NI CONNECT

LabVIEW Interfaces: Things Better Left Unstated

Stephen Loftus-Mercer
“Aristos Queue”
Principal Software Architect
LabVIEW R&D, NI

LabVIEW Interfaces

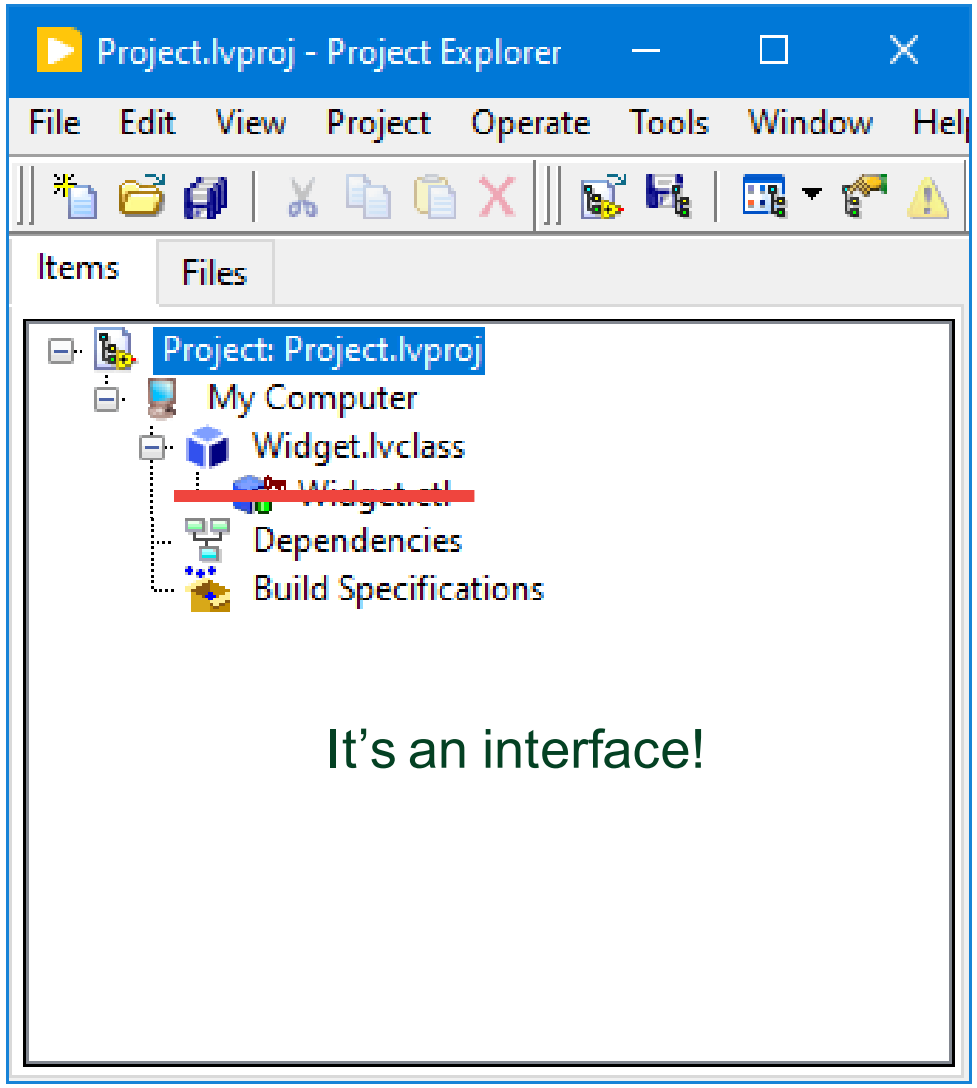
Brand New!
(in LabVIEW 2020)
(Thanks, COVID-19)

My Earlier “Interfaces” Videos

Stephen Loftus-Mercer and Allen Smith
Introduction to G Interfaces in LabVIEW 2020 (YouTube)

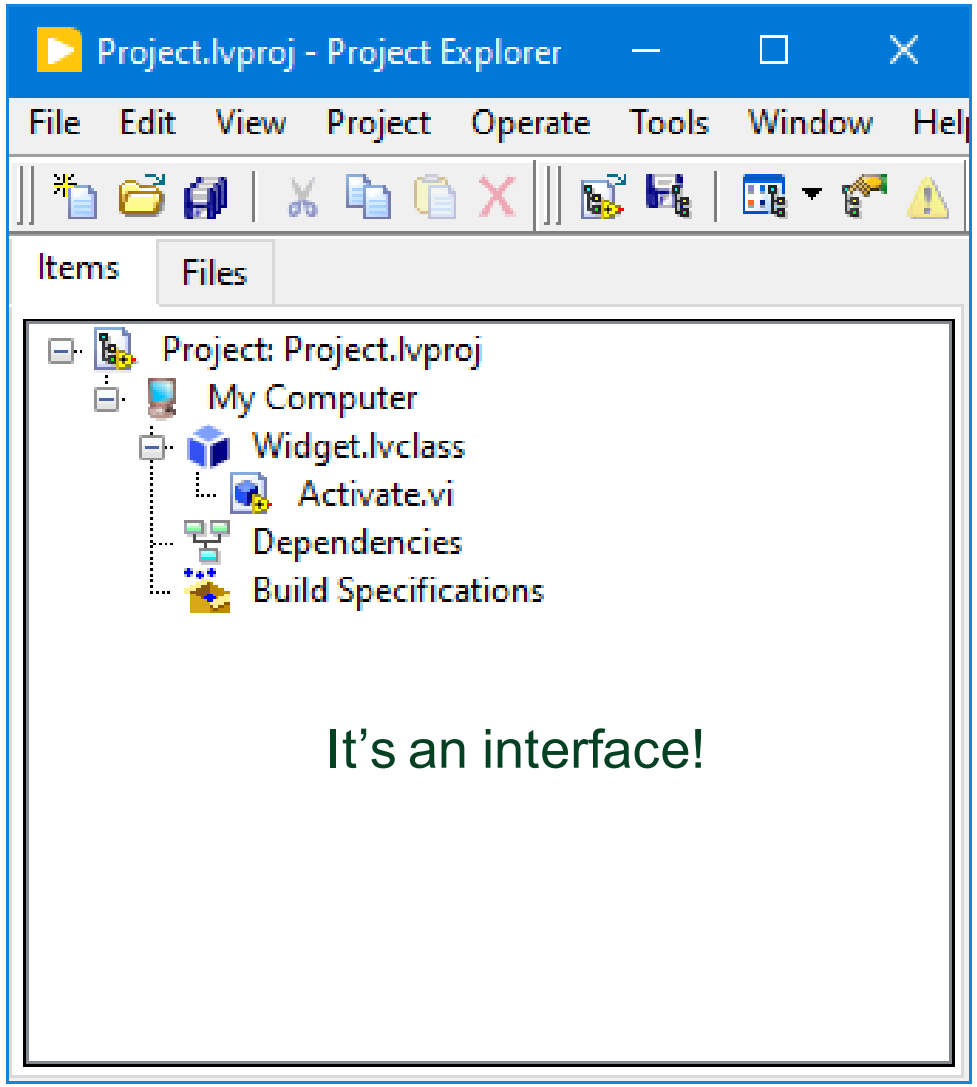
Stephen Loftus-Mercer and Jon McBee
Using LabVIEW Interfaces for Better Orbital Satellite Support (LabVIEW Wiki Video)

Interface = Class Without Private Data Control



1. Interfaces are stateless types.

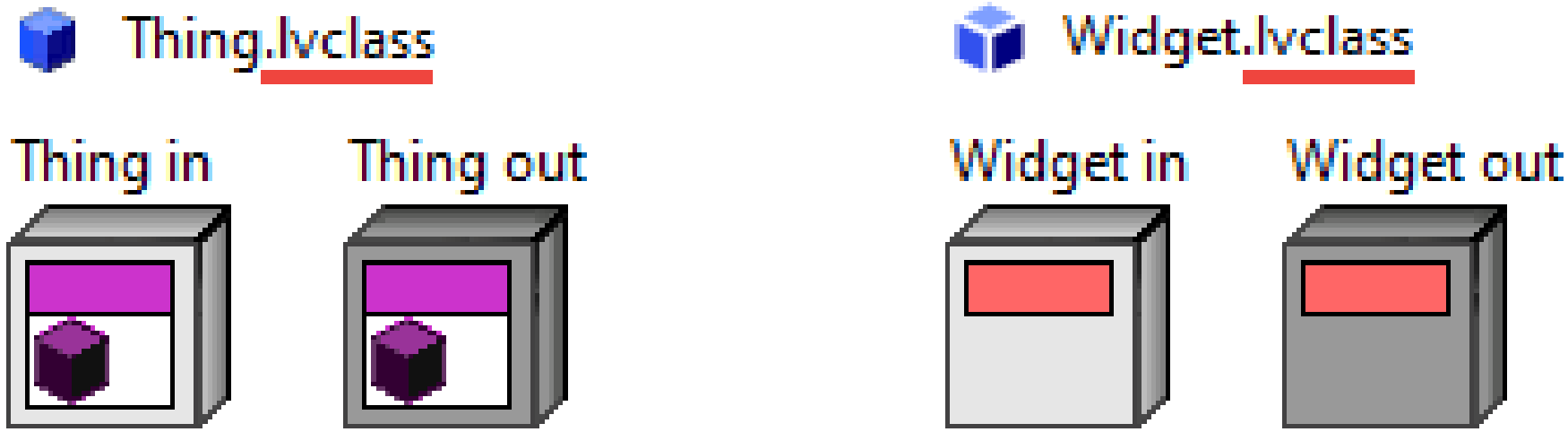
Interface = Class Without Private Data Control



1. Interfaces are stateless types.
2. Interfaces only define behaviors.

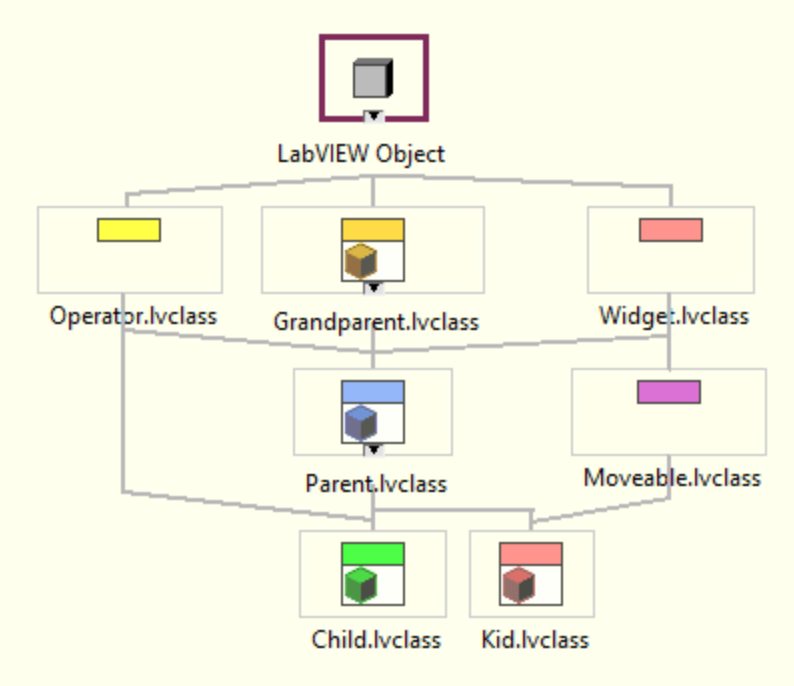
That's mostly it.

Different Icon; Same File Extension



- Users of types see them as essentially identical.
- Developers of types see them as distinct.

Inheritance



Class: One class parent, infinite interface parents

Interface: LabVIEW Object, infinite interface parents

Use of Interfaces

Scalability

Interface Segregation

Mock Testing

Cross-Hierarchy Functionality

Modularity

Separation of Concerns

Interface vs Implementation

Decorator Pattern

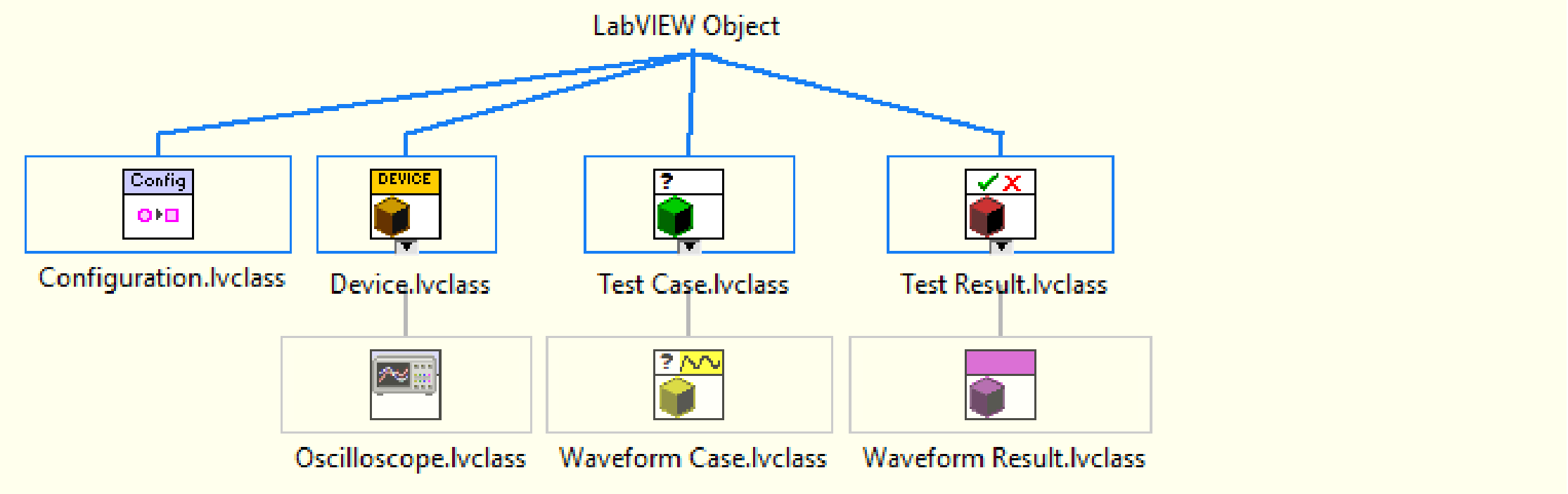
Dependency Inversion

Abstraction

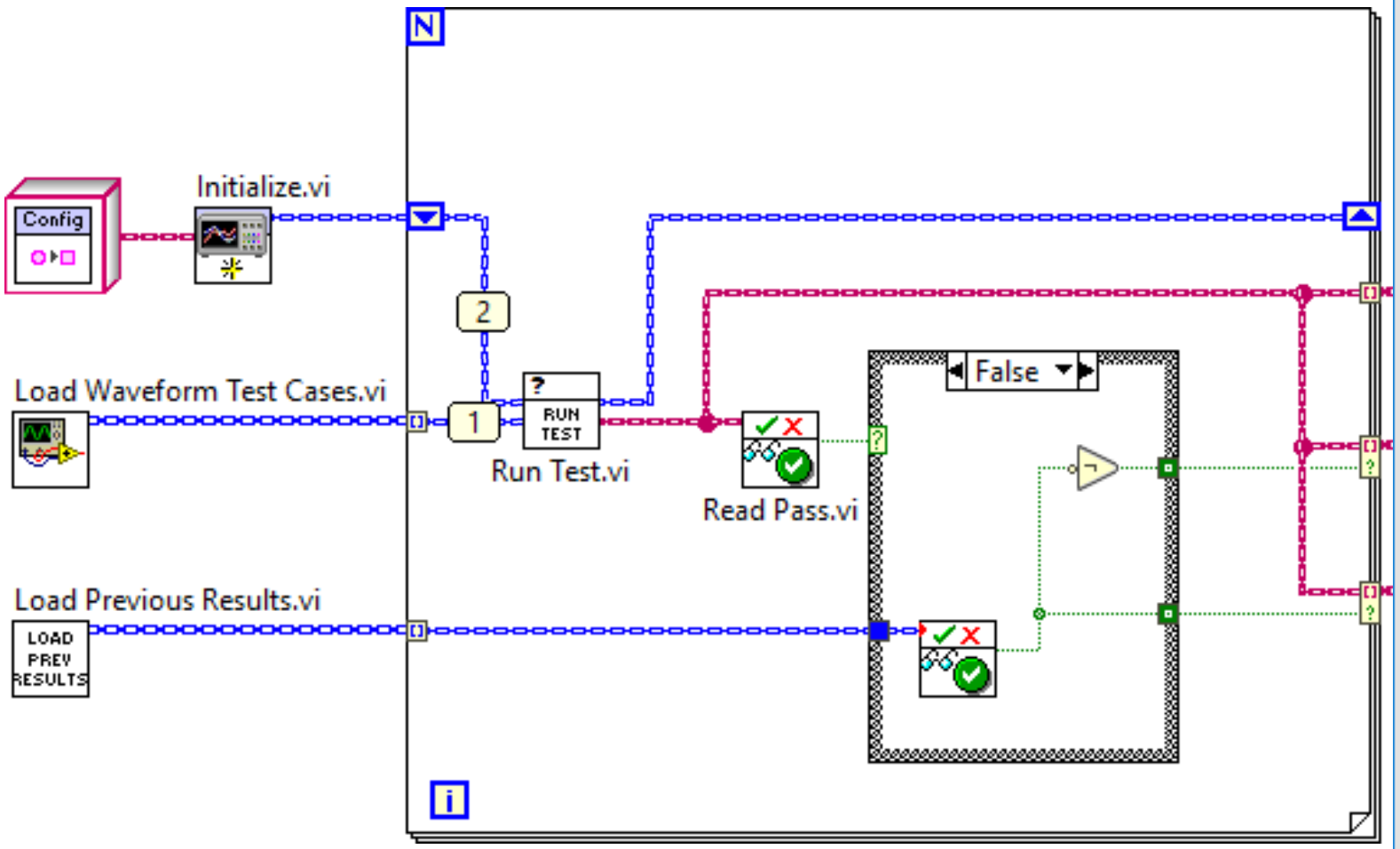
I Have Written A Simple Test Executive

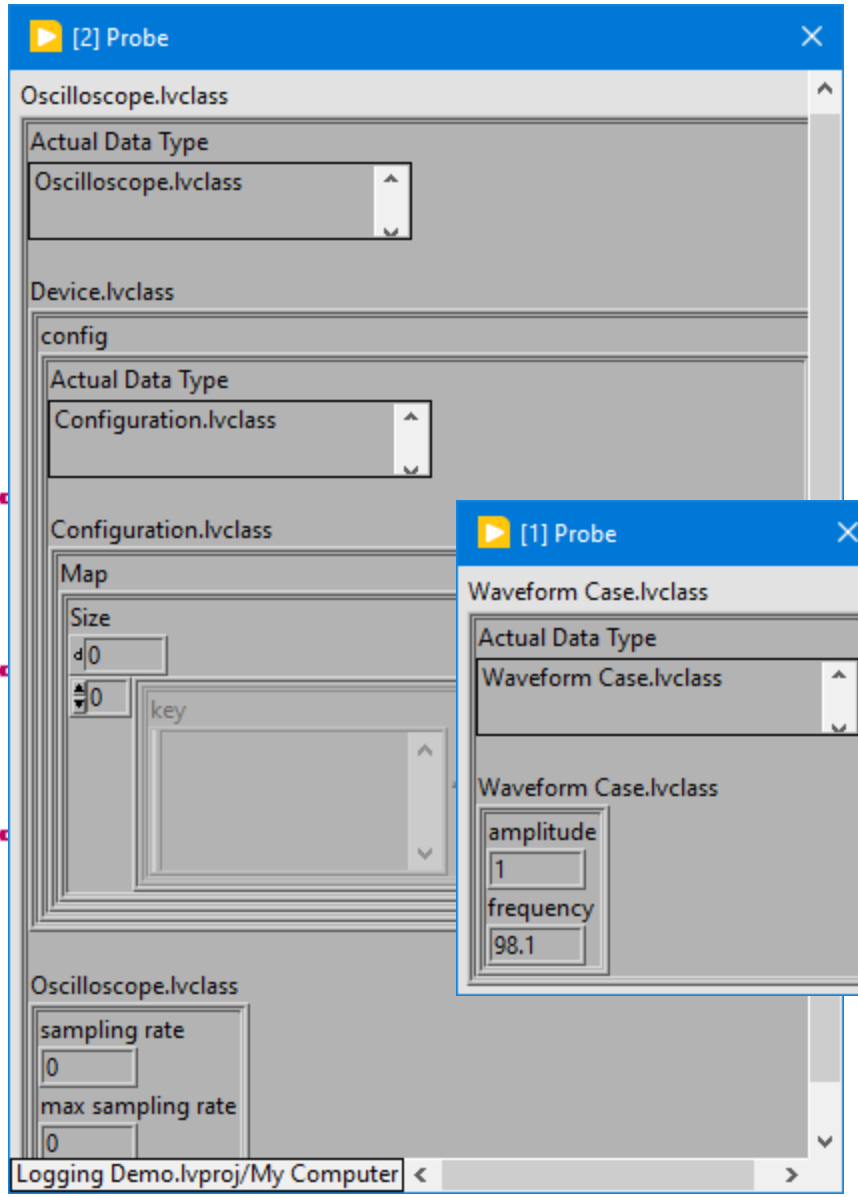
- **WARNING!!!**
- **There are plenty of good test executives in the world.**
- **You should probably go buy TestStand.**

A Simple Test Executive

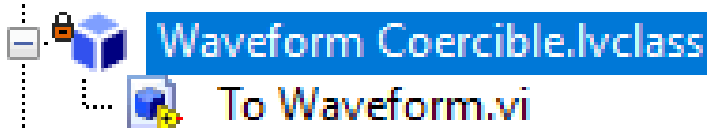


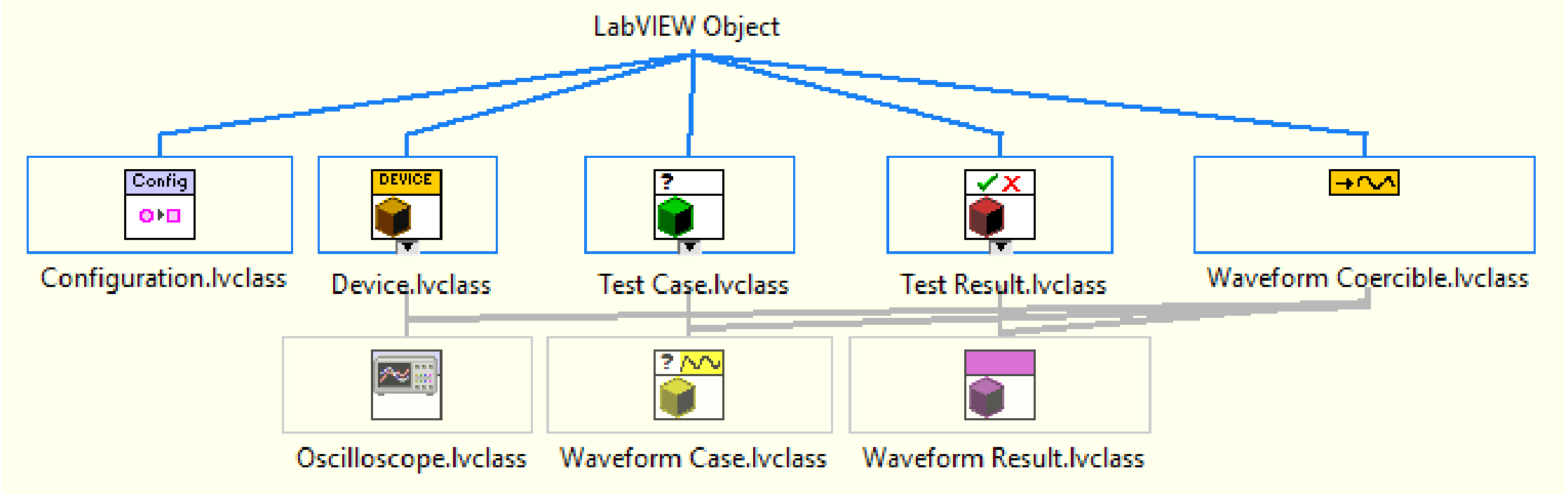
A Simple Test Executive



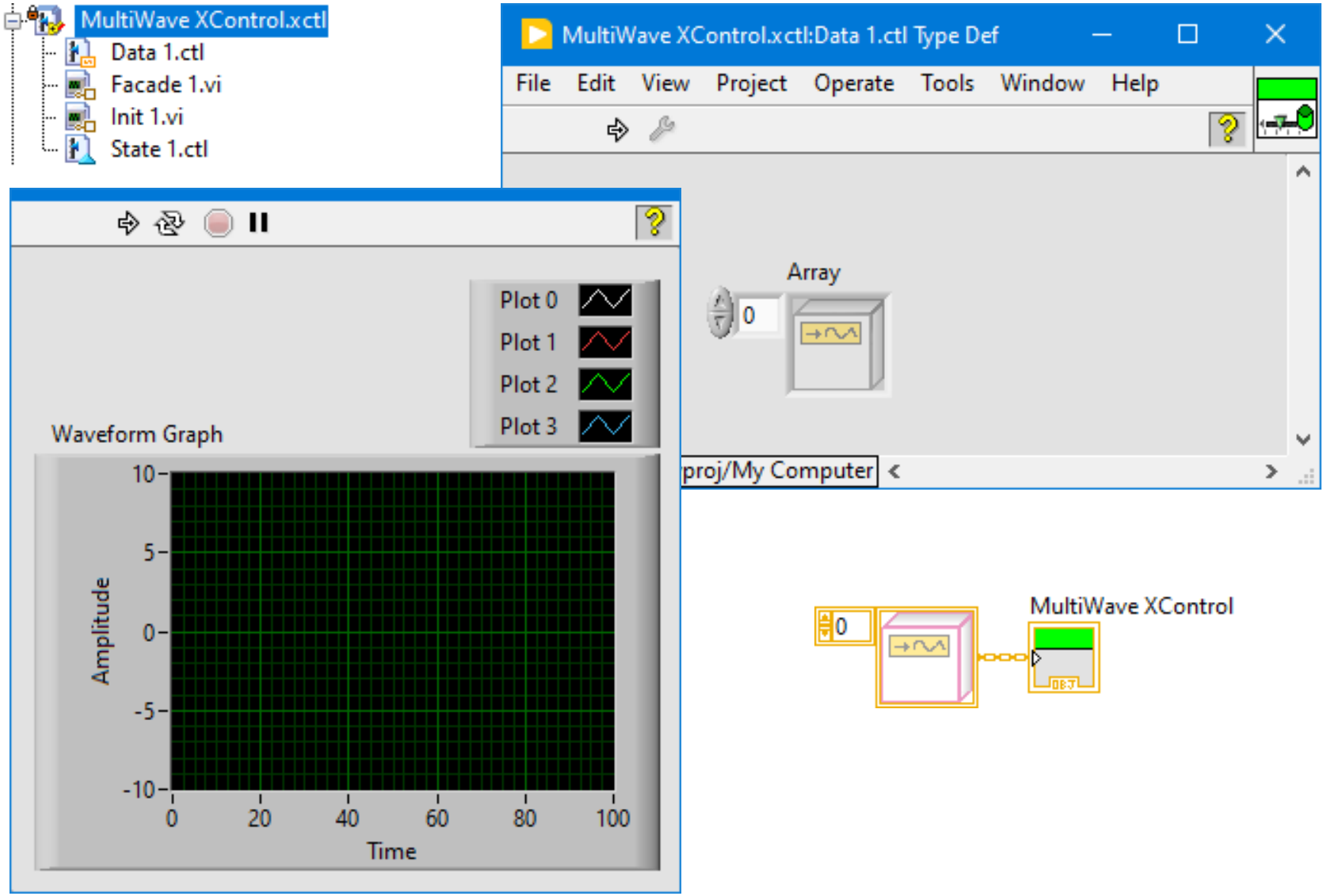


A Simple Test Executive

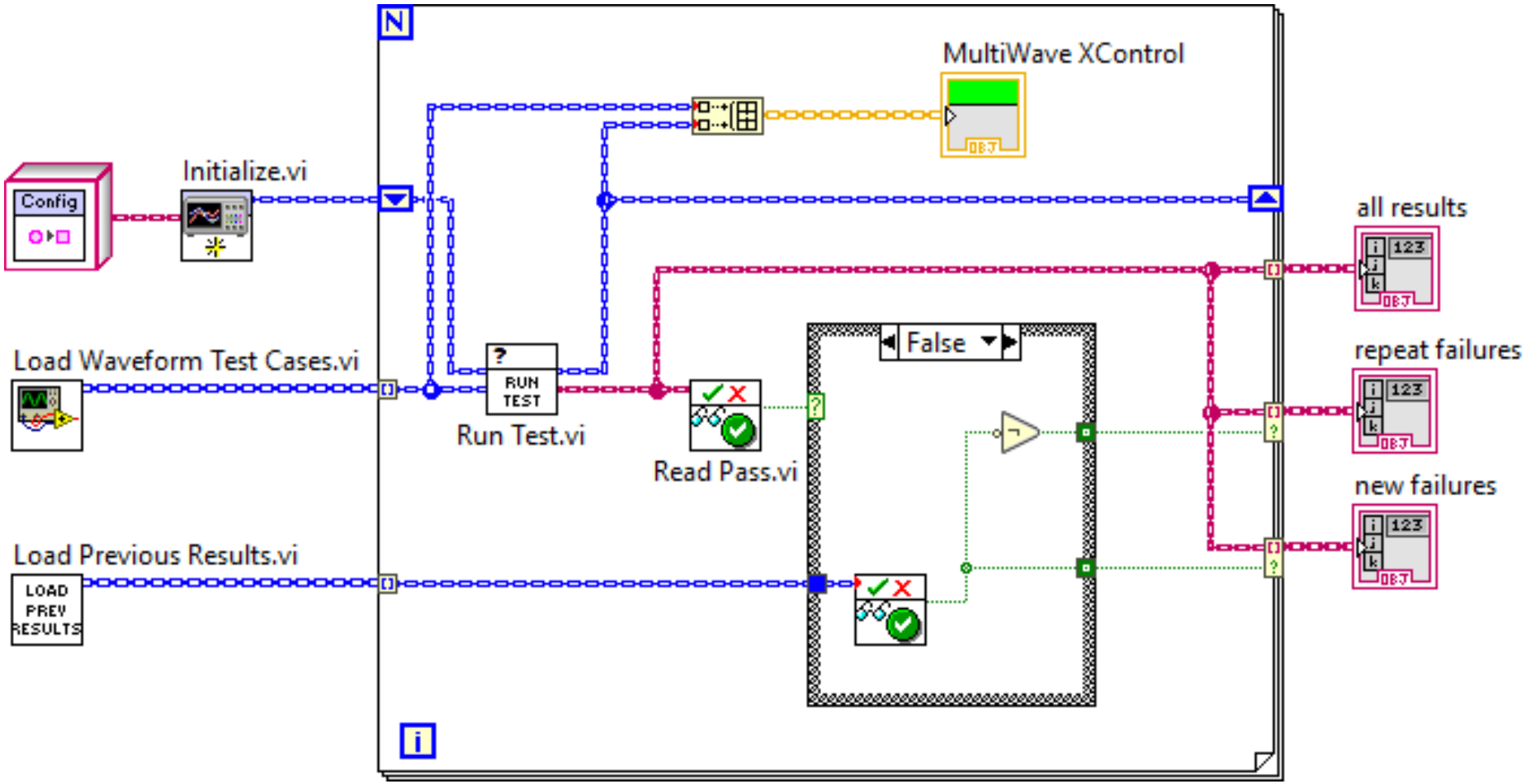




A Simple XControl to Display Waveform Data



A Simple Test Executive



Interfaces Can Use Most Method Options

- Most methods on an interface will be public scope.
  - But you can have methods of any scope.

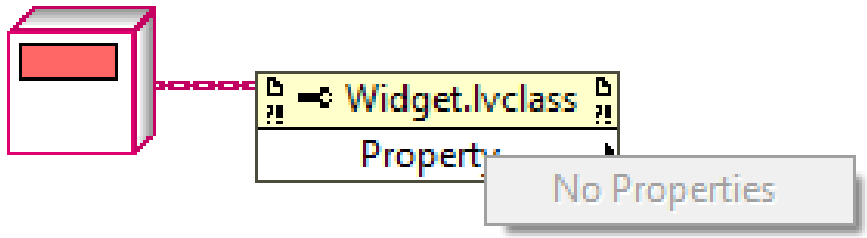
- Most methods on an interface will be dynamic dispatch.
  - But you can have static dispatch methods.

- Dynamic dispatch methods on an interface default to “must override”.
  - But you can turn that option off as needed to create default method implementations.
  - But methods cannot invoke the Call Parent Class Method node once overridden.

- But interfaces always enable restrictions on New/Delete of Data Value References.

Interfaces Do Not Support Property Node Syntax

- We wish interface properties worked, but they turn out to be tricky to define, so we axed the feature.



Prefer Interfaces Over Abstract Classes. Why?

Philosophical Answer

- Interfaces define what to do, not how to do it.
- Classes define how to do it.

- Therefore: use of interfaces limits the temptation to pile everything in one type.

Practical Answers

- Ability to apply across hierarchies.
- Multiple inheritance.
- An interface does not lock when simultaneously loaded onto multiple targets.

Customer Education

Full Courses:

1. Newly Revised for Interfaces:
   Actor-Oriented Design in LabVIEW
2. Revision Coming in Q3 2022:
   Object-Oriented Design and Programming in LabVIEW

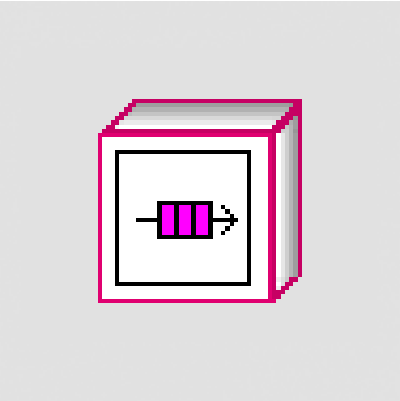
On-Demand Training: https://learn.ni.com/

Background Documentation:
LabVIEW Interfaces: The Decisions Behind the Design

More content can be found in LabVIEW Help and in customer presentations found online!

Any Questions?

One Final Personal Note...



Aristos Queue