VeriStand Custom Device Design for Usability and Maintainability

Tristan Cowan
Simulation Systems Chief Engineer
tristan.cowan@bloomy.com
Bloomy Quick Facts

- NI Platinum Alliance Partner
  - Designated LabVIEW NXG Migration Services Partner in 2018
  - Awarded America’s Partner of the Year in 2017
  - Awarded “Outstanding Technical Resources” 2013 and 2014
  - 15 NI Certified LabVIEW Architects
  - 3 NI Certified LabVIEW Embedded Systems Developers
  - 3 NI Certified TestStand Architects
  - 3 NI Certified Training Centers
  - 9 NI Certified Professional Instructors
- Published “The LabVIEW Style Book” © 2007, Prentice Hall
- ISO 9001:2015 Certified Quality Management System
Agenda

- What Is a Custom Device and Why Do I Want One?
- Anatomy of a Custom Device
- Custom Device Configuration Strategies
- Custom Device Engine Strategies
- Summary
What Is a Custom Device and Why Do I Want One?
What Is VeriStand?

- Out-of-the-box solution for integrating simulation models with common NI hardware in a real-time system
VeriStand Architecture

What Is A Custom Device?

- Custom Devices are plugins which integrate specialized hardware into the VeriStand environment.
So You Want to Write a Custom Device

Custom Devices are great!

- Get anything into simulation environment
- No special knowledge required to use or develop

But…

- Code lives in the VeriStand framework
  - Limits options for test and debug
  - Constrains options for interacting with user and other entities in the system
  - Poor choices early on can complicate maintenance later
Anatomy of a Custom Device
Parts of a Custom Device

- Host and target have different responsibilities
  - Host: Configuration and user interface
  - Target: Runtime operations
- Custom device is built as two LLBs and an XML file
  - Host: Configuration LLB, custom device XML
  - Target: Engine LLB
- Custom device must have mechanism for organizing data obtained by the configuration side and passing it to the engine side
VeriStand Workflow

Configuration

Deployment

Runtime
Custom Device Design Patterns

- Focus on Inline Async pattern*
  - Configuration Library
    - ActionVIOnCompile
  - Shared Library
  - Engine Library

*Available as part of NI VeriStand Development Tools package
Configuration
ActionVIONCompile

- Sort and organize data channels and configuration information to pass down to engine.
Engine
Custom Device Configuration Strategies
Problem Statement

- Configuring a large system definition can be tedious and prone to human error

- Failed deployments due to incorrect configuration cost time, money, and user satisfaction
Strategies for Better Custom Device Configuration

- Strategy 1: Document Upfront
- Strategy 2: Control Tree Behavior
- Strategy 3: Place Smart Checks on User Behavior
Strategy 1: Document Up Front

- Feature help files directly in custom device
- Allow no implicit limits – clearly state ranges and out-of-range behavior
- Pay attention to UX
  - Make required information easy to find and well-organized
  - Take as much burden off the user as possible
Strategy 2: Control Tree Behavior

- Use XML tags to control page order and naming
  - Allow the user to rename pages where it makes sense
  - Sort pages in the way that makes them easiest to find

- Do not statically populate channels
  - Allow user to populate only channels they need

- Consider an import/export utility
  - Make configuration files human-readable (and scriptable)
Strategy 3: Smart User Behavior Checks

- Use XML tags to limit what a user can delete or rename
  - Make sure XML rules and configuration rules match

- Check for and prominently warn for obvious configuration errors.
  - Use Action VIs to your advantage

```xml
<XML tags>
  <ActionVIModuleDelete type="Path"/>
  <ActionVIModuleCopy type="Path"/>
  <ActionVIModuleSave type="Path"/>
  <ActionVIModulePaste type="Path"/>
  <ActionVIModuleDelete type="Path"/>
  <ActionVIModuleCopy type="Path"/>
  <ActionVIModuleSave type="Path"/>
  <ActionVIModulePaste type="Path"/>
</XML tags>
```
Custom Device Engine Strategies
Problem Statement

- Deployed systems are often really big and really complicated

- Limited error-reporting and debugging options can make errors difficult to troubleshoot
Strategies for Better Custom Device Engines

- Strategy 1: Modularize Code
- Strategy 2: Design for Robustness
- Strategy 3: Plan for Scale
- Strategy 4: Plan for Debug
Strategy 1: Modularize Code

- Consider a wrapper class around your core hardware API
  - If developing your own IP, test core hardware interaction outside VeriStand
  - Manage and organize information relevant only to VeriStand
  - Limit coupling to VeriStand
Strategy 2: Design for Robustness

- Find and sort channels by GUID

- Be aware of the namespacing implications of the Engine LLB
  - Dynamic dispatch and class overrides will be affected
  - PPLs can get around this, but complicate deployment and require additional maintenance over the long term

- Be aware of initialization and shutdown behavior
  - Consider executing these in the inline engine rather than the async engine

- Consider version-locking deployments
Strategy 3: Plan for Scale

- Make async loop rate a parameter rather than a constant
- Only read/write channels when something changes
- Segregate non-deterministic operations
- Plan for multiple instances of your custom device
Strategy 4: Plan for Debug

- What kind of information should always be reported?
  - Print critical errors and essential status information to console – but sparingly!

- Make error reports meaningful
  - Identify custom device and hardware and report detailed error info

- Consider a debug mode

- Test outside VeriStand where possible

- For complex custom devices, consider custom workspace tools for testing
Summary
Summary

- Custom devices are a powerful tool for integrating custom equipment and protocols into simulation systems.

- With careful planning, they can provide a clean user experience and robust and extensible deployed systems.
Questions?

Tristan Cowan
tristan.cowan@bloomy.com
(860) 607-2046

Stop by the Bloomy pavilion at **Booth #123**

Follow our blog at [www.bloomy.com/support/blog](http://www.bloomy.com/support/blog)
Before you go, take the survey.
Stay Connected During and After NIWeek

- ni.com/niweekcommunity
- facebook.com/NationalInstruments
- twitter.com/niglobal
- youtube.com/nationalinstruments