IVI Driver Terminology

Purpose: Describe IVI driver nomenclature and using functions and attributes.

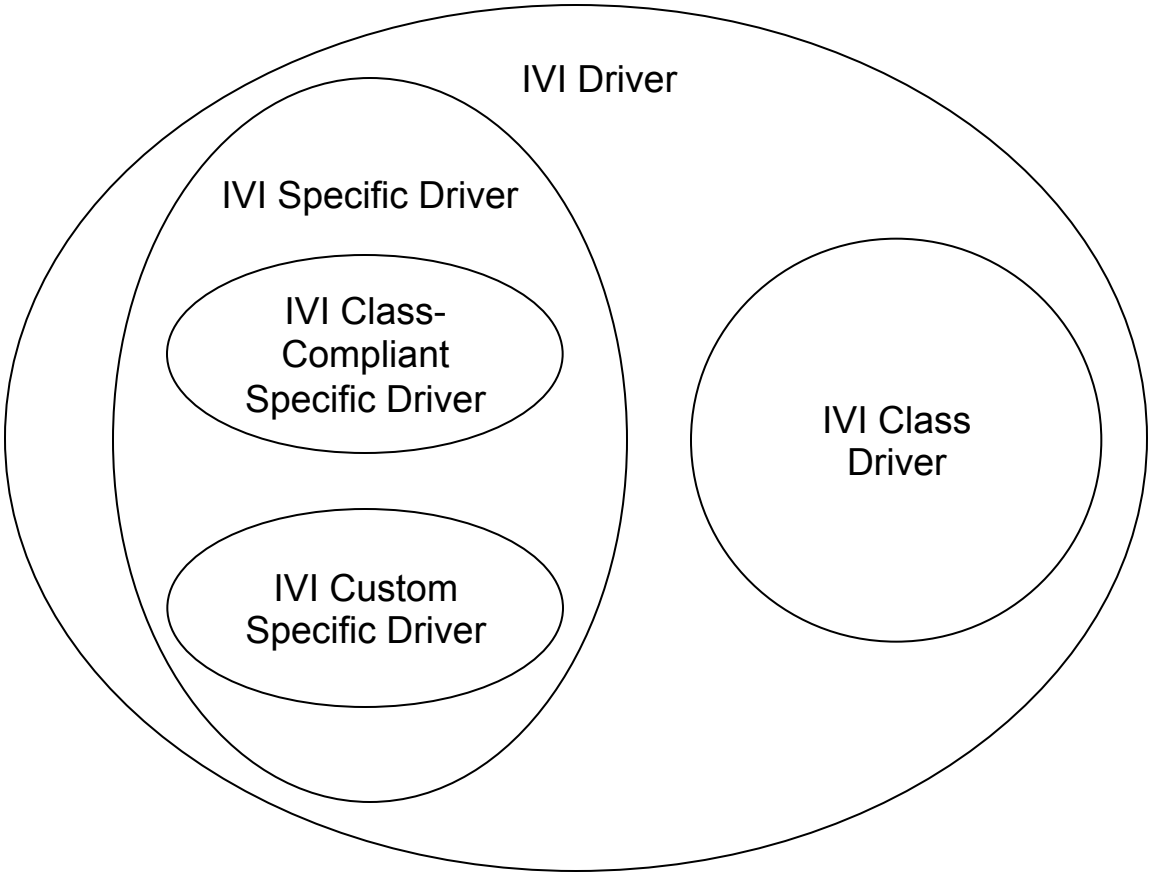
Topics:

- Describing Drivers
- Functions and attributes

Formal IVI Driver Terminology

- IVI Driver
  - Implements IVI-3.2: Inherent Capabilities Specification
  - Complies with all of the architecture specifications
  - May or may not comply with an instrument class
  - Is either an IVI specific driver or an IVI class driver
- IVI Specific Driver
  - Written for a particular instrument
- IVI Class-compliant Specific Driver
  - IVI specific driver that complies with one (or more) of the IVI-defined class specifications
  - Used when hardware independence is desired
- IVI Custom Specific Driver
  - IVI specific driver that is not compliant with one of the IVI-defined class specifications
  - Not interchangeable
- IVI Class Driver
  - IVI driver needed only for interchangeability in IVI-C environments
  - Class may be IVI-defined or customer-define

Types of IVI Drivers



IVI Driver Terminology in Practice

- Formal terminology is confusing
  - Overly generic because of need to describe IVI-C, IVI-COM and IVI.NET drivers
  - Much simpler in practice
- For IVI-COM and .NET:
  - There is no class-driver (more on this later), so the distinction between “class driver” and “specific driver” is not applicable
  - Only one software component (the specific driver or simply, the “driver”)
    - Implements instrument-specific interfaces and (optionally) class-compliant interfaces
- For IVI-C:
  - Class driver required for interchangeability
    - Delegates function calls to the specific driver
    - National Instruments is the only supplier (free download)
  - Vendors provide a “specific” driver as a separate component

Three API Standards in IVI

- IVI class and architecture specifications layout rules for all APIs
  - IVI-COM
  - IVI-C
  - IVI.NET
- To choose which one you need, consider:
  - Development environment (C, .NET, LabVIEW, MATLAB, …)
- IVI specs also define and specify “driver wrappers”
  - Specifies how to support multiple types of drivers (C, COM, .NET)
  - IVI-COM on top of IVI-C (unusual)
  - IVI-C on top of IVI-COM (common)

Functions and Attributes

- IVI uses the generic terms functions, attributes to refer to the elements of the API exported by an IVI driver
- Functions
  - Refers to standard COM methods in IVI-COM
  - Refers to C entry point functions in IVI-C
- Attributes
  - Properties in IVI-COM and IVI.NET
    - [propput] and [propget] used in COM IDL definitions
  - IVI-C uses attribute access functions
    - Functions for Get and Set of various data types

```
Ag34401_GetAttributeViReal64(vi, "", Ag34401_ATTR_RANGE, &range)
```

Attribute Values

- Attribute Values
  - Numeric or discrete
  - Discrete values represented by enums in IVI-COM
    - Example: IviScopeTriggerTypeEnum, with defined enum values
    - Enum defined in class-compliant type library provided by IVI
  - Discrete values represented by #defined values (macros) in IVI-C
    - Example: IVISCOPE_VAL_EDGE_TRIGGER
    - Values defined in header file (iviscope.h) supplied by specific or class driver

Demonstration: C# Hello World

- This demonstration shows
  - Open/Close a driver
  - Set property
  - Call a method
- Using IVI-C C++ Demo

Demonstration: C Hello World

- This demonstration shows
  - Creating a C project calling IVI Library
  - Opening a driver in C
  - Calling some functions to get a result
  - Setting and getting an attribute
- Using IVI-C CVI Demo