



Software Maintainability & How Not to Have the Next Developer Curse Your Name

Carter Dietz

10/19/2017

Obligatory Dilbert



Dilbert.com DilbertCartoonist@gmail.com

8-12-14 ©2014 Scott Adams, Inc./Dist. by Universal Uclick

Content

- Lessons Learned from Inherited Code
- Suggested Structure to Help Subsequent Programmers



Following slides contains simple P concepts. You've been warned.

What Is Maintainability?

- IEEE Definition:
 - “The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.”



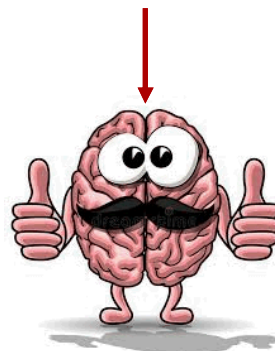
Correct
Faults



Improve
Performance



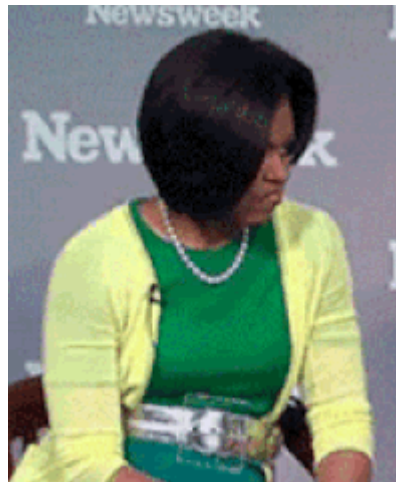
Adapt



Understand

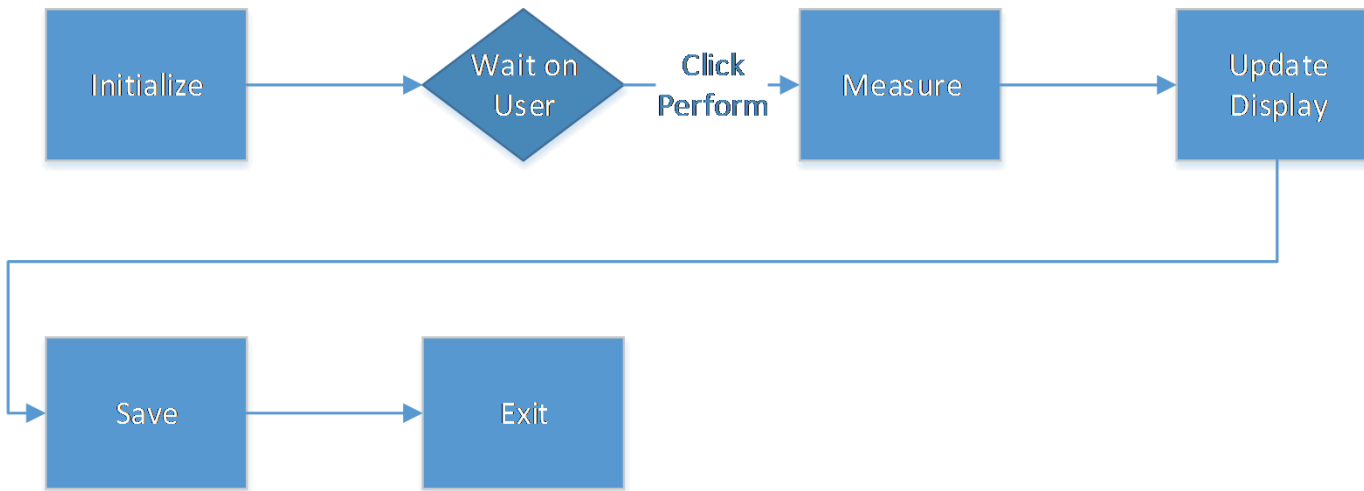
NI Guidance

- “By using forethought in designing and creating an application, you can create VIs that are more maintainable.”

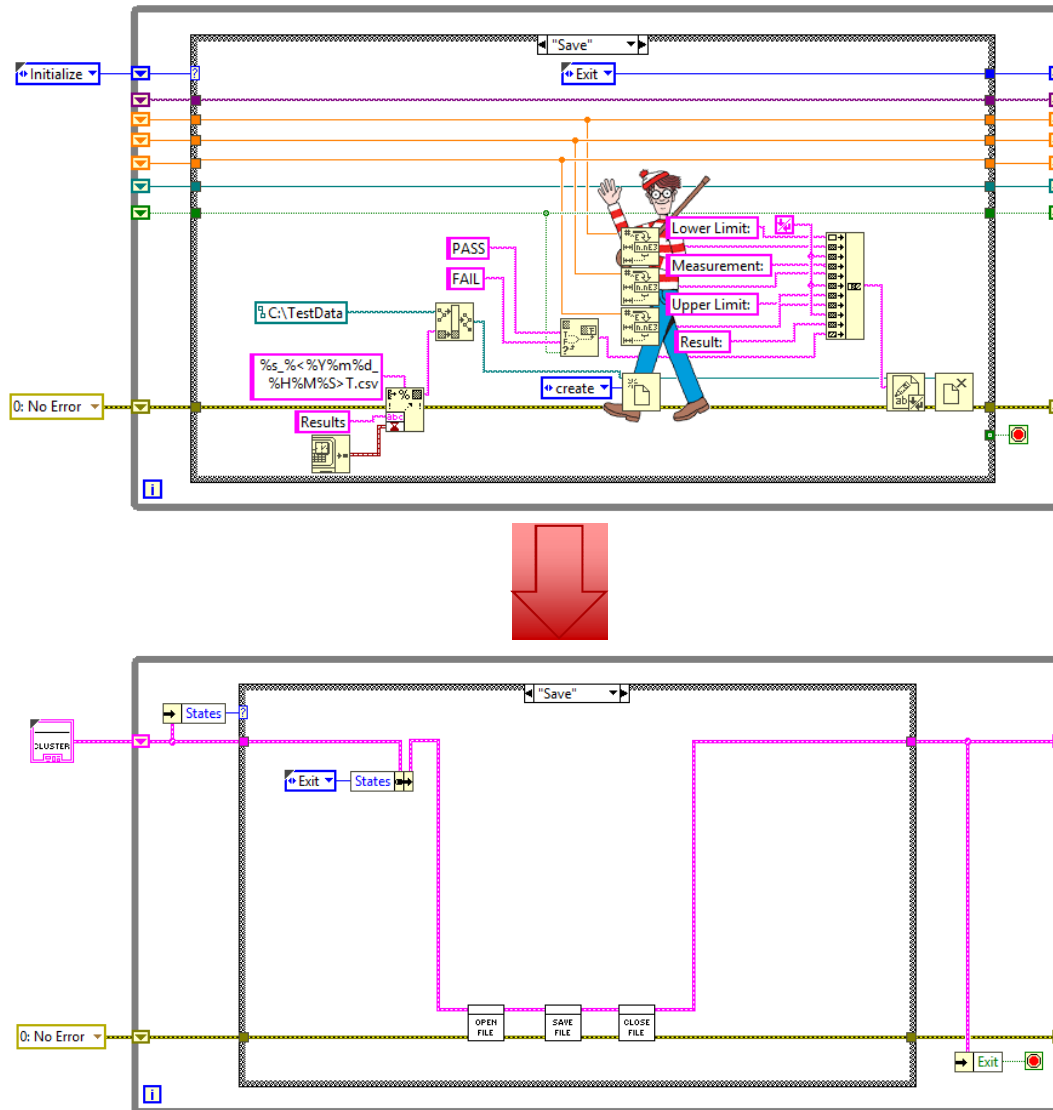


Example

- Flow Diagram



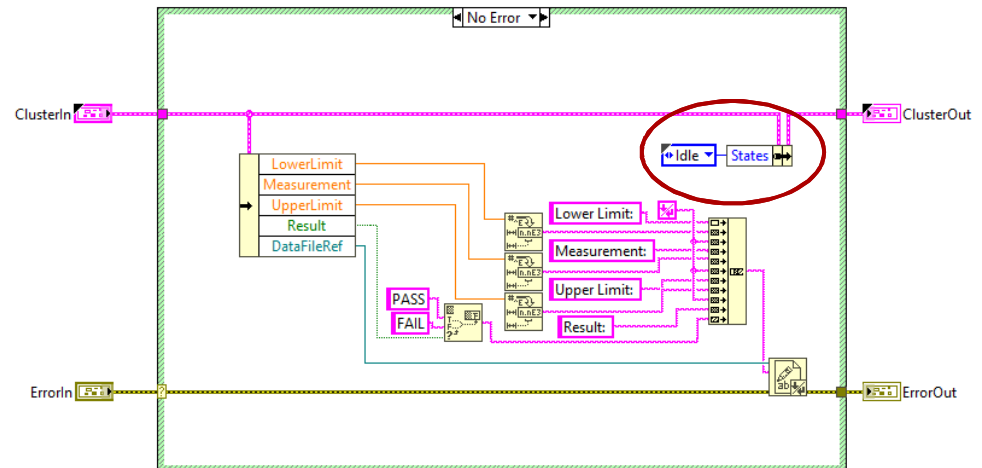
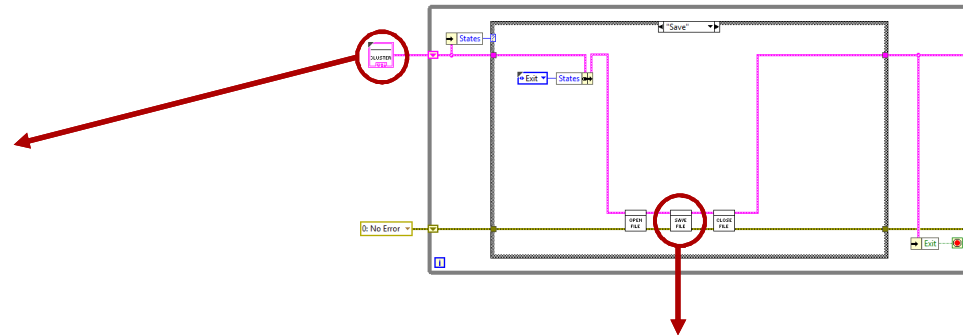
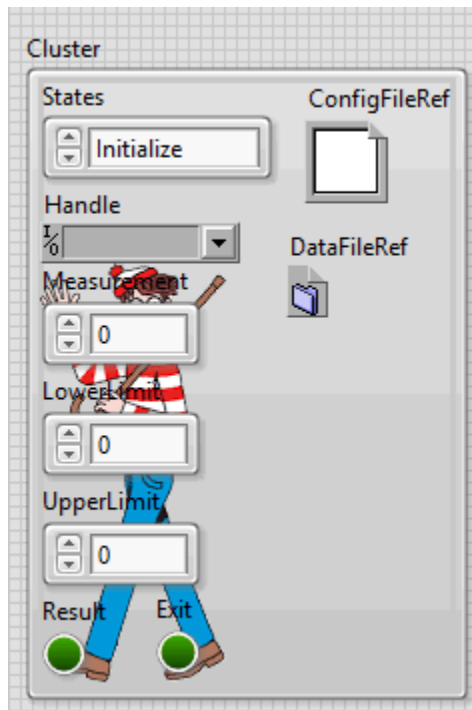
Improvement?



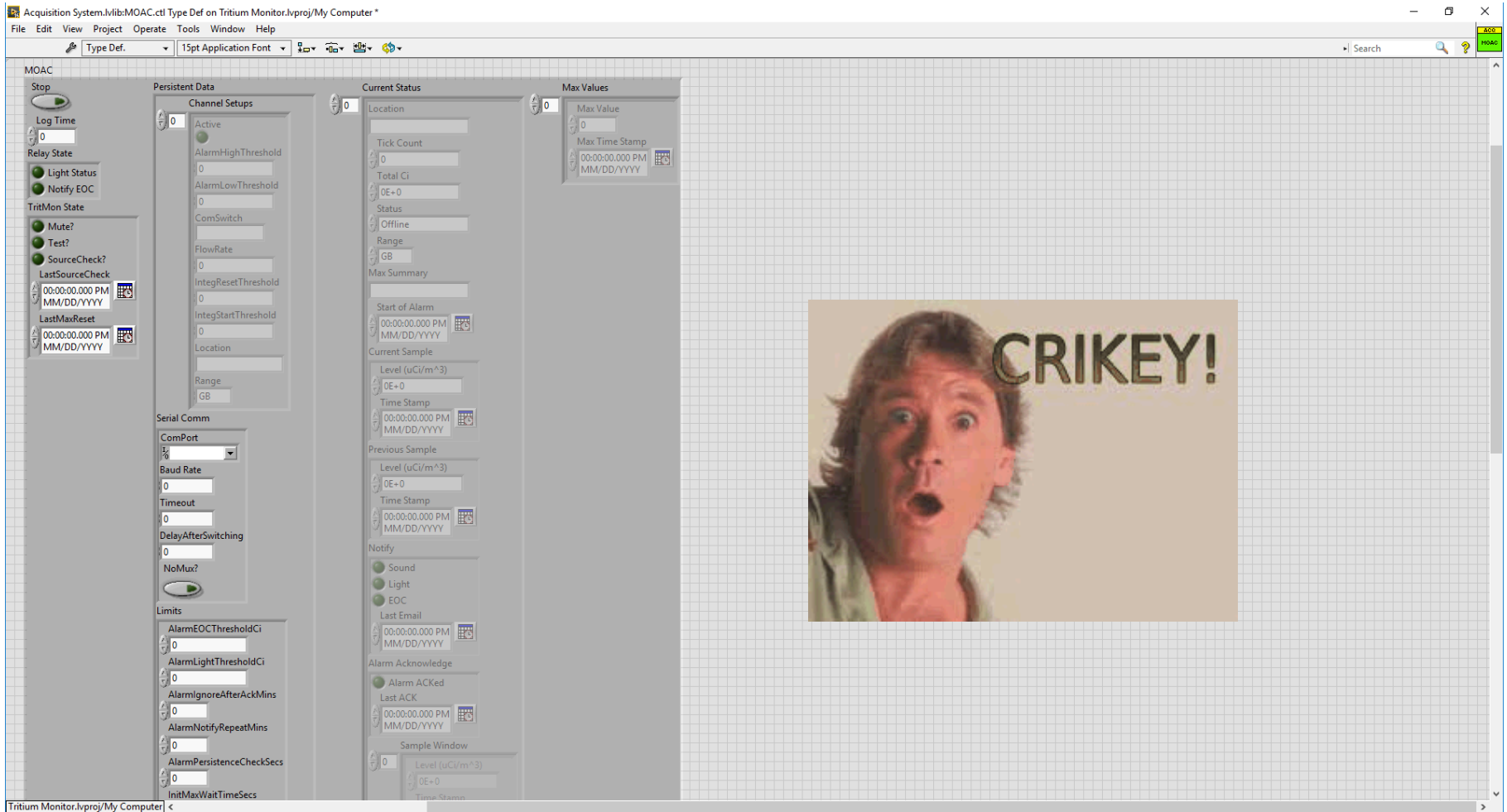
Airing of Grievances (Part 1)

- MOAC: Mother of All Clusters
 - Variables not appropriately scoped

"I got a lot of problems with you people and now you're gonna hear about it!"



MOACs in the Wild

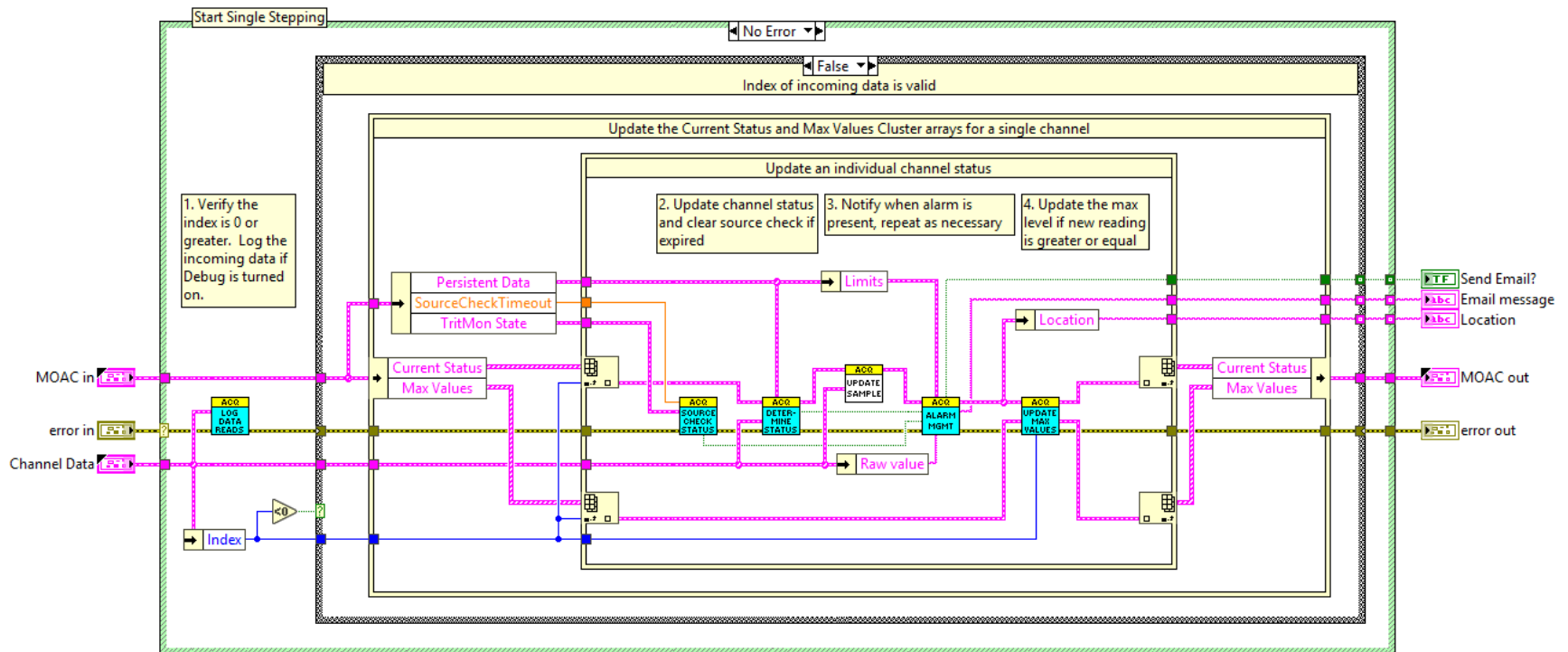


The screenshot displays the MOAC (Monitoring and Alarm Control) software interface. The window title is "Acquisition System.Ivlib:MOAC.cti Type Def on Tritium Monitor.Ivproj/My Computer". The interface is divided into several panels:

- MOAC:** Contains a "Stop" button, "Log Time" (0), "Relay State" (Light Status, Notify EOC), and "TritMon State" (Mute?, Test?, SourceCheck?, LastSourceCheck, LastMaxReset).
- Persistent Data:** Includes "Channel Setups" (Active, AlarmHighThreshold, AlarmLowThreshold, ComSwitch, FlowRate, IntegResetThreshold, IntegStartThreshold, Location, Range) and "Serial Comm" (ComPort, Baud Rate, Timeout, DelayAfterSwitching, NoMux?).
- Limits:** Lists thresholds for AlarmEOCThresholdCi, AlarmLightThresholdCi, AlarmIgnoreAfterAckMins, AlarmNotifyRepeatMins, AlarmPersistenceCheckSecs, and InitMaxWaitTimeSecs.
- Current Status:** Shows "Location", "Tick Count", "Total Ci", "Status" (DE+0, Offline), "Range" (GB), "Max Summary", "Start of Alarm", "Current Sample" (Level, Time Stamp), "Previous Sample", "Notify" (Sound, Light, EOC), "Last Email", "Alarm Acknowledge" (Alarm ACKed, Last ACK), and "Sample Window".
- Max Values:** Displays "Max Value", "Max Time Stamp", and "MM/DD/YYYY".

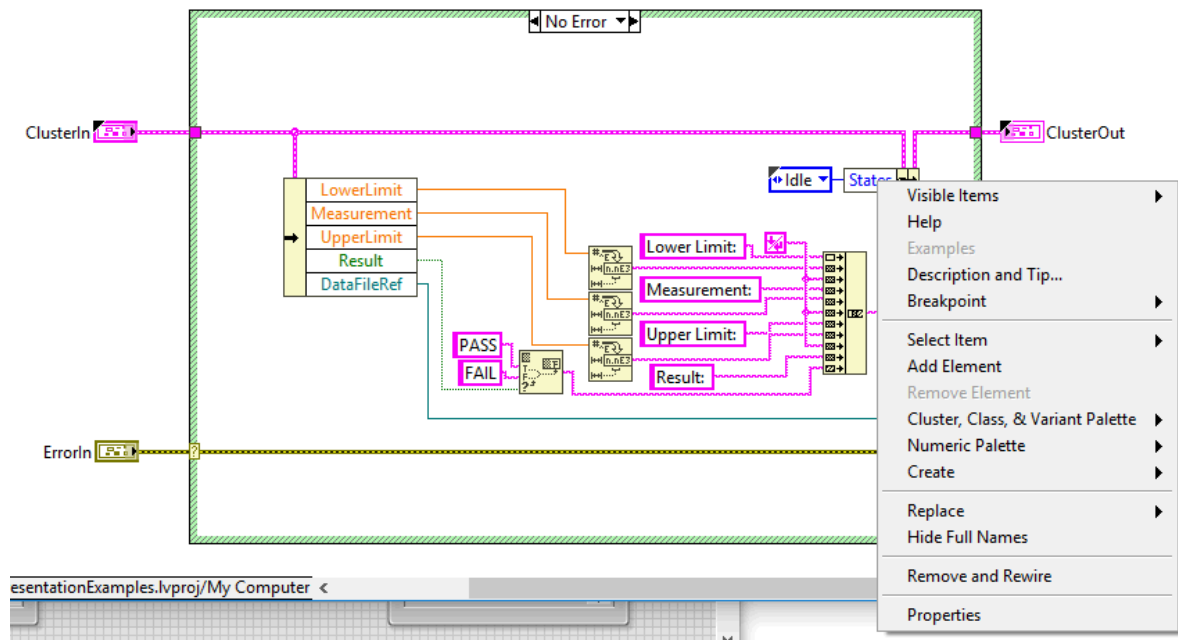
A meme image of a shocked man with the text "CRIKEY!" is overlaid on the right side of the interface.

Mo' Wires, Mo' Problems



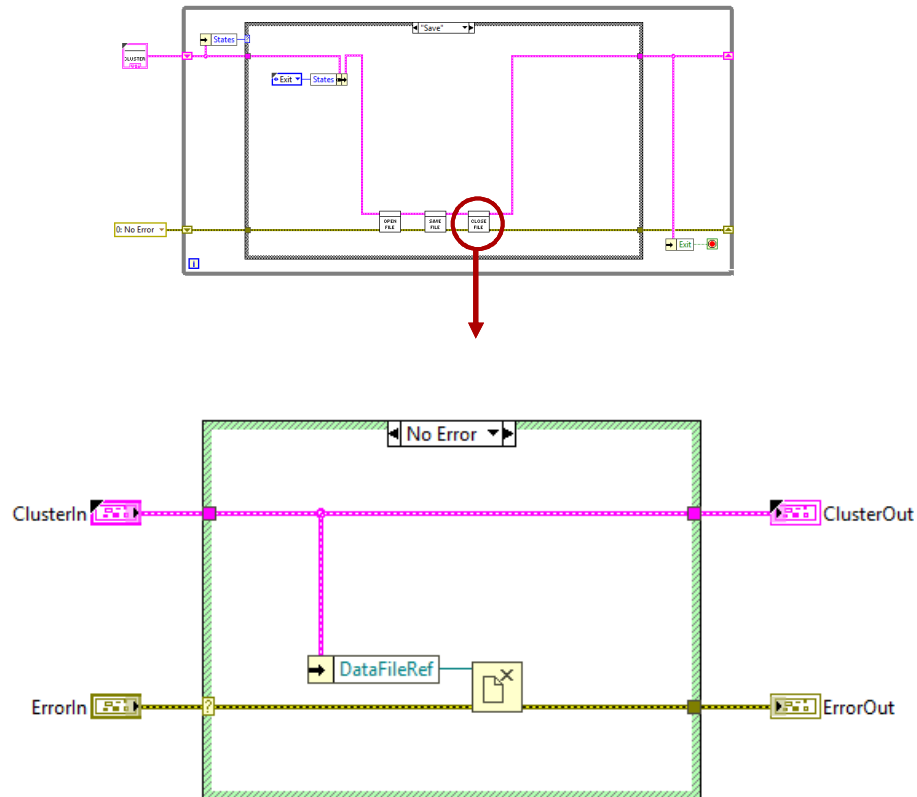
Airing of Grievances (Part 1.A)

- There's No Find All Instances for a Bundle By Name
 - Where is this property being accessed?
 - Where is this property being updated?
 - How many times is this property updated?



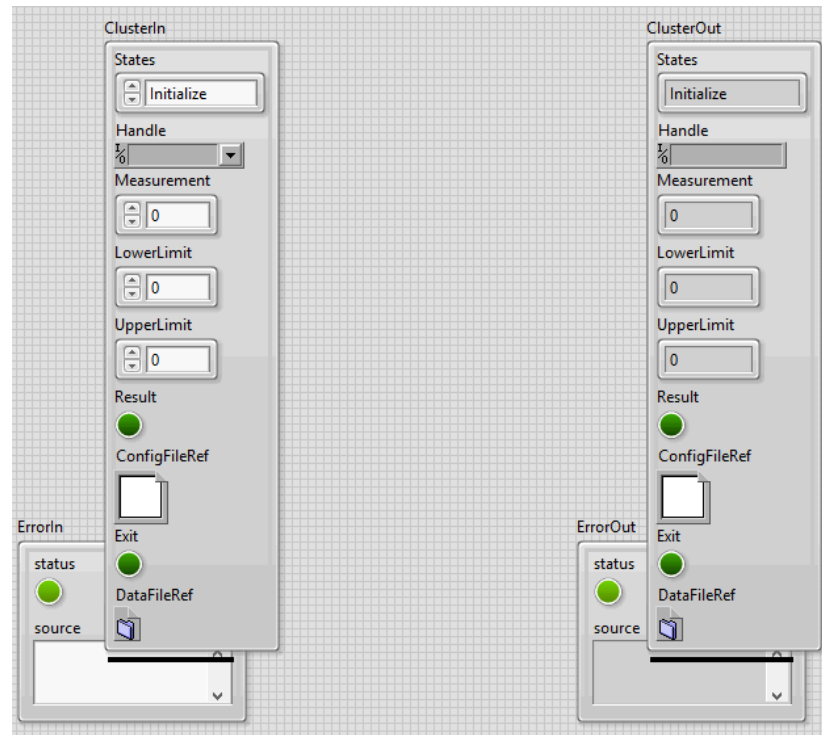
Airing of Grievances (Part 2)

- Unnecessary SubVI Returns
 - A method return implies the data is being manipulated



Airing of Grievances (Part 3)

- Controls on SubVIs Front Panels
 - Overlapping
 - Hard to Fit on the Screen



LVOOP to the Rescue?



Improved Maintainability

- Use LV Classes to Encapsulate Methods
- Create Accessors to Provide Scope for Properties

- LabVIEW Object-Oriented Programming FAQ
 - What is the benefit to an experienced G programmer who does not practice OOP?
 1. Easier to debug—More modular code allows you to narrow the list of VIs where a bug might be.
 2. Easier maintenance—OO design leads to more structure, which helps developers to be more productive over time. Developers can easily identify areas where they want to add features, and are less likely to introduce errors into unrelated sections of code.

Quick Detour on LVOOP

- LV Class = Library + Cluster + Scope

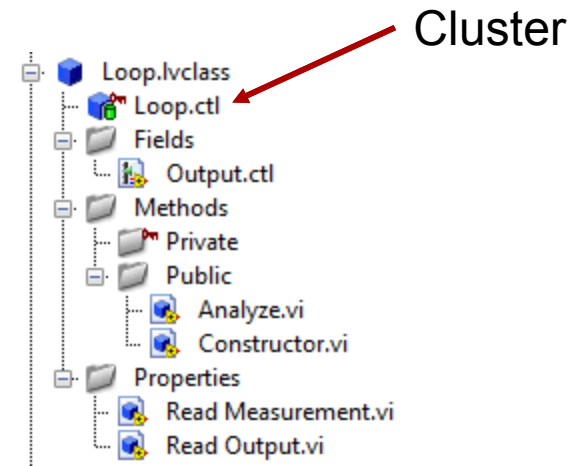
- Scope

- Accessibility

- Public – Anybody
- Private – Only VIs in the class library
- ~~– Protected – Any class that inherits from~~
- ~~– Community – Friends of class library~~

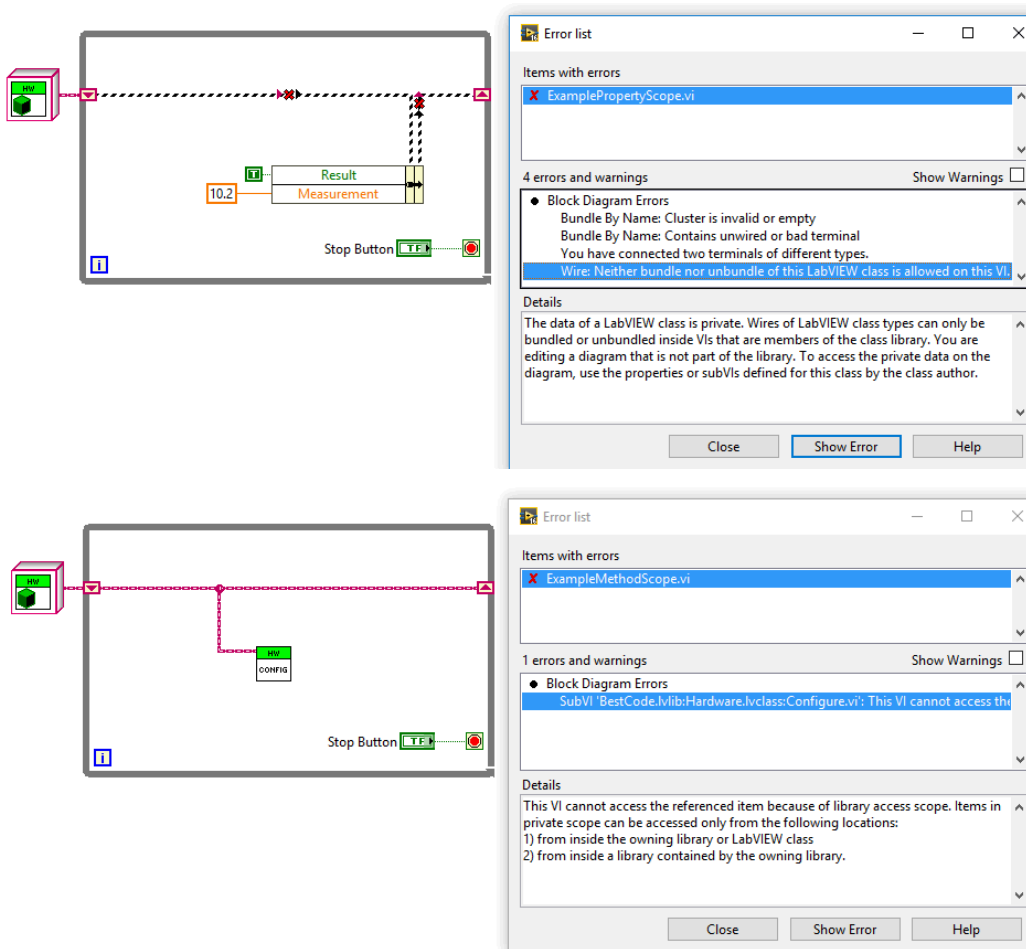
- Where an element of a cluster can be read from and/or written to

- Where a subVI can be called from or under what context a subVI can be called



Quick Detour on LVOOP

■ Scope Example



The top diagram shows a subVI call with a cluster element out of scope. The error list shows:

- Items with errors: ExamplePropertyScope.vi
- 4 errors and warnings: Show Warnings
- Block Diagram Errors:
 - Bundle By Name: Cluster is invalid or empty
 - Bundle By Name: Contains unwired or bad terminal
 - You have connected two terminals of different types.
 - Wire: Neither bundle nor unbundle of this LabVIEW class is allowed on this VI
- Details: The data of a LabVIEW class is private. Wires of LabVIEW class types can only be bundled or unbundled inside VIs that are members of the class library. You are editing a diagram that is not part of the library. To access the private data on the diagram, use the properties or subVIs defined for this class by the class author.

The bottom diagram shows a subVI call with a method out of scope. The error list shows:

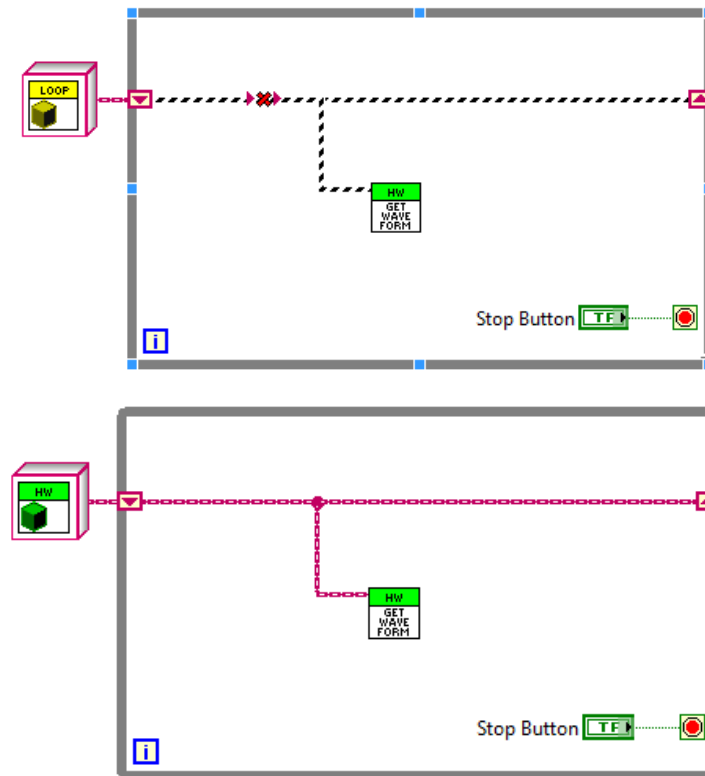
- Items with errors: ExampleMethodScope.vi
- 1 errors and warnings: Show Warnings
- Block Diagram Errors:
 - SubVI 'BestCode.lvlib:Hardware.lvclass:Configure.vi': This VI cannot access the
- Details: This VI cannot access the referenced item because of library access scope. Items in private scope can be accessed only from the following locations:
 - 1) from inside the owning library or LabVIEW class
 - 2) from inside a library contained by the owning library.

Cluster elements are out of scope of the calling subVI

Method is out of scope of the calling subVI

Quick Detour on LVOOP

- Type Safety Example



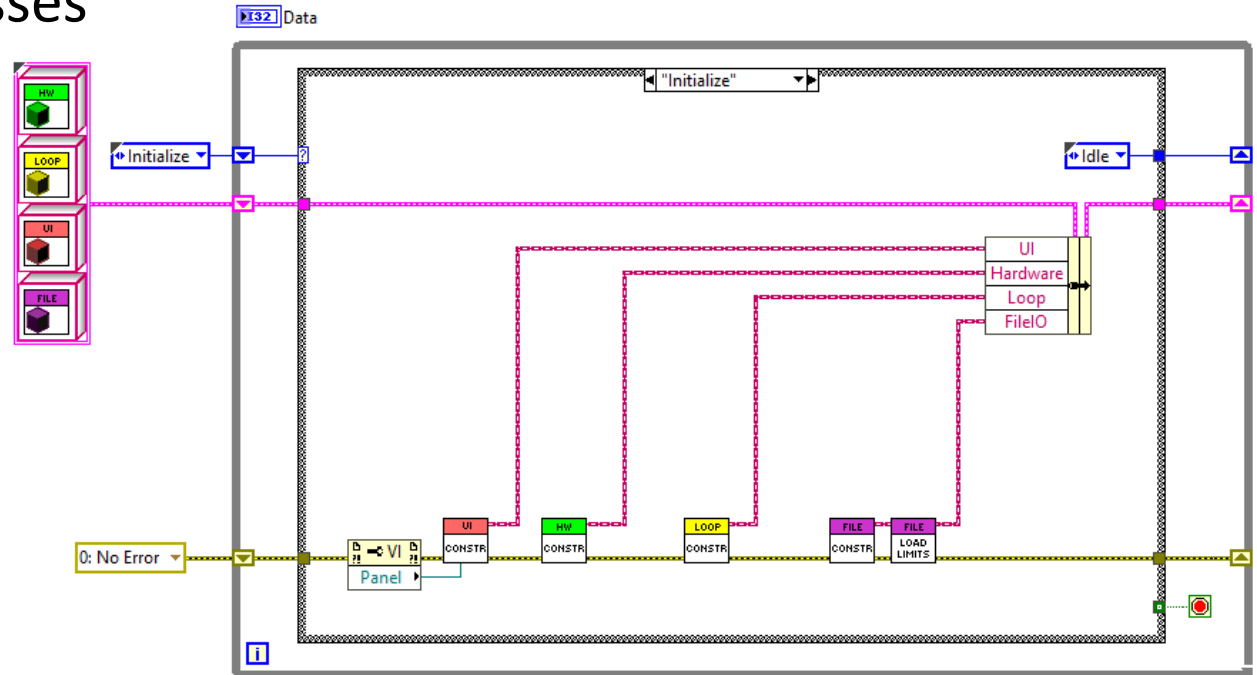
Class input type
needs to match the
class of the
connector pane

Class input type
matches the class
of the connector
pane

Recommend Structure

- Create Four Classes

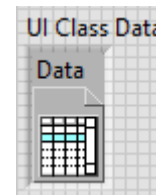
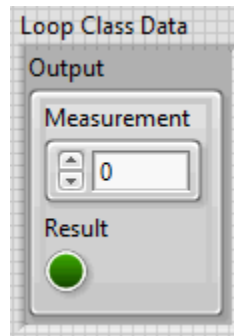
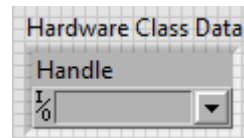
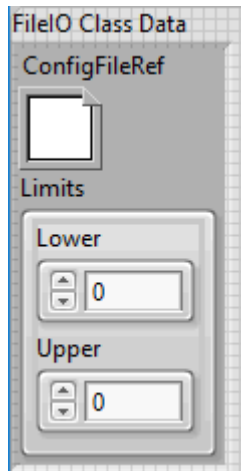
- Hardware
- Loop Data
- User Interface
- File IO



- Place Classes in Cluster Shared Between States

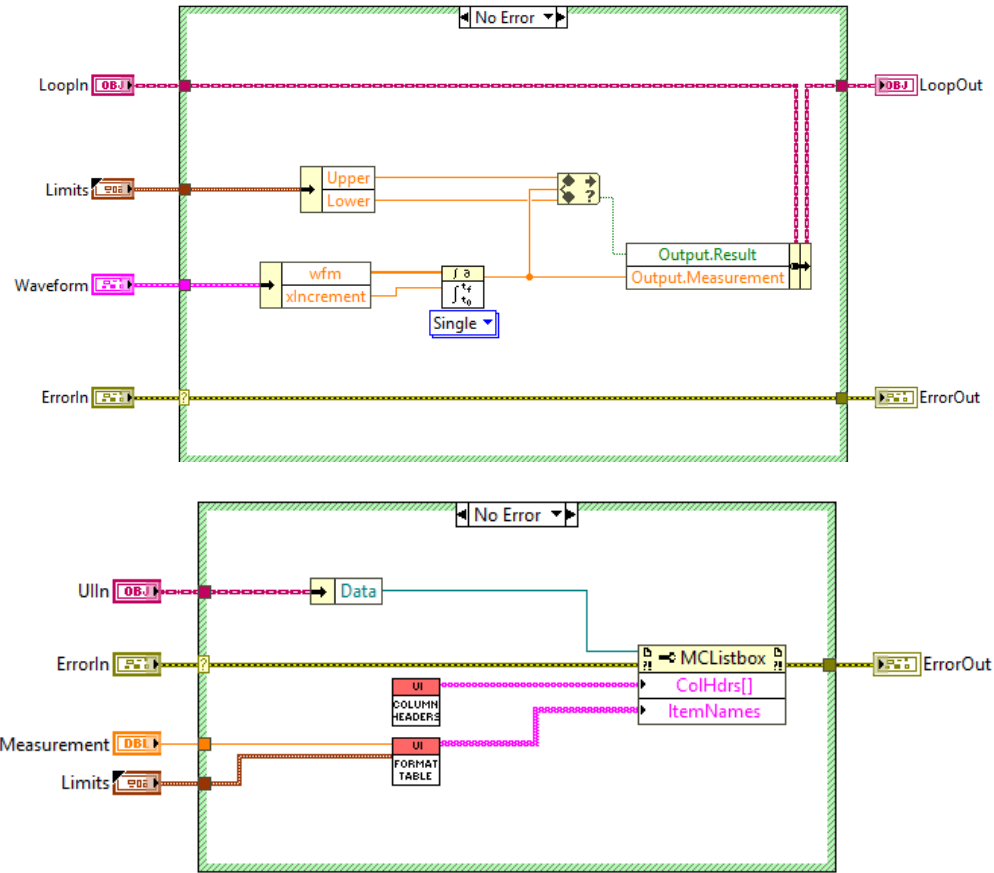
Benefits

- “A place for everything and everything in its place.”
 - Properties are grouped functionally



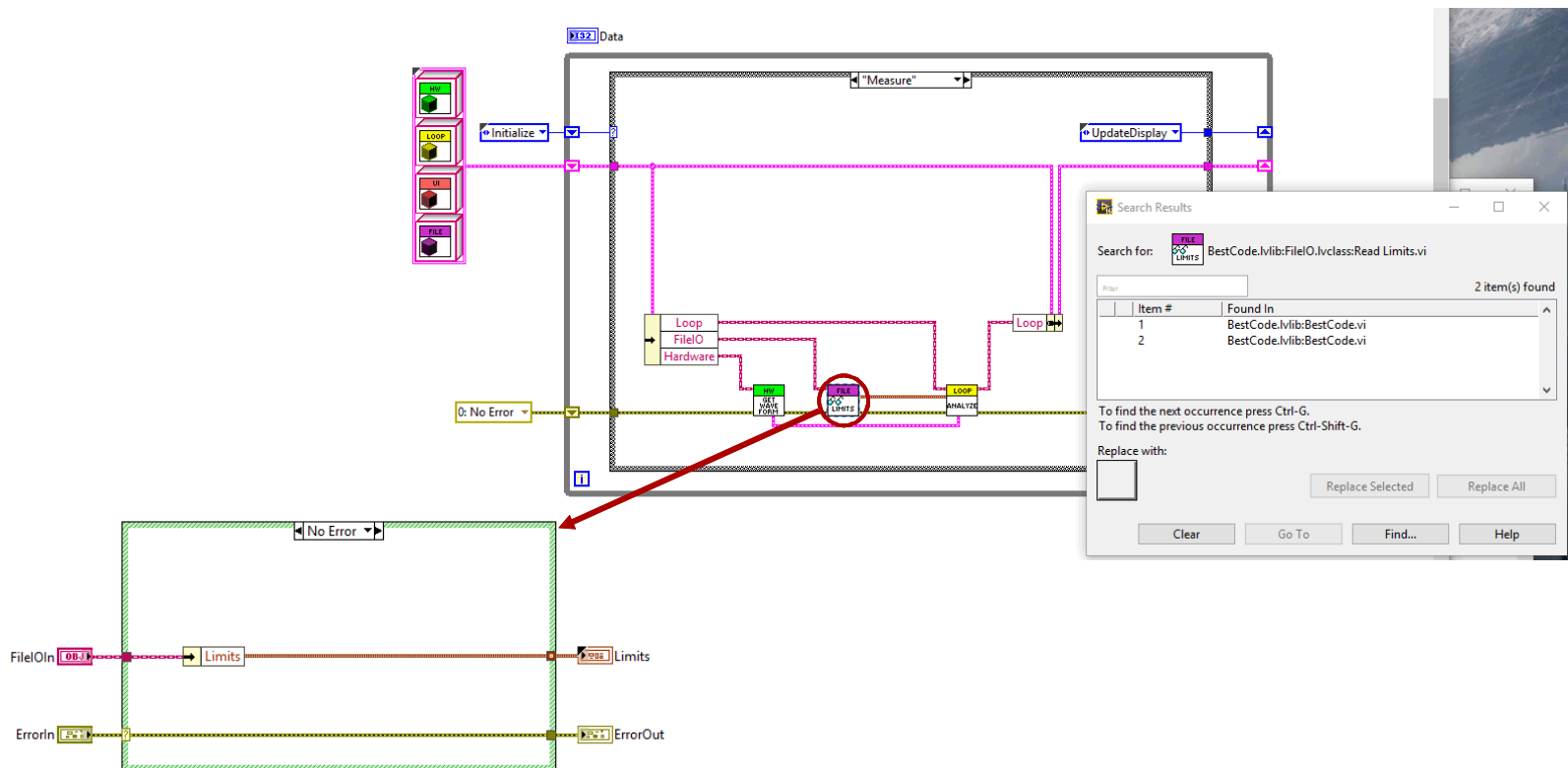
Benefits

- Properties and Methods are Scoped



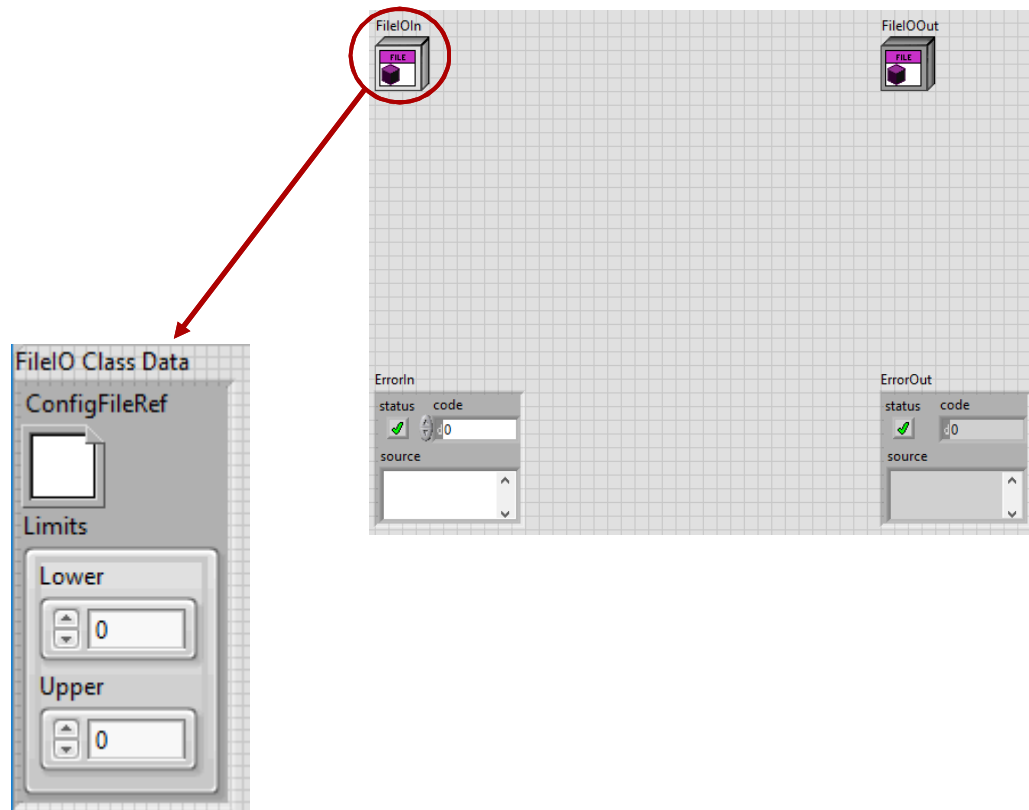
Benefits

- Easier to Find Where Properties are Modified
 - Methods of the class – Subset of all the project VIs
 - Accessors – Searchable with LabVIEW IDE



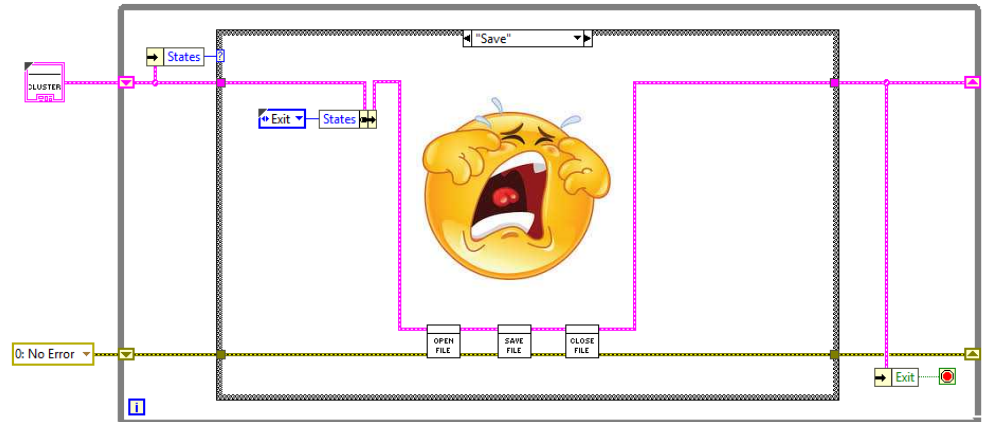
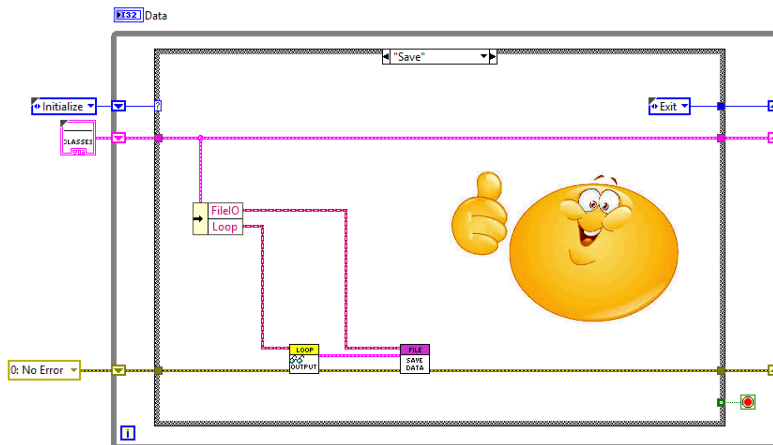
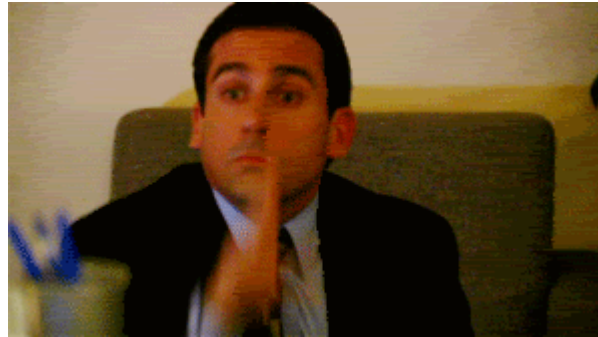
Benefits

- SubVI Front Panels Are Cleaner
 - Private class data is represented by LVOOP object image

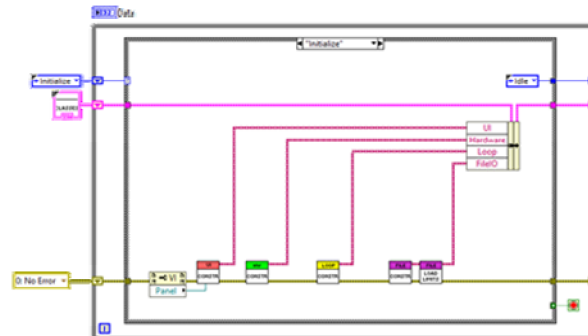
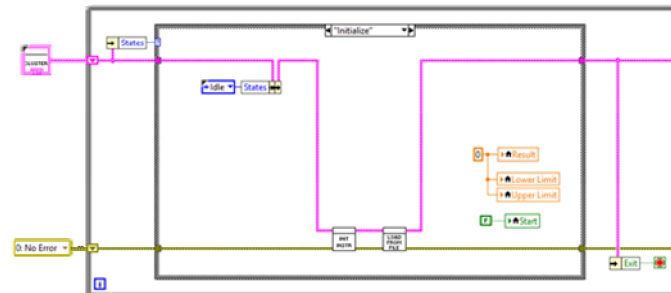
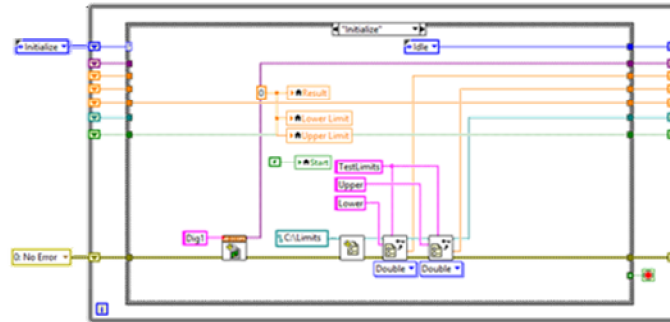


Shortcomings

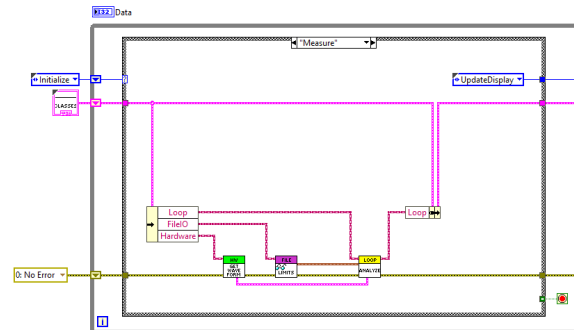
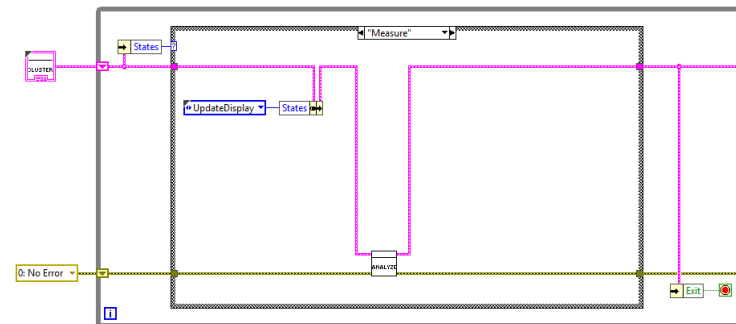
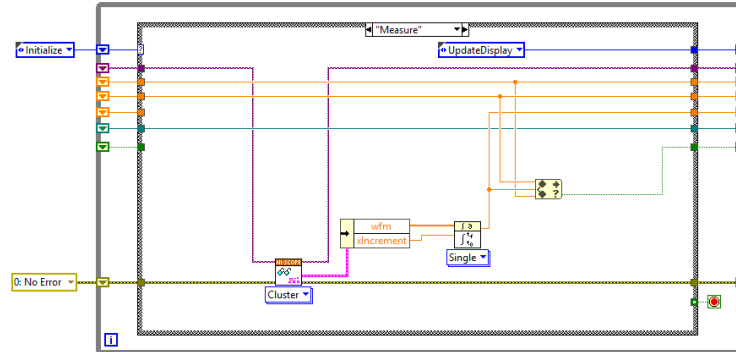
- No Enforcement of Non-Modified Method Returns
 - Just don't it!



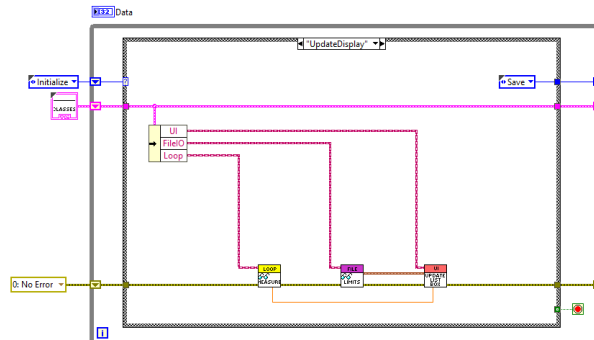
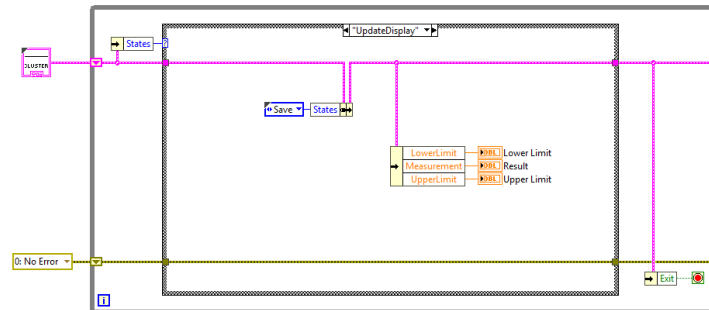
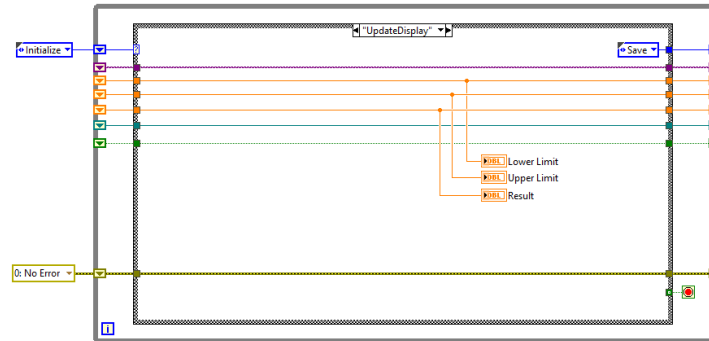
Comparison of States - Initialize



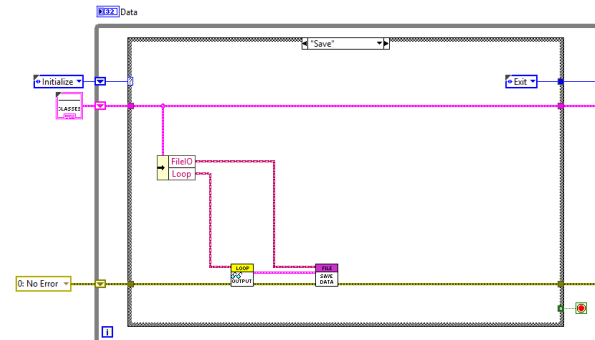
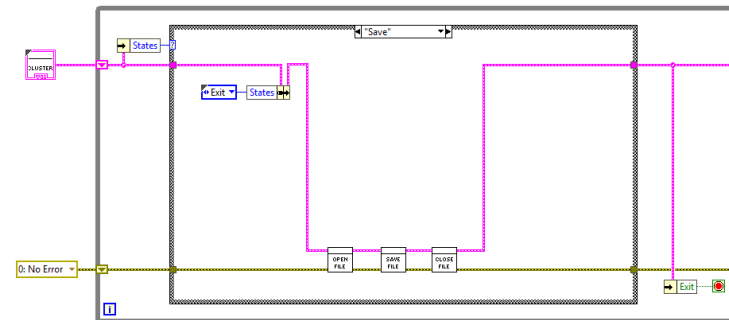
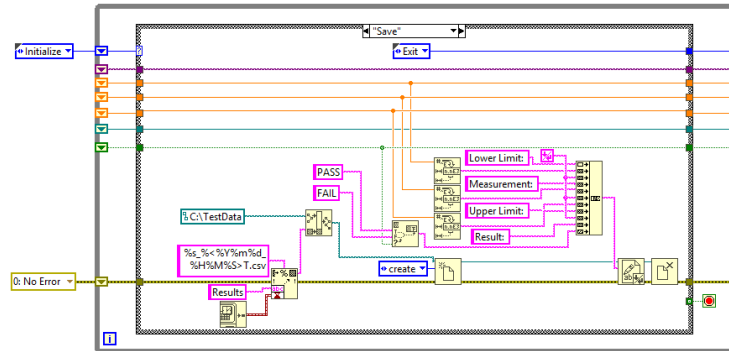
Comparison of States - Measure



Comparison of States – Update



Comparison of States - Save



Comparison of States - Exit

