

Function keys:

Function keys can be programmed for company-wide use or for personal use. They are a real handy means of trimming long keystroke sequences down to just one. The most-common use is to speed-up common tasks like completing cash, charge, COD invoices on the counter.

To set up function keys for use company-wide:

- Back to the dot and type **FIXTRN** and press enter.

The system should state that it is “Editing function key translation file: LIB:WYSE50.IFX”. This is the file that is automatically loaded at startup.

- You will be asked for a “delimiter” key. This key is used to tell the program when you are done programming and to exit. Use the up-arrow key “↑” since we do not use that key in any of our software.
- Press the function key you wish to program.
- Type **Y** (and then enter) to acknowledge you want to update the function key.
- “Program” the key by entering any and all keystrokes you want executed when the function key is pressed. When you’re done, press the up-arrow key to stop recording.
- You can program another function key by pressing that key or the up-arrow key again to exit.

To set up function keys for personal use:

- Back to the dot and type **LIB** and then enter. This should take you to DSK0:[7,0].
- Type **FIXTRN <filename>.pfk** and press enter. Example: **FIXTRN SAMPLE.PFK** for personal function keys for user/terminal name of “SAMPLE”.

If found, the system will state that it is “Editing function key translation file: SAMPLE.IFX”.

If not found, the system will state that the file does not exist and ask if you want to create it. Answer by typing **Y** (and then enter).

- You will be asked for a “delimiter” key. This key is used to tell the program when you are done programming and to exit. Use the up-arrow key “↑” since we do not use that key in any of our software.
- Press the function key you wish to program.
- Type **Y** (and then enter) to acknowledge you want to update that function key.
- “Program” the key by entering any and all keystrokes you want executed when the function key is pressed. When done recording, press the up-arrow key.
- You can program another function key by pressing that key or the up-arrow key again to exit.

In order for personal function keys to work, you must load them so they “overwrite” the function keys that have been set up company-wide. Some commands have been created to help shorten the keystrokes necessary. If they don’t appear to work, contact Datatron Support for assistance.

- | | |
|--------------------------------|---|
| EK <filename.pfk> | Edit and create personal function keys “filename.pfk”.
Note: this command assumes LIB: as the location of the .pfk file. |
| LK <filename.pfk> | Load personal function keys “filename.pfk”. |
| DK <filename.pfk> | Delete personal function keys “filename.pfk” from memory to revert back to using the company-wide function keys. |

If a certain user has a personal function key file which is used everyday and the company-wide ones are never used, so long as the user has a unique login (i.e. not “user”), the system can be set up so the custom set is loaded when they login to the system so the user doesn’t have to load the personal function keys every day for each “flip-screen” they use. This can be really helpful to a Controller, to Accounts Payable, etc.! Contact Datatron Support for help in setting that up.

File locks:

File locks can sometimes be a real pain, particularly to a system administrator. It’s inevitable that once in a while you will get someone that will get exclusive use of a file then maybe leave their desk or terminal without backing-out of the program to clear the file lock.

Common occurrences include part numbers pulled-up in Inventory Maintenance and customers pulled-up in Customer Maintenance which can both cause warehouse background processors to stall and invoicing screens to hang.

The program “A-Shell QFLOCK.SYS/JOBTBL.SYS Utility Program” can be used to help identify what user or terminal has a certain file lock so that you can then get them to relinquish the lock so others may continue what they were doing.

Identifying who has an “exclusive” file lock:

- Back to the “dot” prompt (i.e. “LNX .” or “IBM .”).
- Type **QU** and then enter.
A list of options will be displayed. The “list of all FLOCK.SBR locks” is what you will use most often.
- Type **FL** and then enter. This is short for **FLIST** which will also work.
You will be asked to “Enter file to output list or RETURN for screen:”
- You will want to output to a file so you can search it for a particular string.

Here, we recommend using either “1.1” or “a.a” to be quick and simple. These are filenames we never use in our programming so in the event that you forget to erase these files and you run across them in the future, you’ll know that they are most-likely “temporary” files that can be erased.

- You will next be asked “Display output list on terminal?” Type **N** (and then enter).
- Type **EX** at the “main menu” (and then enter). This is short for **EXIT** which will also work. You should now be back at the “dot” prompt.

- VUE the file you just created. Search for the word “exclusive” and note any terminal(s) that have exclusive use of the file in question. **CTRL-X** (in Vue) to be sure that you note all terminals with exclusive locks (don’t assume the first one that comes up is the one you are looking for!). Hit **ESCAPE** and enter **Q** to exit VUE.

- Erase the output file if you don’t think you’ll need it any longer.

Note: Depending on the version of Ashell you are running, you may also see a “pending” which will indicate a terminal which is waiting for an exclusive file lock to be granted. This may help you determine which ones needs to be freed.

- Once you know the terminal with the exclusive file lock which needs to be freed, you can find the user running the program and ask them to back-out or, if needed, kill their job using the “KILL” command in the OPR: account.