**LabVIEW 101 – Weeks 4**

Before class:

1. Download updates from <https://github.com/rizett/LabVIEW-101>

Required equipment:

Computer with LV installed

Topics:

Saving data exercise – different ways to save

Property nodes

Cleaning up diagrams

Elapsed time

Exercises

To start: If you haven’t already, save last week’s exercise 4 (labVIEW101\_week 2\_ exercise 4) into a library. Duplicate the hierarchy to the new location, and confirm that exercise3\_subVI and Julian Day subVI were saved to the library as well. Give the library a useful/descriptive name.

**1. Saving data exercises**

* + See GitHub / examples and functions / output examples
  + Open /examples and functions / examples/week3\_path-save.VI
    - Write to Measurement file:
      * Setup options
      * Link filename / path
      * Data columns (merge signals with sig. manip.)
      * Wire-in / out options

**2. Property nodes & Invoke nodes**

* Property nodes can be used to control the appearance/behavior of an object/element in the VI
  + E.g. use to modify display properties: control plot scales, plot colours,
  + Add them by right clicking on the element of interest > Create > Property Node > *select the property that you wish to control.*
    - To CHANGE the property, set to write
    - To READ the property, set to read
* Open labview101\_week2\_exercise4.vi from the LabVIEW library
  + Add 2 property nodes to change the colour of the plot background and the line colour on the plot
    - Control the background colour using a COLOUR CONSTANT on the block diagram (BD > Graphics & Sound > Picture functions > Color box constant)
    - Control the plot line colour using a COLOUR CONTROL on the front panel (FP > numeric > Framed color box)
    - NOTE the placement of the property nodes – what happens if the node is placed outside of the while loop vs inside?
    - Save As labview101\_week4\_exercise0
    - Now, imagine you have 2 lines plotted on the same figure …
      * Add a second plot line to the same figure (i.e. modify the plot to have 2 x-inputs and 2 y-inputs)
      * Add another front panel control to control the plot colour.
      * NOTE, now we need to add an “active plot” property to control the plot 0 and plot 1 appearances separately.
    - Save
* Invoke nodes can be used to invoke an action on a reference item
  + E.g. use to save a figure, or copy data to excel
  + In week4\_exercise0, add an invoke node to save the figure as a jpeg when the user clicks an OK button
    - BD > Graphics & Sounds > Graphics formats > Write Jpeg File
    - Note, change the mechanical action on the button so that the figure doesn’t continuously save (note difference between latch and switch!)
    - Save as week4\_exercise0

**3. Elapsed time** *(we’ll do more timing – loops and structures next week or the following)*

* In week4\_exercise0, add an elapsed time function.
  + Add the elapsed time function to control when to begin executing the main part of the VI
  + Expand the elapsed time function to note the input / output functions.
  + Now, imagine if you want to run something repeatedly at specific intervals – e.g. a calibration curve that performs a different set of operations than the main program.
    - The timer would look different, in that it would be permitted to reset
    - Open /examples and functions / examples/week4\_timer.VI

**4. Cleaning up a diagram**

* Front panel:
  + Aligning elements
  + Saving as defaults / resetting to default
  + Tab control
  + Hiding indicators/controls
* Block diagram:
  + View as icon
* Decorations – boxes on front and block diagrams
* NAMING everything logically – really helps for local variables, sharing information between subVIs etc.

**Next 2 weeks / before the end of year:**

More on timing – time structures

Other structures – e.g. sequences

Some common errors

Reading from files

Using arrays

Error handling (maybe)

**Exercises**

Exercise 1: Open week4\_exercise0 (from the Library).

Add a **Wait** or **Wait until next MS Multiple** function within the while loop so that the data are plotted less frequently. Use a **front panel control** to set the frequency. Save-as “labview101\_week4\_exercise1” into the LabVIEW101\_library.

Exercise 2: From labview101\_week4\_exercise1, use a combination of **string and path** **controls** to set the location and filename of an output data file.

Use the path control to select the folder to which some data will be saved, and the string control to set the filename. Append a formatted date stamp to the filename/path, so that the data file is saved as: /path/to/outputdata/filename\_YYMMDD.

Add a front panel path indicator to show where the data are saved.

Be selective about **where** you build the output filepath / name (i.e. inside vs outside of the while loop), as this will affect the output filename.

Remember to use descriptive names for the objects in the VI.

Save-as “labview101\_week4\_exercise2” into the LabVIEW101\_library.

Exercise 3: Add a save to measurement file function to the “labview101\_week4\_exercise2” VI . Save two numeric signals.

DON’T link the path/filename created above to the save to measurement file function just yet. Instead, run the VI to **manually** select the save file name / destination.

Afterwards, use the file path created in the previous exercise to automatically set the filename of the output data file.

Finally, modify the VI to include **three saving options**: Write to measurement file, save to CSV file, don’t save. Use a combo box/e-num and case structure for this – pay attention to the order of items in the e-num. Again, be selective about where you place the e-num on the block diagram (inside vs outside the while loop).

Add a Boolean control to toggle data saving on/off in the first 2 cases (best to use a local variable).

Consider using local variables to make it easier to share the numeric signals to the saving functions.

Save-as “labview101\_week4\_exercise3” into the LabVIEW101\_library.

Exercise 4: Add a property node to **control the upper and lower y-limits** on the plot in your VI.

Use front panel controls to set these so that the values can be adjusted while the program is running.

**Change the precision** of the front panel controls to allow you to increase/decrease the plot scales in 0.25 increments (right click > properties; change Data entry and Display format). Consider whether the y-scale should be set to auto-scale.

Save-as “labview101\_week4\_exercise4” into the LabVIEW101\_library.

Exercise 5: Spend some time cleaning up the VI, and adjusting the appearance. Use decorations, comments, descriptive (yet concise) names for each variable, etc.

Save “labview101\_week4\_exercise4”.