|  |  |
| --- | --- |
| Mercury LabVIEW library Release Notes |  |
| Version 2.0 • | Farbverlauf_mit_icon.png |
|  | |
| MercuryiPS power supply.jpg | |
| **Mercury Support**  Oxford Instruments NanoScience  tel: +44 (0)1865 393200  fax: +44 (0)1865 393333  email: [helpdesk.nanoscience@oxinst.com](mailto:helpdesk.nanoscience@oxinst.com)  <http://www.mymercurysupport.com/> | |
| OI.png | | |
| Orange_baseline_dots copy.jpg | | |

# Mercury LabVIEW library Version 2.0

**General bug fixes and features added**

1. Added library functions for Lambda fridge control.
2. Added many more examples including iTC and iPS projects.
3. Added configurable initialisation VI’s and in-loop (IL) monitor VI’s for iTC and iPS.
4. Moved Query control from private to public to facilitate better VI structures.
5. Various bug fixes and additional functions.
6. Use with Mercury firmware versions 2.x.x.x.

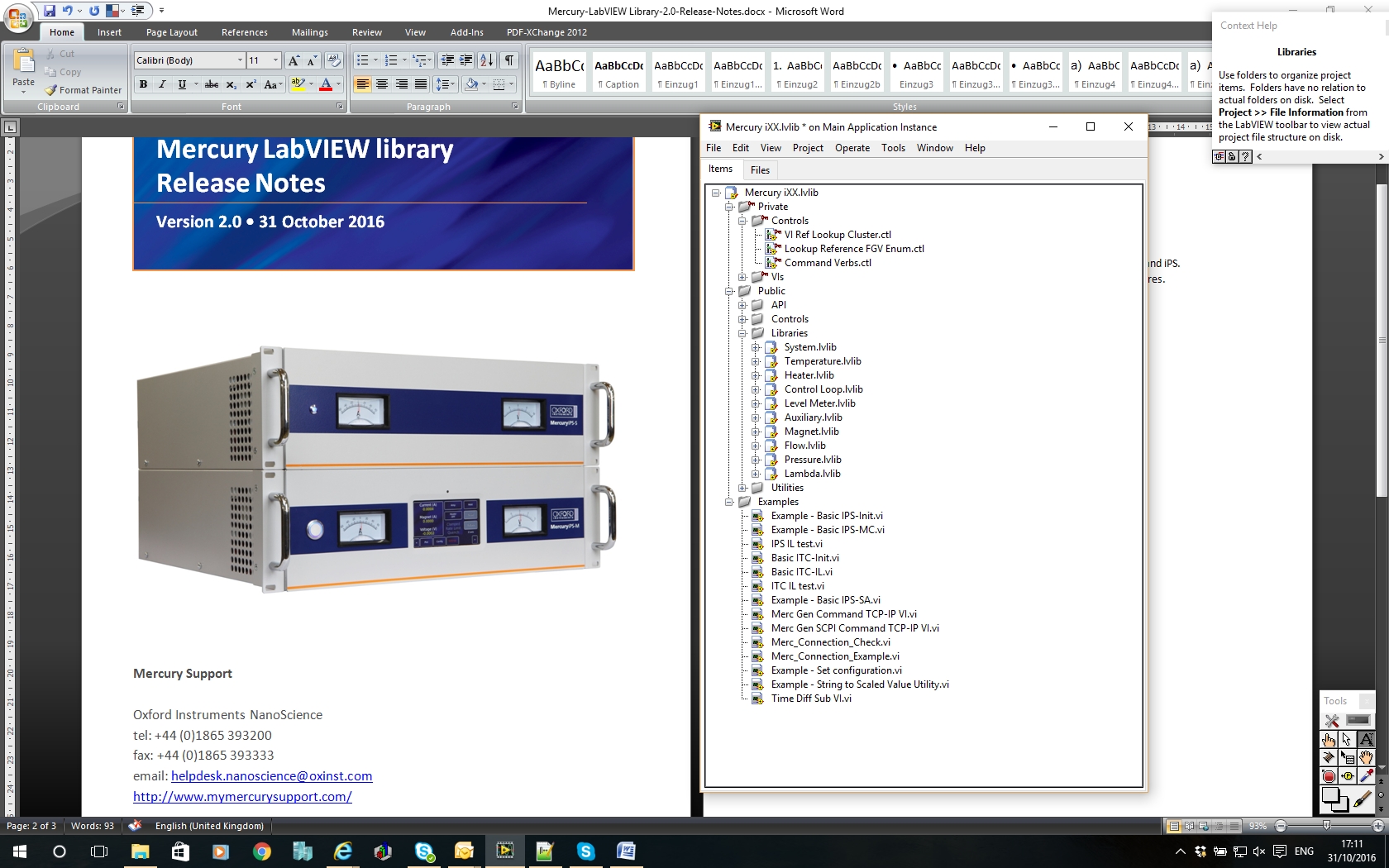


Figure 1. Mercury LabVIEW function library

**Notes**

The version 2.0 function library can be used with LabVIEW 2013 version 13.0f2 or later and is suitable for 32-bit or 64-bit machines. On a 64-bit machine the zip archive should be extracted to...

“C:\Program Files (x86)\National Instruments\LabVIEW 20xx\instr.lib” where “xx” will be the last two digits of the year of issue of the LabVIEW version you are using, e.g for LabVIEW 2013....

C:\Program Files (x86)\National Instruments\LabVIEW 2013\instr.lib.

This will create a “Mercury iXX” sub-directory which contains the function library (Figure 2).

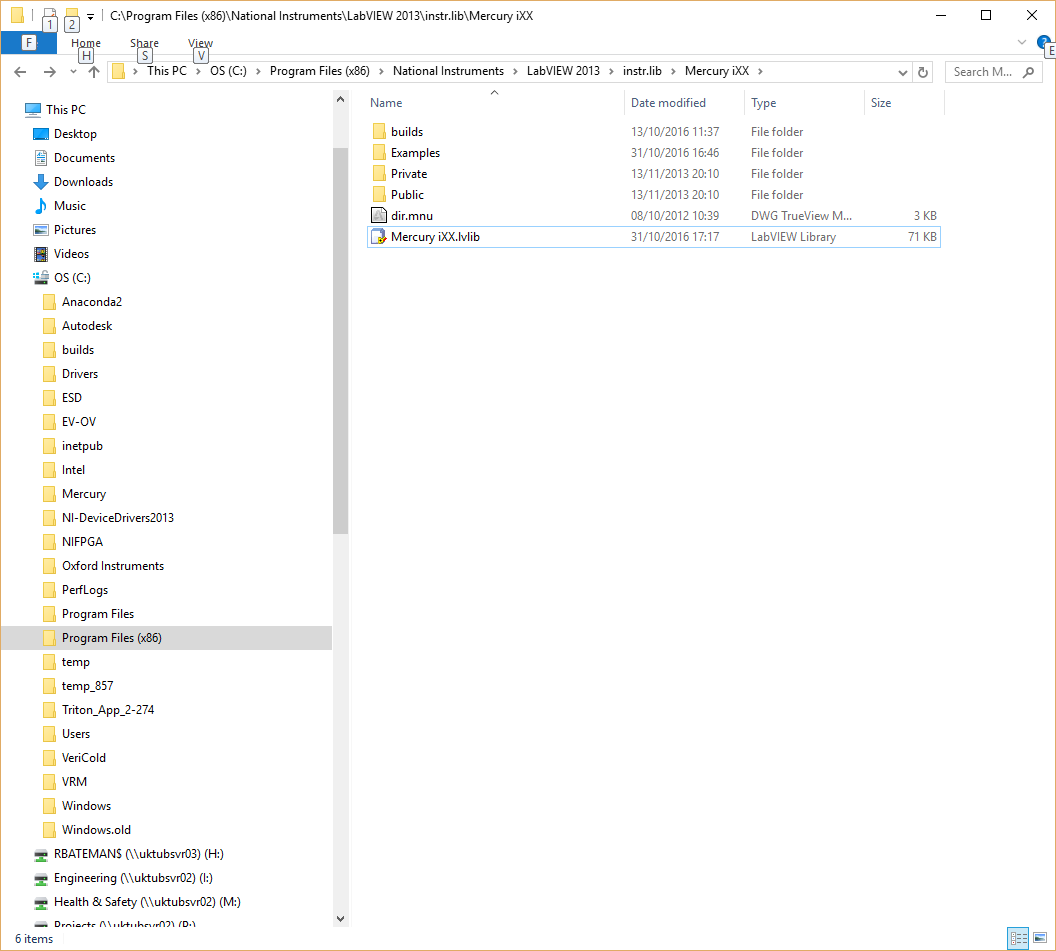


Figure 2. Typical directory structure fo a 64-bit pc running LV2013 (the “builds” directory unless executable applications have been built from user projects).

Once the library is installed in this directory the functions will be available via the instrument library pallets ().

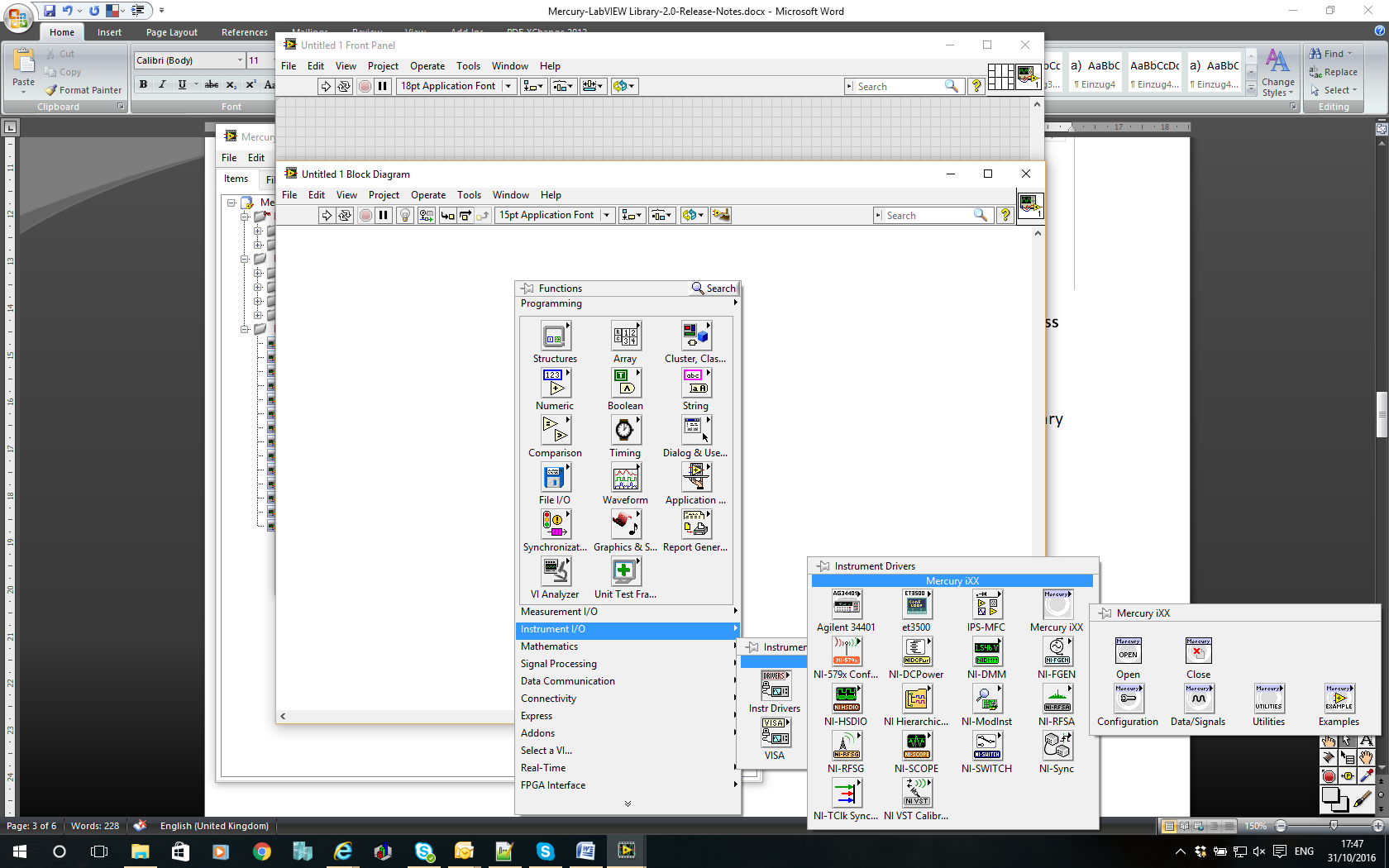


Figure 3. LabVIEW IDE showing instrument sub-pallet for the Mercury functions.

|  |  |
| --- | --- |
|  |  |

Figure 4. Mercury functions pallet and “Examples” sub-pallet.

From the Mercury functions sub-pallet the new set of user example VI’s will appear in the Examples sub-pallet. Alternatively, the example projects “\*.lvproj” can be opened from the LabVIEW File‑>Open menu or from Windows Explorer.

One example is the “Basic iPS-SA”. This is a VI which will run “stand alone” (SA) and can be used to monitor and control various configurations of Mercury iPS units. It is only intended for single axis configurations (i.e. NOT Vector Rotate) but will auto detect and self-configure for multiple units in parallel, series or matrix configurations. The basic front panel is shown below (Figure 5). This shows the optional temperature and cryogen level indicators which will appear if the connected unit has these device installed. The ammeter and voltmeter ranges will also adapt to the configuration of the connected unit.

This example supports connection via TCP/IP, USB, RS232 or GPIB which can be configured via the “Connection”. OINS recommends TCP/IP connection for the best connection performance. This example also includes basic plotting and logging to file of the iPS output current, output voltage, magnet current and magnet temperature.

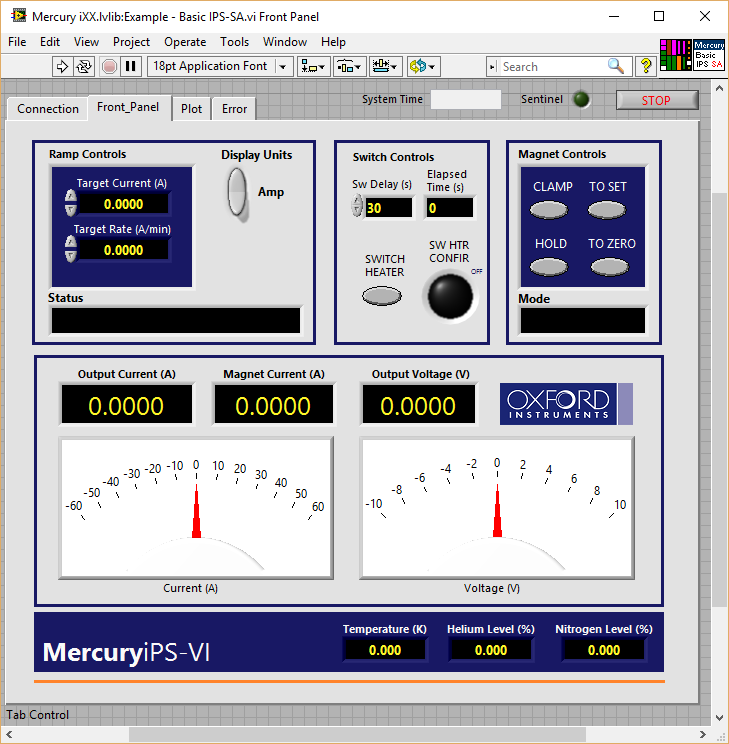


Figure 5. Front panel of the “Basic iPS-SA” VI showing temperature and cryogen level indicators.

The SA version of the iPS VI is useful to test run the the magnet but cannot be used in higher level applications as it runs its own control loop. For this reason the “Basic iPS‑INIT” and “Basic iPS‑IL” (or MC) functions have been created. These example VI’s are ready to use in higher level applications. The “Basic iPS‑INIT” VI establishes a connection to the instrument and creates a reference for each installed device it detects. The “Basic iPS‑IL” (or MC) VI can then be used in a higher level application control loop to read and monitor the iPS instrument. An example “IPS IL test” VI is included to show how these VI’s should be used. The block diagram of “IPS IL test” is shown below (Figure 6).

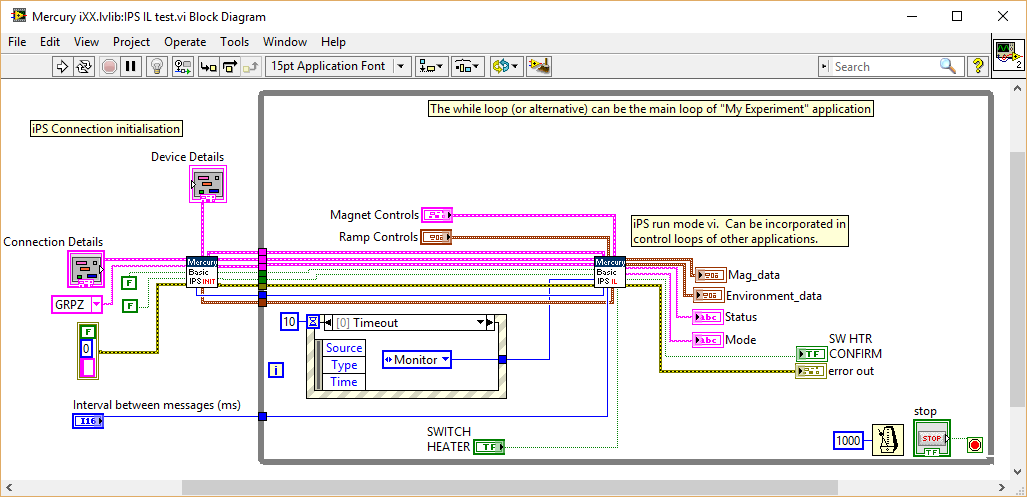


Figure . Block diagram of “iPS IL test” VI showing how IPS-INIT and IPS-IL (or MC) should be used.

There is a similar example for the Mercury iTC with SA, INIT and IL examples. However, the iTC example only captures basic configurations. The iTC is typically much more configurable than the iPS. To create single VI’s which capture all the possible options means the VI’s become very complex and difficult to decipher. Therefore only basic configurations are include as clear examples. The “iTC‑INIT2” VI begins to show how it is possible to create VI’s which will adapt to all configurations and this will be developed further in later releases.

[www.mymercurysupport.com/documents](http://www.mymercurysupport.com/documents) .