

Inficon UL1000 Fab Helium Leak Detector Software Manual

v. 1.0

Alexey Drobizhev
UC Berkeley Bolometers R&D

January 29, 2015

1 Introduction

This is a basic instruction manual for operating the RS232 interface program for the Inficon UL100 Fab Helium Leak Detector. As of this writing, this is a not-standalone LabView VI that runs on Mott, and allows the user to start and stop the device, make a real-time plot of leak rate versus time, and save a live-updating ASCII file of this information.

This manual also explains basic plotting of the leak checker data using pre-written ROOT scripts.

This manual does not perform a detailed treatment of leak detector interfacing, leak detector operation, or standard cryostat leak checking procedures. For those, see the [Interface Description](#), the [Technical Handbook](#), and the Berkeley Bolometer R&D Cryostat Manual ([Interactive in browser](#) or [PDF](#)), respectively.

2 Operation

1. Begin by positioning the leak checker such that your pumping line can reach the desired pumping port, the power cord can reach the mains outlet, and the RS232 cable can reach the computer. Typically, we

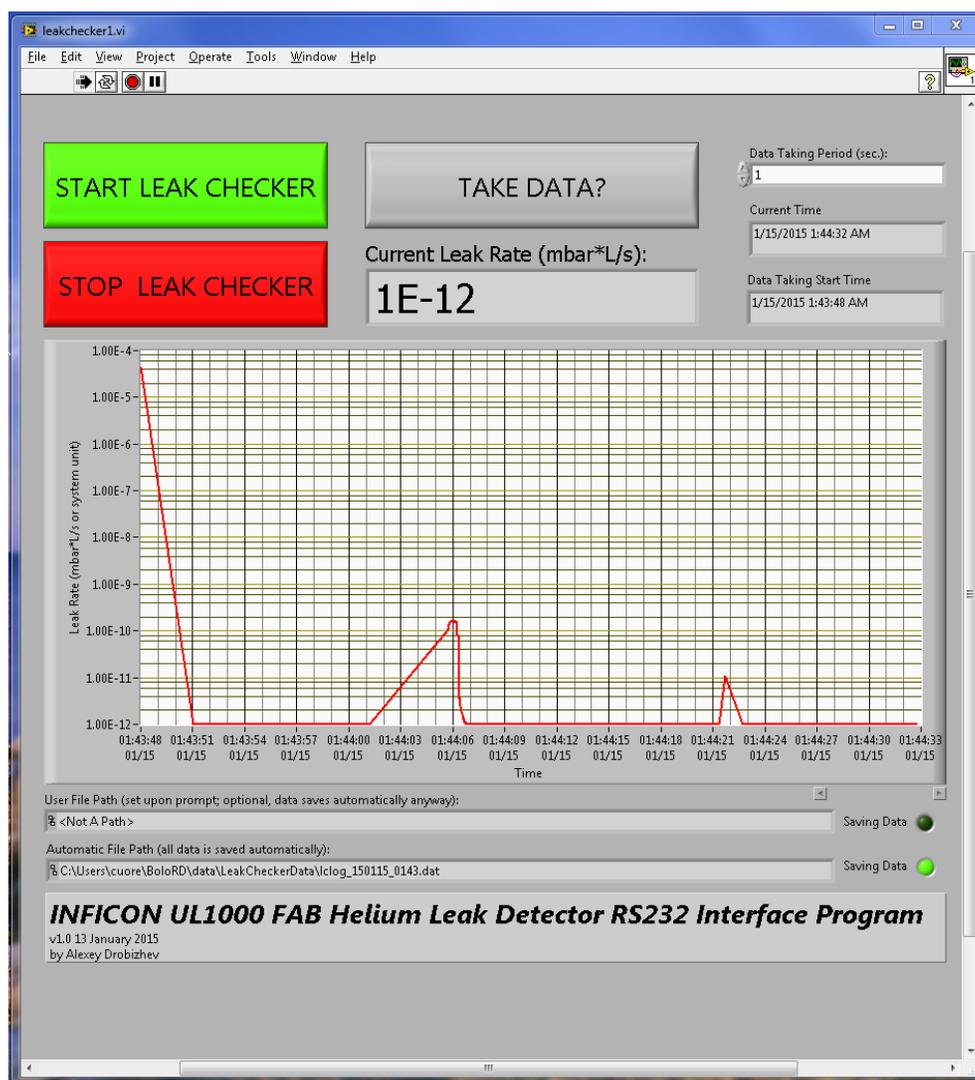


Figure 1: The graphical interface for the Inficon UL1000 Fab helium leak detector RS232 interface program (leakchecker1.vi).

are leak checking the OVC, in which case the leak checker is placed adjacent to the two cryostat equipment racks/cabinets, with the front protruding into the "clean" enclosure. In this case, 2 vacuum lines could be routed through the cryostat supports to the pump cart auxiliary valve, the power cable can be plugged into the outlet on the western wall of the cryostat alcove, and the RS232 cable can be run to Mott behind the gray equipment rack. As of this writing, said cable (multi-colored twisted pair ribbon terminated on both sides with DB9's, with channels 2 and 3 NOT flipped) is kept plugged into Mott, with the leak checker end hanging from the side of the rack for easy access.

2. Plug the leak checker into the mains outlet and into Mott, and turn it on with the black power switch while keeping it capped at the inlet.
3. You expect (as of this writing) two error messages to pop up on the leak checker display, press "OK" for both (we are aware of them, they are not yet a problem). If you see more errors, stop leak checking, look into the error, and notify the others. Afterwards, the leak checker will load calibrations and configurations and spin up its internal turbomolecular pump.
4. Once the leak checker is ready to go (will display "STANDBY" on its screen), hold down the "STOP" button for 3 seconds to vent (will make small hissing noise and will display "VENTED").
5. Connect the leak checker to the desired volume (most commonly the OVC via the pump cart's auxiliary valve; could also be something else, such as the OVC directly while at base or one of the other possible DU leak checking configurations).
6. If the leak detector control program on Mott's Windows 7 partition is not already running, open it by double clicking on the leakchecker1.vi icon on the desktop.
7. Start running the leak detector control program by pushing the white arrow button near the top left of the LabView window.
8. When ready, start the leak checker by pushing the green "START LEAK CHECKER" button in the software interface. The leak checker

should start immediately. If it doesn't, and/or you get an error from LabView:

- Make sure the interface cable is plugged in on both ends.
- If it is, make sure that BOTH RS232 and local control are enabled on the leak checker by pushing the "MENU" button below the screen, then selecting "Settings," followed by "Interfaces" and selecting the option to enable BOTH local and RS232 protocol interfaces.

If only local control is enabled, the leak checker will not respond to commands from the computer or send it data.

Do not select the RS232-only option. If only RS232 control is enabled, the computer program will be able to control the leak detector and read out data. However, it is very important that you don't choose this setting. The computer program is only able to start and stop the device and take data. It is intentionally unable to vent leak checker, to avoid damaging cryostat elements by accidentally venting at an inappropriate time. Venting must be performed manually, directly on the cryostat control panel. Venting is a necessary step of the leak checking process, turning it off and/or disconnecting it without venting is highly undesirable and can damage the leak checker. Thus, when doing any leak checking work, it is imperative to maintain the local interface functional and unlocked.

Of course, the fact that to even get this far in the procedure you need to perform several tasks with direct control is a safety check. Nonetheless, I include this very verbose warning to prevent you from choosing the incorrect RS232-only setting.

Additionally, the program is unable to perform more advanced tasks like calibration either, since this would be of limited utility. These tasks may be necessary to you, and are an additional reason to keep the direct interface active.

WORST CASE SCENARIO: if you've enabled RS232-only control

on accident, and your control panel is now disabled, the [Interface Description](#) provides simple RS232 commands that you can use to perform virtually any manipulation with the leak detector. Use these via Terminal or LabView's VISA device to stop and vent the leak checker (after isolating it from the cryostat by closing the auxiliary valve or equivalent), and then re-enable direct control. In principle, you can fairly easily run a whole leak-checking process using only direct text commands. I have done this. However, I do not recommend it simply because direct control tends to be faster, which can be useful in case of emergency.

- Try restarting the software.
 - If none of these work, you should check the RS232 cable to make sure all necessary connections are not broken or shorted (see the [Interface Description](#) for which channels of the cable are used and for what—not all of them are) and let Alexey know.
9. You can also start the leak checker using the “START” button on the device itself. This is perfectly fine. However, for the computer program sequence to run correctly, the green “START LEAK CHECKER” button in the program still needs to be pressed. When you are ready to use the program, do so. Nothing will happen to the device, but it will properly enable the gray “TAKE DATA?” and green “STOP LEAK CHECKER” buttons.
 10. When you're ready to take data, set the data taking period with the text control in the upper-right corner of the program window. Setting it to 0sec. will take data at the maximum possible rate. I am not sure what that is exactly, but it is several data points per second. This parameter can be changed at any time, if you want better time resolution at some points, and to keep file size down at others.
 11. Push the gray “TAKE DATA?” button to enable data taking. It will turn a darker shade of gray when engaged.
The program will automatically save data to a file at `C:\Users\cuore\BoloRD\data\LeakCheckerData\lclog_YYMMDD_HHMM.dat`.
It will also prompt you for a directory and file name. You can provide these if you want a duplicate file, or if you want to append an existing

file (it will append automatically if an existing file name is provided) here. If you push “Cancel,” no duplicate file will be created, but all data will still be saved in the default location.

When data-taking commences, you will see the leak rate value display in the window in the upper-middle, and the leak rate vs. time plot start updating real-time (if you don’t see it, check if autoscale is enabled, and what the range and domain are if not).

Also, below the plot, there are two text indicators and two green boolean indicators. When data is saving, the directory and file name will be displayed, and the green indicator next to it will light up. This should always happen for the default data file, and for the custom one if it was created by the user.

If any of the above do not happen, tell Alexey.

NOTE: The leak rate units used in the program for the indicator, the plot, and in the files are the same units as on the leak detector’s display. Currently, these are set to $\frac{\text{mbar} \times \text{L}}{\text{s}}$, which is a commonly-used standard unit and is unlikely to be changed in the future. Keep in mind, however, that if units on the leak detector are changed, the software will display the reading in the new units, while still stating that the readout is in “mbar*L/s.”

12. Once you have the program set up, proceed with leak checking procedure.
13. At any point in time, you can push the “TAKE DATA?” button and pause the acquisition. You can push it again at any time, and data taking will resume, appending the previously-created file(s).
As of this writing, disabling acquisition will also disable the real-time plot. The ability to draw the plot without data taking might be added later, if it is deemed sufficiently useful. For quick checks without taking data, there is a plot on the leak detector’s display itself.
14. If you have finished leak checking, or want to stop the leak checker temporarily, push the red “STOP LEAK CHECKER” button. Make

sure that the leak checker is isolated when you do this, and not, say, connected to an evacuated OVC.

15. You can restart at any time by pushing “START LEAK CHECKER” and following the same procedure. If you do, it will create a new automatic data file and prompt you again for an optional duplicate. If you want to append your old file, you can select it as the duplicate.
16. If you are finished leak checking and the leak checker is isolated and stopped, you can stop the program (red stop sign button near the upper-left of the LabView window) at any time. Vent the leak checker, disconnect the pumping lines, close the leak checker’s inlet, turn it off and unplug it. Keep in mind that until you hear the leak checker’s internal turbo pump spin down, you should not wheel it to its usual parking spot. Wait a few minutes before doing so.

You can also stop the leak checker using the leak checker’s local interface. The program will keep taking down data, and if you later want to re-start the leak checker and take more data, you have to either start the leak checker using the local controls (data acquisition never stops), or push “TAKE DATA?” followed by “STOP LEAK CHECKER” (or just “STOP LEAK CHECKER”), then restart the leak checker with the “START LEAK CHECKER” and data acquisition with the “TAKE DATA?” buttons.

3 Leak Checker Data

3.1 Leak Checker Files

The leak detector programs saves data to ascii files. By default, these are found at

`C:\Users\cuore \BoloRD\data\LeakCheckerData\lclog_YYMMDD_HHMM.dat` on the Windows partition of Mott. You can choose to save your own copy of the file with a different name and directory by providing the program with the appropriate information when prompted (or don't, and hit "Cancel" for just the default file).

The files contain three tab-separated columns: 1) seconds since the start of data taking (6 digits after decimal), 2) leak rate (written by LabView as 6 digits, but actually just 3 digits from the leak detector), and 3) Unix time stamp (seconds elapsed since 01/01/1970 00:00 UTC; 16 total digits precision without scientific notation).

The slightly awkward formatting of time:leak rate:time, along with not using scientific notation for the unix time stamp, comes from the problem that the standard LabView plain text file writer limits all numbers to 6 digits precision, and it is not possible to change this. This is not acceptable, since we would only get 1000 second time resolution in saved data. There are several work-arounds, none of which are particularly pleasant. The chosen workaround turns the unix timestamp (itself calculated from LabView time with the SubVi unixtime.vi, also on the desktop) into a string, and adds it as a "comment" at the end of the line—the length of comments is not restricted. If you have a better way of writing a sufficiently precise unix timestamp and would like to update the program, feel free.

3.1.1 DOS/Windows Carriage Returns

Since the leak detector program runs on the Microsoft Windows 7 operating system, the above-describe ascii files have DOS/Windows-standard carriage returns, which differ from unix-standard carriage returns used on Linux and Mac OS X. At the end of each line, Windows places both an end-line character `\n` and a carriage return character `\r`, whereas normal ascii text/unix standard has only an end-line character.

To be able to work with the leak checker data files on a unix system, the windows carriage returns must be removed. There are several ways to do

this.

- In unix terminal, use the `tr` (translate) command, which can translate characters into others or simply delete them. Specifically for this case, run `tr -d '\r' < inputfile > outputfile`.
- I personally like the [Dos2Unix/Unix2Dos](#) file formate converter, which is also available through macports.
- Others, such as a simple python script.

3.2 Processing, Plotting and Analysis with Standard Scripts

I have prepared four ROOT scripts for performing basic manipulations with the leak checker data files:

- `makeLCrootfile.C` creates a ROOT data file from the raw ascii data file. Execute the script by starting ROOT, loading the script with `.L makeLCrootfile.C`, and then running `makeLCrootfile("filename.dat")`. This will create `filename.root`, which contains one tree/ntuple of doubles called `lclogs`, which in turn has three branches called `t_rel` (seconds since starting data acquisition), `leakrate` (leak rate in leak checker's set units—almost always $\frac{mbar \times L}{s}$), and `t_abs`, the unix timestamp.
- `makeLCroot_draw.C` creates the same ROOT file as described above, but additionally makes a plot of leak rate vs. time. The script should be executed the same way, with the input syntax being `makeLCroot_draw("filename.dat", time option string, axis option string)`. If the time option string is set to "rel", then the x-axis of the plot will show seconds since the beginning of data taking; if it is set to "unix", then a unix timestamp will be used; a blank string or any other input will make the x-axis be in human-readable time and date format. If the axis option string is set to "log", then the y-axis will have a logarithmic scale; otherwise it is linear.
- `drawLCroot.C` makes a leak rate vs. time plot from an already existing root file. It is executed the same way, with the input syntax being

`drawLCroot("filename.root",cuts string, time option string, axis option string)`. The time and y-axis options are the same as in `makeLCroot_draw.C`, while the cuts string lets you make cuts on the data to be plotted, using the same syntax as ROOT's standard `Draw` command.

- `drawLCdat.C` makes a leak rate vs. time plot directly from an ascii file (with DOS carriage returns already removed), without making a root file. Syntax and options are the same as `makeLCroot_draw.C`, it just doesn't save the ntuple to a file when it's done.

As of this writing, these scripts can be found on Moller at `/home/lyosha/leakchecker`, and are not standalone programs. Making them such is on the agenda (see below).

4 Plans For Future Software Improvement (feel free to contribute)

- Make `leakchecker1.vi` a standalone program, for robustness and so that only the LabView runtime environment (as opposed to LabView itself) would be necessary for operation.
- Possibly move the software to Cabibbo, along with the cobalt-60 thermometer program, so that the LakeShore AC370 bridge software on the Linux partition of Mott could be run in parallel. This would involve first making the program standalone, and possibly re-writing in LabView 11 (in which the Magnicon SQUID software is written).
- Alternatively: get the program, or one with equivalent functionality, working on Mott's Linux partition.
- Compile the file-processing root scripts as standalone programs.
- Setup automatic upload of leak checker raw data files to a directory on Moller, possibly along with automatic removal of Windows carriage returns and running of `makeLCrootfile` to create root files (analogous to the setup with the Oxford fridge software or the cobalt-60 thermometer software).