Benefits of Drivers

Purpose: Describe benefits of drivers in general and IVI drivers specifically.

Topics:

- Drivers and SCPI
- Structure of IVI specifications
- Specific IVI driver benefits



Comparing Drivers and SCPI

Programming with SCPI

viPrintf(vi, "MEAS:VOLT? %f, %f", range, resolution); viScanf(vi, &reading);

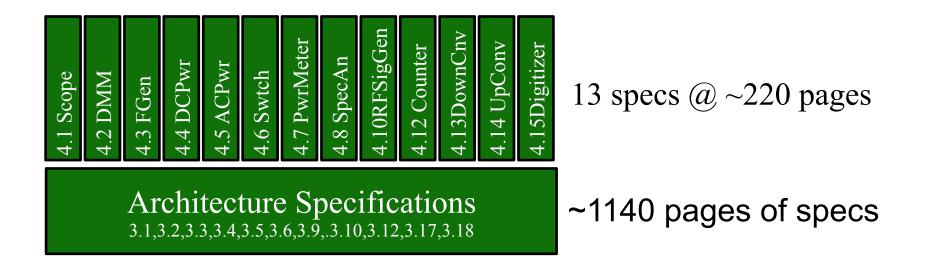
- Program deals with strings sent to/from the instrument
- Syntax errors caught by instrument when program is run
- Checking for errors requires another sequence to read error
- Simple model that requires no driver install
- Programming with IVI-C

Ag34410_MeasureDCVolt(vi, range, resolution, &reading);

- Program variables sent directly no chance for SCPI syntax errors
- Syntax errors caught by compiler or editor
- No performance impact due to string manipulation
- Uses debug tools and techniques the programmers knows

What are IVI Drivers – Really??

- Architecture specifications
- Instrument class specifications
- A library of shared software components



The IVI Architectures

IVI Provides: C, COM, and .NET

- C dll for environments that use DLLs
- COM Components for COM and .NET ADEs

Details in next section

• .NET Assemblies for .NET ADEs

Architectures make use of same class definition Architectures have specific rules for installation, style, etc.

IVI Shared Components

IVI Provides several common components to enable multivendor systems (more information in the final section)

- C Shared Components
- Floating Point Services
- IVI-COM Session Factory
- Configuration Server
- COM Type Libraries
- .NET PIAs
- .NET Shared Components

What is IVI Compliant -Really??

IVI Compliant

- Common behavior model
- Support for IVI Features
 - Simulation, IO, doc,
- Standard install
- Common API for common tasks
 - ~40 common functions
 - Simulation, Caching, Open, Close, Initialize, SW Trigger, Status check, Version

Consistent API

 Common organization, data types, naming

2013-04-13

Class Compliant

- Instrument Class API
- Custom API still available
 - Especially for capabilities beyond the class
- Simplifies exchanging instruments



Why IVI? – Simpler to use

Uniform way of doing common tasks

- Instantiation, initialization, shutdown
- Controlling driver features state caching, error query, simulation, etc.
- Configuration and installation
 - Fixed locations for binaries, source, headers, documentation, examples
 - Proper registry entries always made
 - Common protocol to open/close (standard I/O address is a big benefit)
 - Consistent solution for managing driver versions
- Standard mechanism for handling multi-channel devices
 - aka repeated capabilities in IVI parlance
- Standard error reporting

Why? – Common Features

Key Capabilities that simplify program development

- Syntactic Interchangeability
- Simulation
- Fine grained control through properties
- Usable in many ADEs
- Documentation of SCPI commands used by function
- DirectIO (drivers provide access to SCPI)
- Attributes for all parameters (fine grained control)
- VI-2014 Buildable source for message based instruments (SCPI)
- Tested using a IVI-defined process

Why IVI? – One Driver for any ADE

- IVI Drivers (C/COM/.NET) provide a first class experience in *nearly any ADE*
 - Visual Basic 6
 - Visual C++
 - Visual C# and Visual Basic.NET
 - VBA (Excel, Word, PowerPoint)
 - LabVIEW
 - LabWindows/CVI
 - MATLAB
 - Agilent VEE



IVI Registration Page

- IVI maintains a registration database
- IVI requires that drivers claiming compliance be registered
- For users:
 - List of drivers, supported instruments
 - Mechanism to address defects

