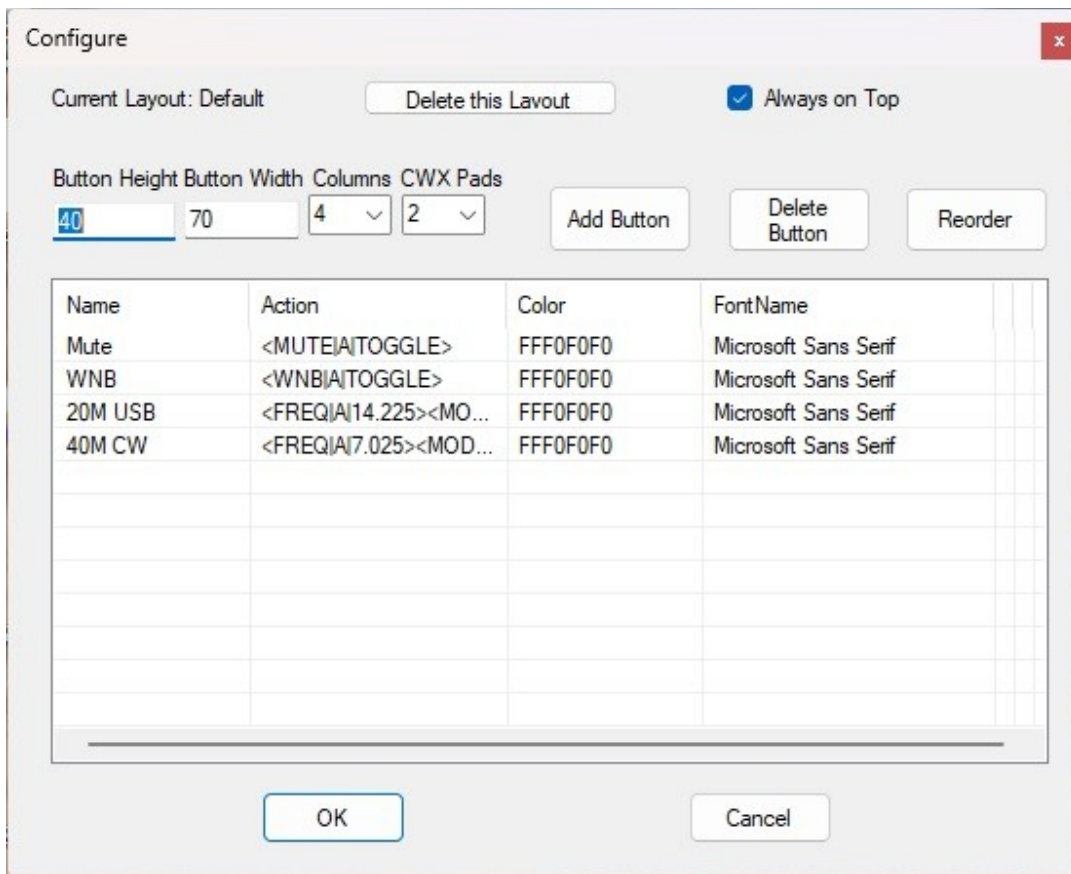
Configuration

Once the program is working you will probably want to add/change some buttons. Start by clicking on “Layout” and then “New”. Pick a name like “Test” to get started. Think about how you operate and start with one or two buttons that will help. Add more once you see how it feels. You can also attach multiple functions to a single button. If you are more aggressive, you can double click on one of the “.reg” files to start with any of the three layouts shown at the top of this document.

Click on the word “Configure” at the top and you will see the configuration screen. We’ll start from the “Default” layout for the examples below (that’s the top left picture above). Note: the name of the layout is shown on the configuration screen. There is also a button to delete the layout.

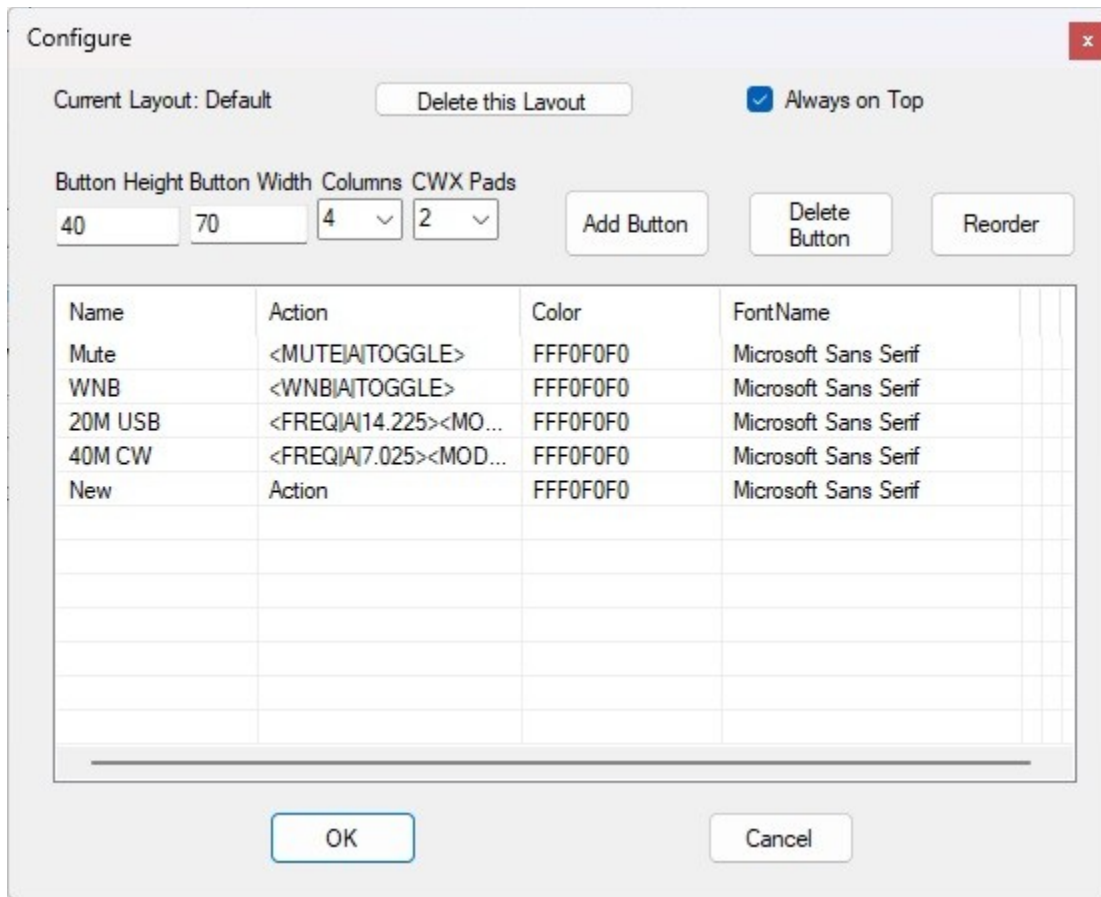
On the first row of text boxes you can select the size of the buttons and the number of columns. Try it, it’s easy to change it back. Adjust the number of columns and button size to help fit the buttons into your overall screen layout. CWX Pads will be discussed later. Change fonts and color to help you recognize buttons quickly. For buttons that are of the ON/OFF/TOGGLE type the color will show when the item is in the ON state; while in the OFF state it will be gray.

Notice that in the screen below you can mark the application as “Always on Top”. This will cause FlexButtons to sit on top of (most) other applications on your screen. You can add a button (at the end of the list) by clicking “Add Button”. The “Add Button” won’t add a function, just a dummy button that will show up as “New”. You can delete a button by clicking on its text/name to highlight it then clicking “Delete Button”. We’ll get to “Reorder” below.



The screen shows, for each button, the button text/name, the encoded function information, the button color and the button font. You can’t change them here – we’ll get to that shortly. You may have to scroll down to see the “New” button as shown below. Any changes you make will be discarded if you click “Cancel”. Clicking “OK” will cause the changes to take effect.

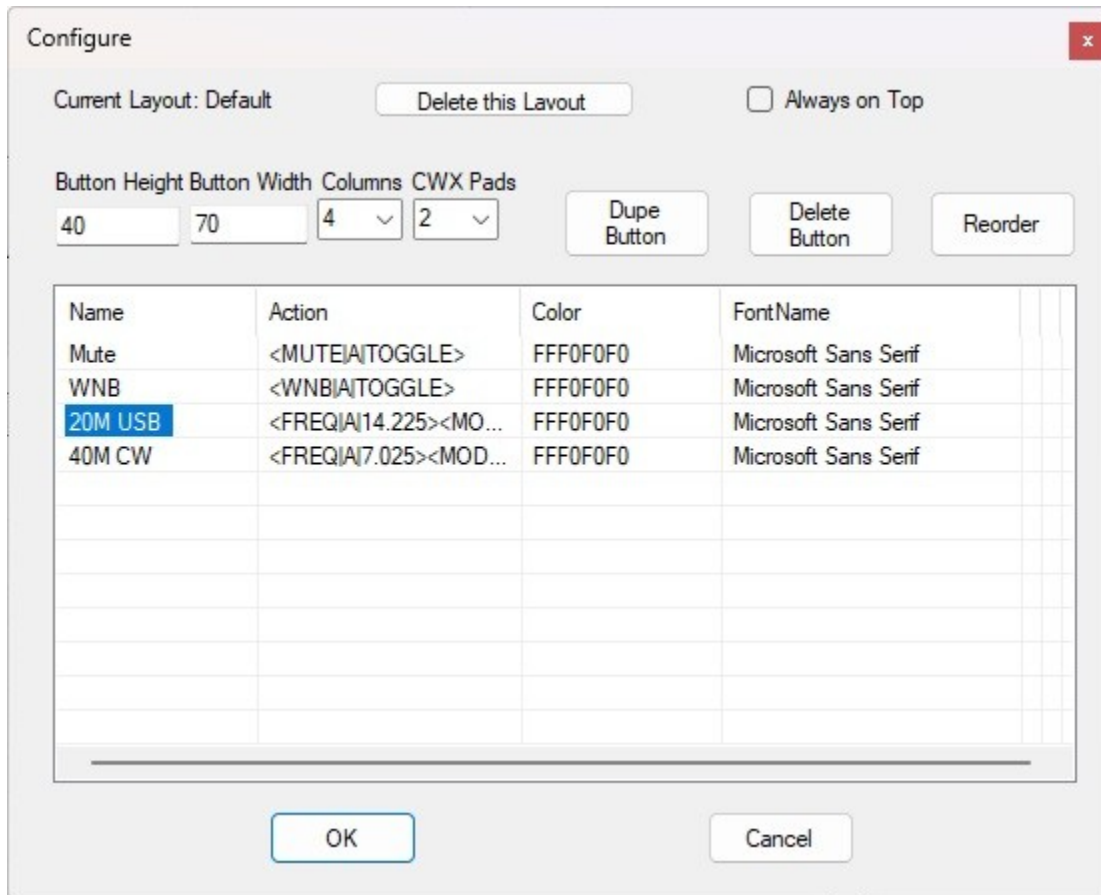
The screen below shows the addition of a “New” button. You may have to scroll down to see it if you have a lot of buttons. You can double click on the name (New) to change the name, color, font and to replace “Action” with an actual function using the Wizard (described below).



CWX Pads are scratch pads that you can put text in will you are operating. The text will be sent through the CWX mechanism when you click the associated “Send” button. This only works for CW.



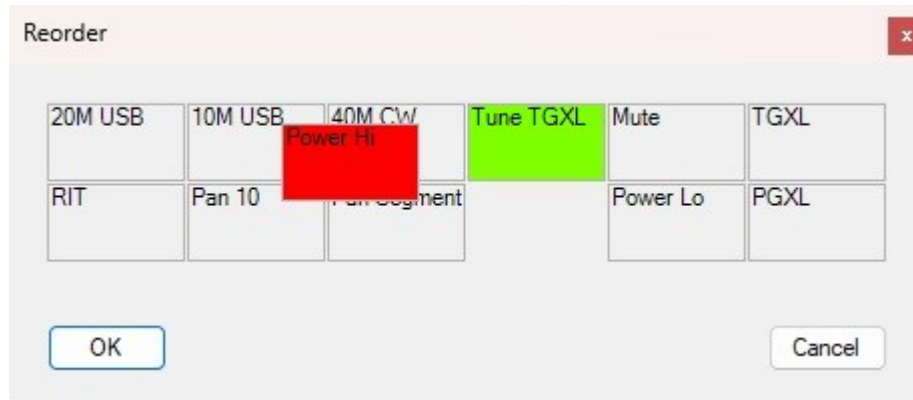
If you first select an existing button by clicking on it the “Add Button” with change to “Dupe Button” and clicking on “Dupe Button” will add a new button with the characteristics of the selected button.



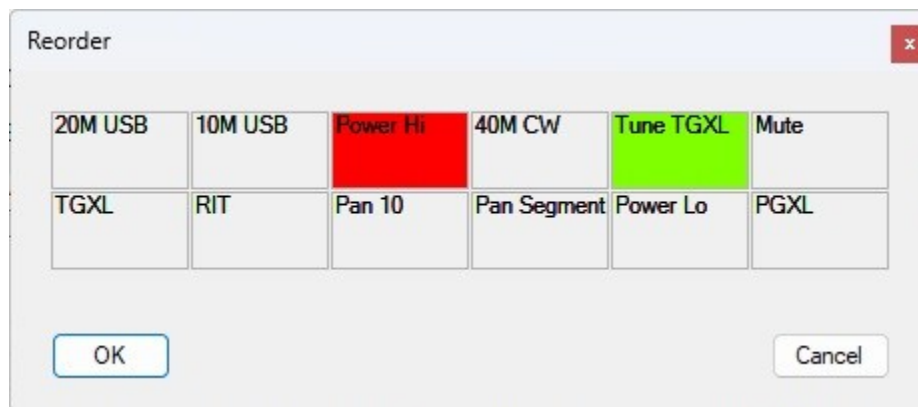
You can also rearrange the order of the buttons by clicking the “Reorder” button. You will see a screen that looks similar to the main screen, and allows you to drag buttons to the location you want them.



Grab the red button with the mouse and drag it up and to the left (don't let go yet). The result should look something like the picture below.



The red button is shown being dragged (the mouse cursor doesn't show in the screen capture). When the mouse button is released (go ahead and release it) the button will be placed at the nearest "reasonable" location and the other buttons adjusted to maintain a clean layout. It takes a little practice to get used to the process.



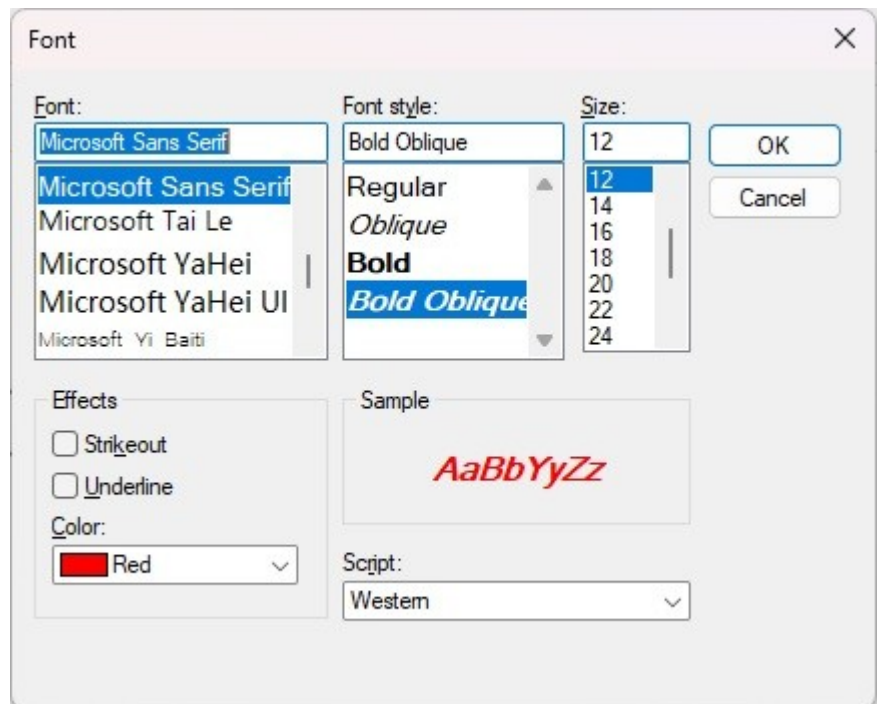
Note: there is an issue with GAP buttons. You can't move one to the bottom of the order. The only way to get one there is to make it the last button you add, then don't move it.

Next we have to discuss how to modify the buttons and their actions. To modify a button you double click on the name of the button you want to modify (including a “New” button). The information about the button will appear in the “Edit” window:



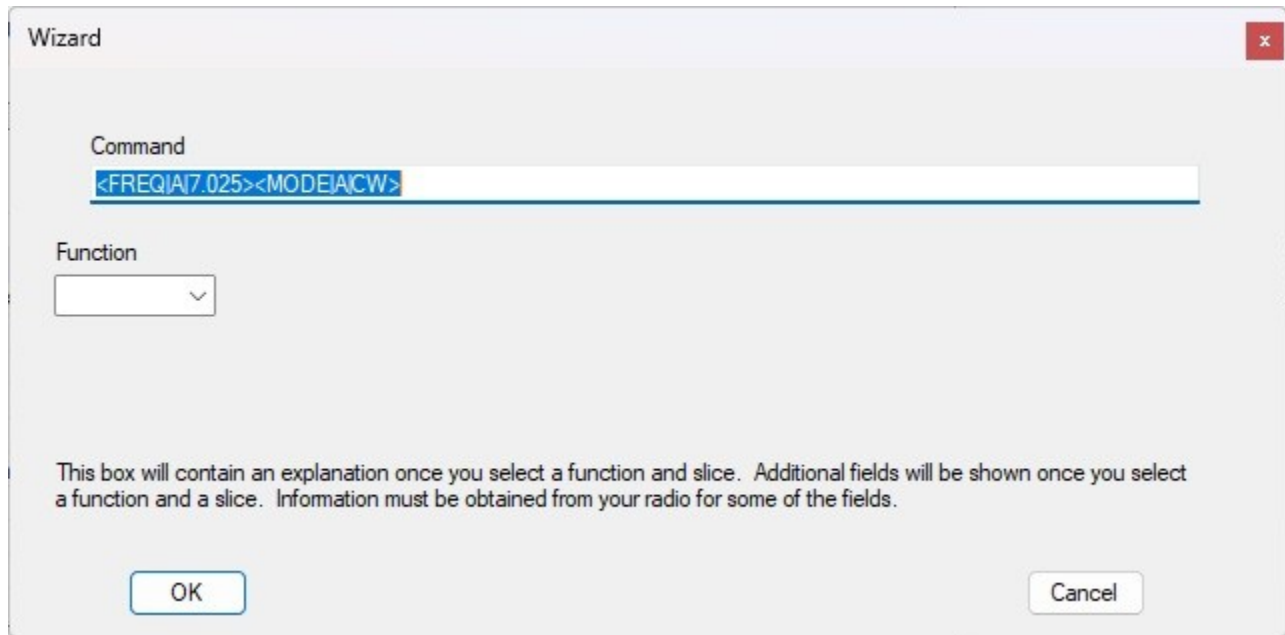
You can modify any of the fields in this window, but at first you should limit that to the Name field. For the other fields there are special tools to make modifying them easier.

If you click the Select button below the Color or Font fields you will see the standard Windows dialogs for picking colors and fonts.



For functions that turn things ON and OFF (or TOGGLE), the selected color will be displayed when the item (i.e. WNB) is in the ON state and the usual gray background will be displayed when the item is in the OFF state.

Under the Action field click the Wizard button. It will help setup one (or more) actions that the button will invoke.



The Wizard dialog box has a title bar with a close button (X). The main area contains a 'Command' text field with the text '<FREQIA7.025><MODEIAICW>' selected. Below it is a 'Function' dropdown menu. At the bottom, there are 'OK' and 'Cancel' buttons. A note at the bottom reads: 'This box will contain an explanation once you select a function and slice. Additional fields will be shown once you select a function and a slice. Information must be obtained from your radio for some of the fields.'

Normally the command field will show the current command. You can either delete that command and make a new command, or add additional commands for the button to invoke (in sequence).

The current command sets the frequency for slice A to 7.020 and the mode to CW. Let's add a second command to set the power level for the radio. Creating a new command requires several steps. First you must select a command and the slice that you want to affect. You can select a specific slice (A, B, C, etc.) or you can select the ACTIVE slice. If you select ACTIVE FlexButtons will send the command to what the radio determines is the active slice. Once you've selected a slice the other relevant fields will show. In this case the "Power Level" field will appear. Note: several commands including VOX, SPOTS, BLANK, GAP, TGXL and PGXL commands don't use a slice so the slice selection box won't be visible.



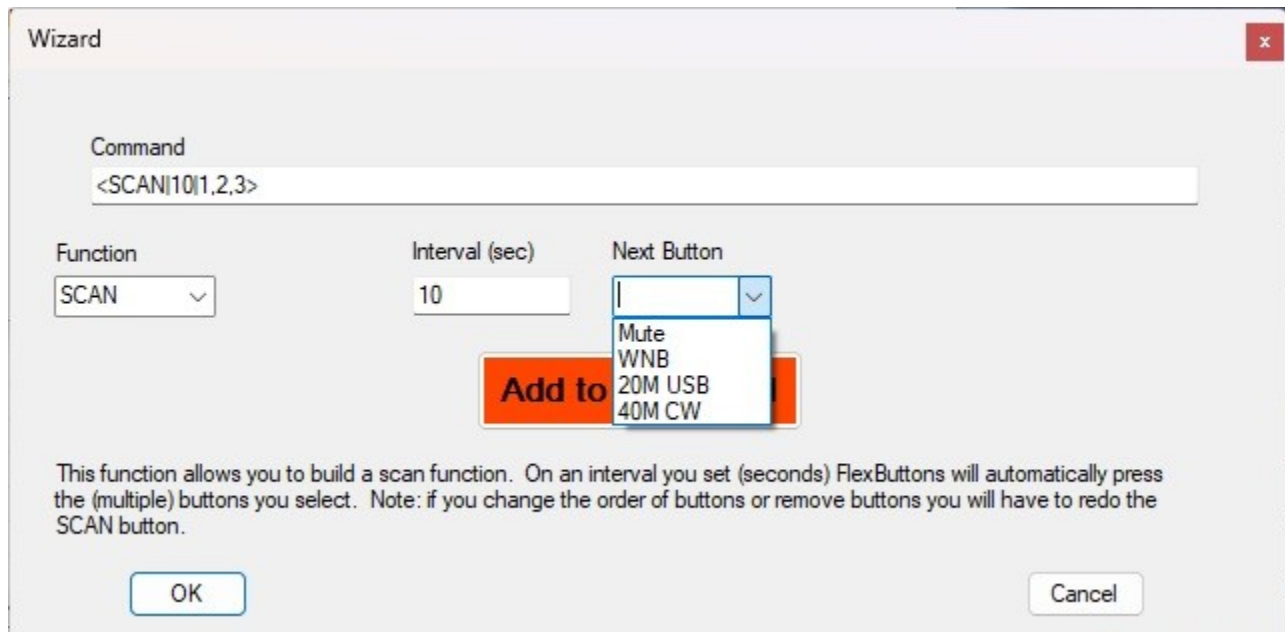
The Wizard dialog box now shows additional fields. The 'Command' field still contains '<FREQIA7.025><MODEIAICW>'. Below it are three fields: 'Function' with a dropdown set to 'PWR', 'Slice' with a dropdown set to 'A', and 'Power Level' with a text input containing '50'. A prominent orange 'Add to Command' button is centered below these fields. At the bottom, there are 'OK' and 'Cancel' buttons. A note at the bottom reads: 'Enter the power setting for the slice 0-100'.

When you click the “Add to Command” button, the new command will be added to the command line and the screen will reset.



An interesting feature is to show the presence of TGXL and PGXL devices and whether they are in active or standby state. You can see them in the menu bar at the top of the window in the examples at the beginning of this document.

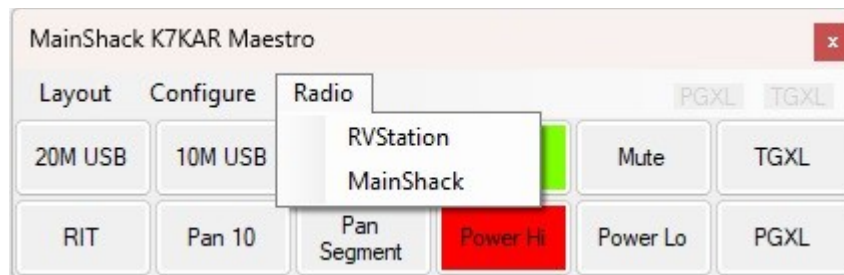
The SCAN command is somewhat special. In the wizard it will ask for a time interval and then allow you to add as many button names as you like (don't go crazy) before clicking on the “Add to Command” button. You should not include other commands on the same button as a SCAN command.



As you can see in the picture above, the SCAN command has a sequence of numbers after the interval. The sequence of numbers is the list of buttons to be sequentially executed when the SCAN is operational. **If you change the order of buttons or remove buttons AFTER creating the SCAN button you will not get what you expect!**

The RUN command requires some special consideration. This allows you to invoke other programs such as a logging program or a spotting program. I added it as it seems a popular feature. It is set up so that you can have a RUN button which functions prior to the radio being present if you only have one command attached to that button. If the radio is present you can have multiple commands attached to the button.

We allow multiple radios. If, during the discovery process, we find more than one radio a third menu selection appears called “Radio”. Clicking on it will allow you to select which radio FlexButtons will connect to. Initially the window title will show as “No Radio”; once it discovers your radio it will show the nickname and callsign.



multiFlex

multiFlex allows multiple programs to access a Flex 6x00 radio and provide services similar to SmartSDR. Currently, a maximum of two programs (stations) is allowed. On receive they can operate independently (band, mode, etc.). On transmit, only one has control. FlexButtons is NOT a program that counts as a multiFlex client, but it interacts with the radio, so it could be associated with either of the two stations. If you are operating in multiFlex mode FlexButtons will add a multiFlex menu. If you click on the multiFlex menu it will show the (two) multiFlex stations. The one that is NOT in control will be “grayed out”. When you select one of the clients, the FlexButtons actions will attempt to relate to that client. In particular, a command that says “set the mode to USB for slice A” will look for a slice A associated with the last station you clicked on and cause that station to switch to USB in slice A. Some functions relate to the station that has control and will go there regardless of which station you’ve selected in FlexButtons. For instance, setting the power level works this way.

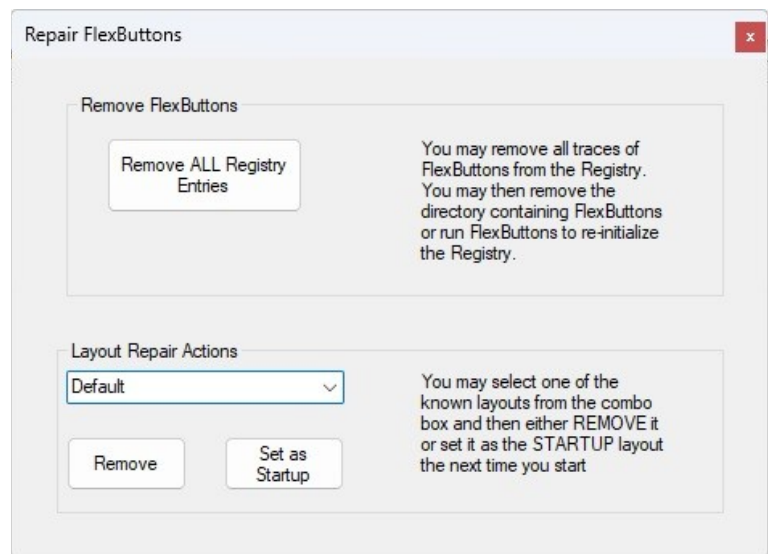
The picture below is a multiFlex environment showing the multiFlex menu. It cannot change the controlling station. That must be done at the station desiring control.



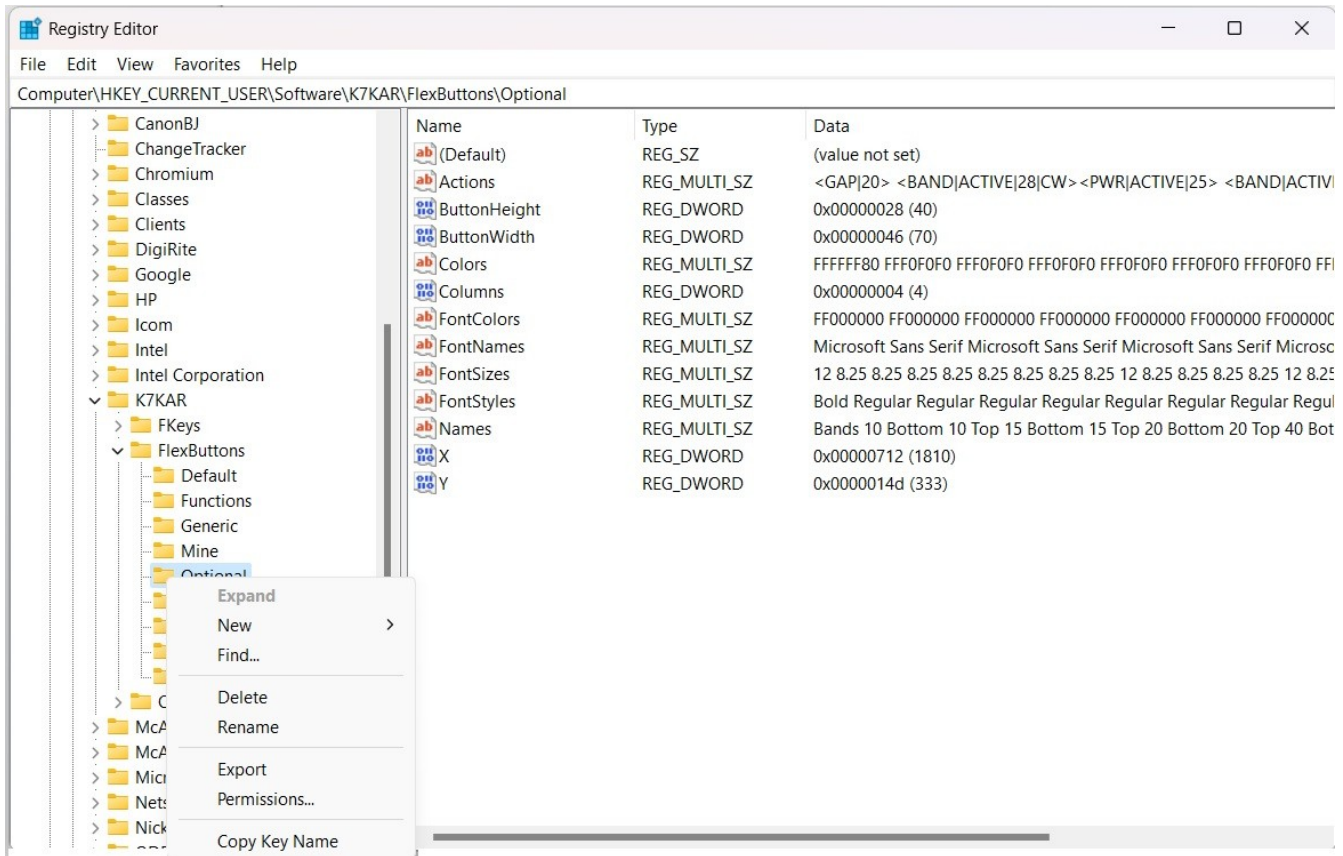
Registry

As much as I try to prevent it, you may end up with a layout that is not viable as your startup. That makes FlexButtons pretty much unusable. You can use Regedit.exe to resolve the issues if you are comfortable with that, but I've added a little utility to help fix the problems. FlexButtonsRepair.exe will allow you to "start over" by removing all trace of FlexButtons from the Registry, or remove a particular layout, or set a different startup layout.

If you want to remove FlexButtons entirely, first use the utility to remove all Registry entries and then simply delete the FlexButtons directory.



The Registry may also be edited using the Regedit.exe tool provided by Microsoft. You MUST be careful with this because you can destroy your Windows system if you get carried away. However, if you stay within the confines of the FlexButtons section you'll be OK. The picture below shows how I export a layout so that it can be installed on a different computer. I expand the view of the Registry until the Optional layout is shown, then right-click on "Optional" to get the menu shown. Then I can select Export on the popup menu. The resulting ".reg" file is similar to the ones provided with the distribution and can be loaded into the Registry by double clicking on it. You can view it with Notepad.exe as it is just a vanilla text file.



Cosmetic Stuff

You can do a little more with configuration to decorate and organize your layout. There are two functions that you can add to “buttons” that will change them into non-clickable separators.

The first is to use the wizard to create a button that has the “BLANK” action. It will leave a blank spacer the size of a button in your layout. You can move it around to even out the layout. You can see an example in the picture below. There is no text, but you can select the background color.

The second is to use the wizard to create a button that has the “GAP” action. It will leave a blank row of the specified height with the text that you specify as the name of the button. As with other buttons you can select the font and the background color. GAP buttons should not be combined with other commands because they require special handling to ensure they fill a whole row.



Final Comments

Remember to start your radios first so that FlexButtons can pick them up when it starts. It can pick up some things while active, but not all.

There is no reason you cannot run multiple copies of FlexButtons with different layouts at the same time, but there is only one *current* layout and that will be the last layout selected, so configuring with multiple copies running is a bad idea.

Here are a couple of layouts for really different operators.



The one on the left is set for someone who checks into several nets and is interested in ARRL activities. Each of these buttons will set the radio to a specific frequency and mode (and possibly invoke the ATU).

The one on the right might be used by a contester. When they're running they'd mostly click the "RIT to 0" button. When they're doing S&P they'd either tune from the bottom of the band to the top and then move back to the bottom with a click (or the reverse). They might want to pan the entire band segment to see where the activity is or where the spots are and then switch back to 10 kHz pan to tune in stations.

My original vision for this program was that I'd have five or six buttons that would do things that were a pain to do on the Maestro. After making it available to others, I found many would start with 20 or so buttons and want to increase. I suspect people will come up with all sorts of configurations that I've never imagined! Tell me about them – I might be able to make it easier to setup or use.

Have fun.! If you have questions or suggestions send an email to me at keith@kregli.com.