The LabVIEW Style Templates

Peter Blume

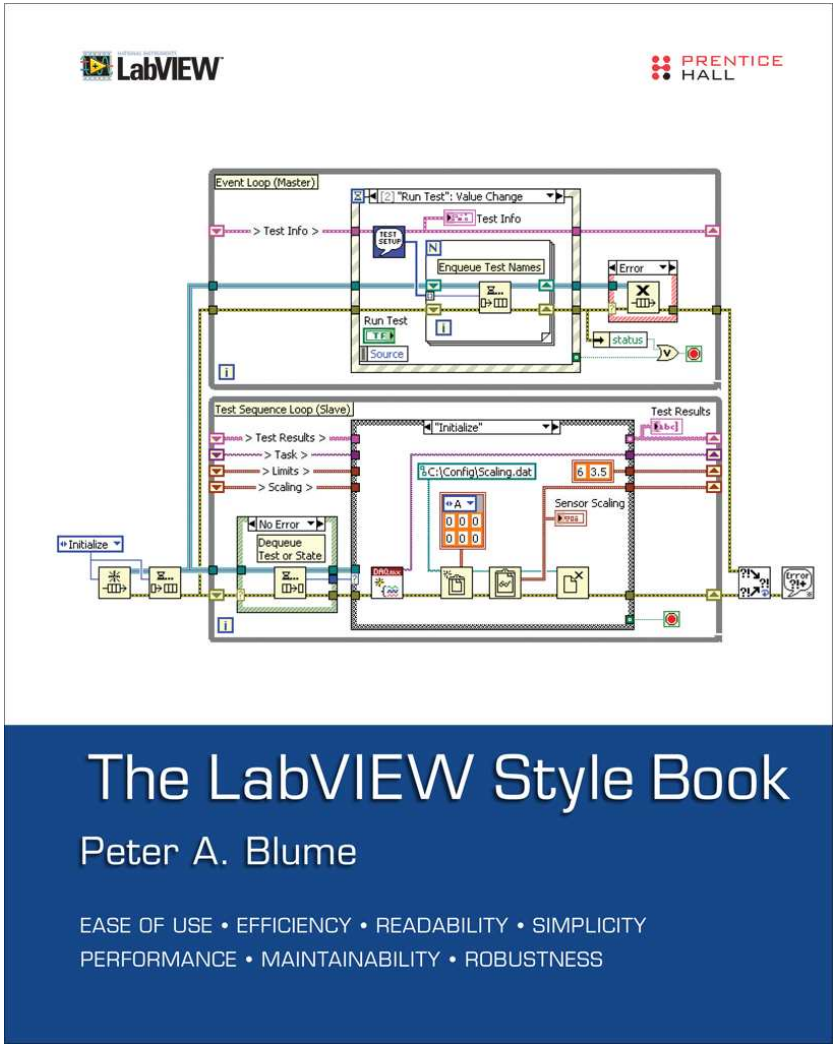
Bloomy Controls, Inc.

OverVIEW

- The LabVIEW Style Book
- ***Free*** Downloads
  - Specification template
  - Project directory template
  - Design pattern templates
  - Utility VIs
  - Style rules checklist
- Questions & feedback

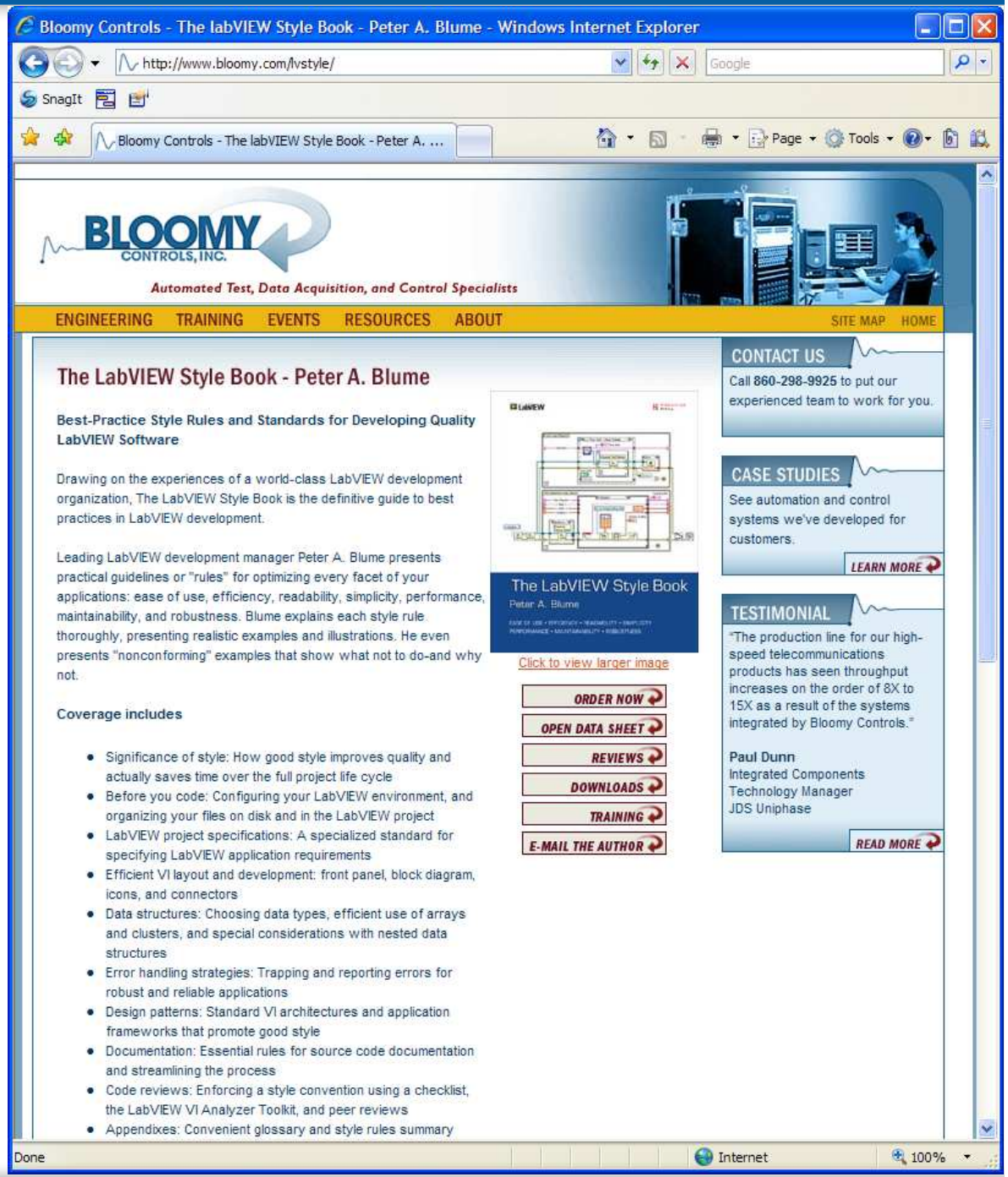
The LabVIEW Style Book

- Prentice Hall © 2007
- Best practices for developing quality LabVIEW applications
- Over 200 style rules
  - Ease of use
  - Efficiency
  - Readability
  - Simplicity
  - Performance
  - Maintainability
  - Reliability
- Drop off your card to win a copy!



Companion Web Site

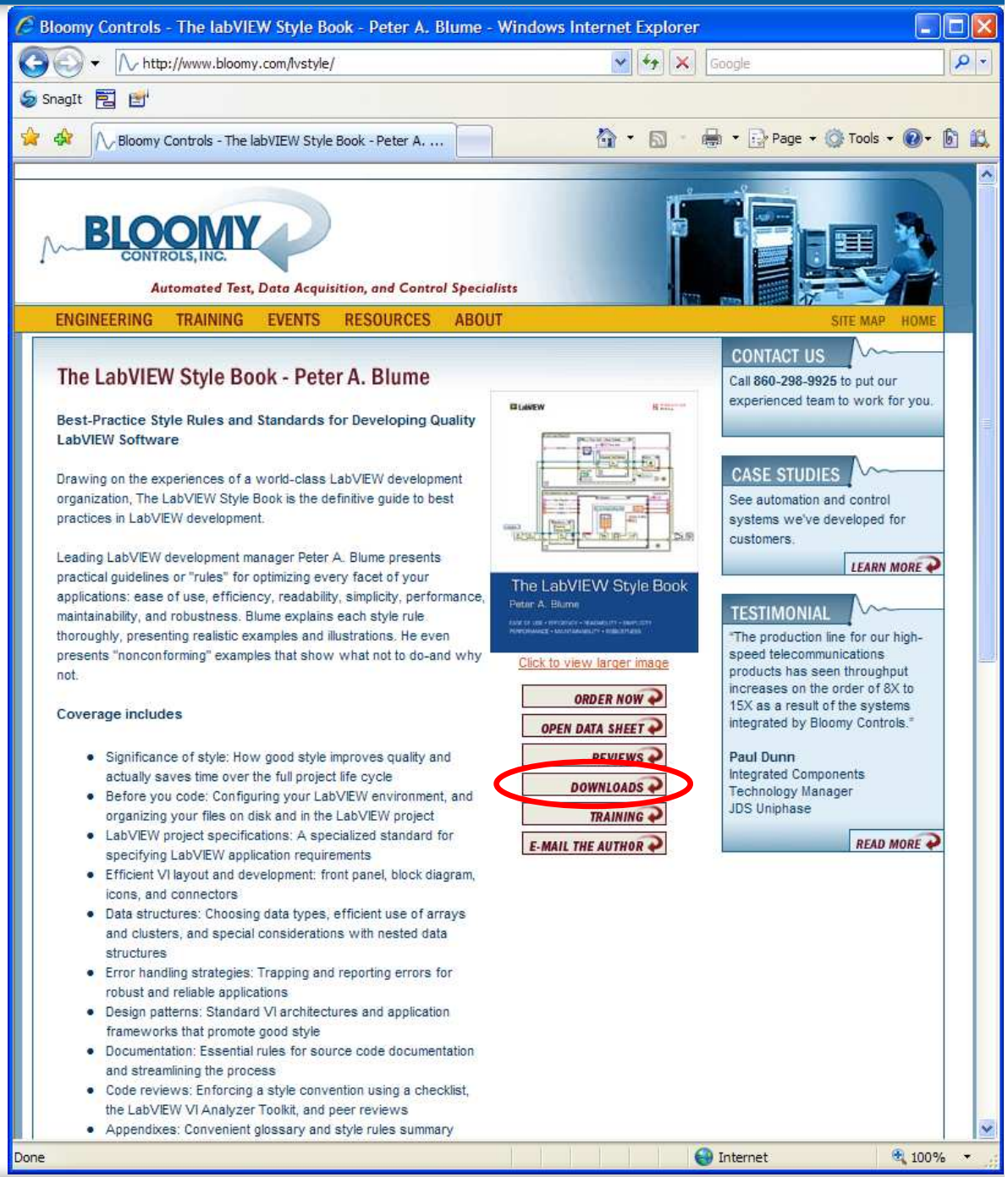
- www.bloomy.com/lvstyle/
- Reviews
- Data sheet
- Downloads
- Email the author
- Purchase the book



Downloads Page

Downloads

Download	Size	Description:	Date
Preface	138 KB	Preface	
Chapter 1	1,910 KB	Chapter 1 - The Significance of Style	05/04/2008
LabVIEW Project Specification Template	90 KB	Use this template for creating a LabVIEW project specification, discussed in Chapter 2.	
Project Directory Template	19 KB	A hierarchy of folders for storing source files on disk, populated with several templates, including a LabVIEW project. The directory and project templates contain folder hierarchies organized according to the illustrations in Chapter 2.	06/14/2007
Industrial Two Button Dialog VI	36 KB	A two-button dialog for industrial applications, as discussed in Chapter 3.	
Table 6-3: Controls and Data Types	50 KB	A spreadsheet containing the LabVIEW controls and supported data types, as shown in Table 6-3, without page breaks. This was contributed by reader Mark Shepard of North Bennington, VT.	06/07/2007
Merge Multiple Errors VI	34 KB	This VI for merges scalar error clusters, as discussed in Chapter 7.	
Clear Error All or Specified VI	21 KB	A utility that clears the specified Code to Clear, as discussed in Chapter 7.	
Design Pattern Templates	268 KB	This download contains several design pattern templates and supporting utility VIs. The templates include the Immediate SubVI (8.1.1), Continuous Loop (8.1.3), Event-Handling Loop (8.1.4), Classic State Machine (8.2.1), Queued State Machine (8.2.2), and Event-Driven State Machine (8.2.3). The templates and utilities are accessed via a LabVIEW 8 style Project Library.	04/20/2008
VI Analyzer Test Configuration	23 KB	VI Analyzer test configuration file contains custom test criteria and rankings according to the recommendations in Chapter 10.	06/07/2007
Style Rules Checklist	115 KB	A checklist for conducting code reviews, as discussed in Chapter 10.	



NATIONAL INSTRUMENTS

Specifications

- ***Theorem 2.1*** *Written specifications positively influence LabVIEW style*
- ***Rule 2.2*** *Write a requirements specification document before you begin coding*

LabVIEW

[Project
[Compar
Revision
[Date]

Introduction

The following pages contain a fun
[Company] in [City, ST]. The speci
[List the names, titles, and compa
referenced by or related to this sp

Objective

[Describe Company: What do the
objective that's driving this project
Describe the approximate budget

System Overview

[Describe the overall system in ve
subsystems that comprise the sys
system. Include an overall system
components that are part of the sy
components. Describe high-level
addresses the Company's challen

Hardware

[Describe the system hardware pl
modules, or DAQ devices. Describ
equipment racks that are required

Input/Output Li

[Insert a table containing
transducers and control d
lengthy, i.e. more than 25

Software

[Describe the software platform in
etc., 3rd party application(s), as we
of each significant application or t

Acquisition

[Describe the software
Specify the desired sam

Analysis

[Describe any on-line a
the equations and algor

Presentation

User Inter

[Describe the g
any ease-of-us
screens. Alway

Data Files

[Describe any d
ASCII, XML, M
or remote netw
logging rates a
applications tha
Server. Oracle,

Reports

[Describe any r
destination, suc
and data fields
if/when possibl

Connectivity

[Describe any network,
control via web browse
ActiveX, TCP, UDP, De

Priority Matrix

[Create a table containing an itemized list of software features, and priority level for each. Priorities should include Critical, High, Medium, and Low. This is essentially a subset of the Project Planning Worksheet, without the hours, rates, etc.]

Test Methodology

[Describe how the system will be tested. Will any in-house testing be performed prior to integration at the customer site? Describe any software and/or hardware that will be utilized or developed to simulate and/or test each feature. Specify any use cases that will be applied to test the integrated system. Describe the customer's responsibility for testing the system, if applicable.]

Appendix A: Glossary

[Define all terms, acronyms, and abbreviations used within the specification. List in alphabetical order.]

Appendix B: Input/Output Channel List

[For high channel count DAQ systems (i.e. > 25 channels), place the I/O list in this appendix instead of the hardware section of the main specification body.]

Appendix C: Sample Report

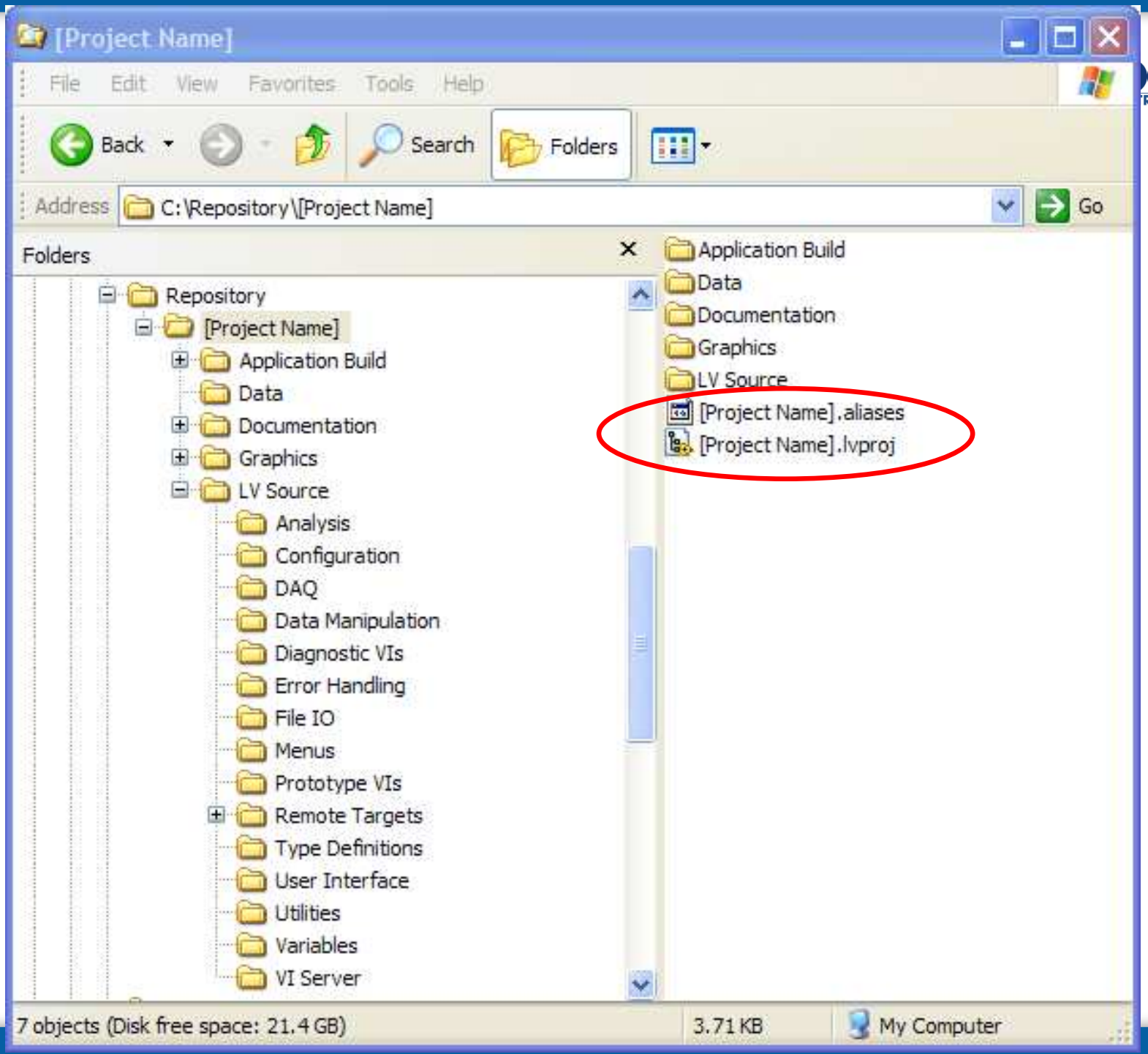
[Create a prototype of any report(s) that must be generated by the system.]

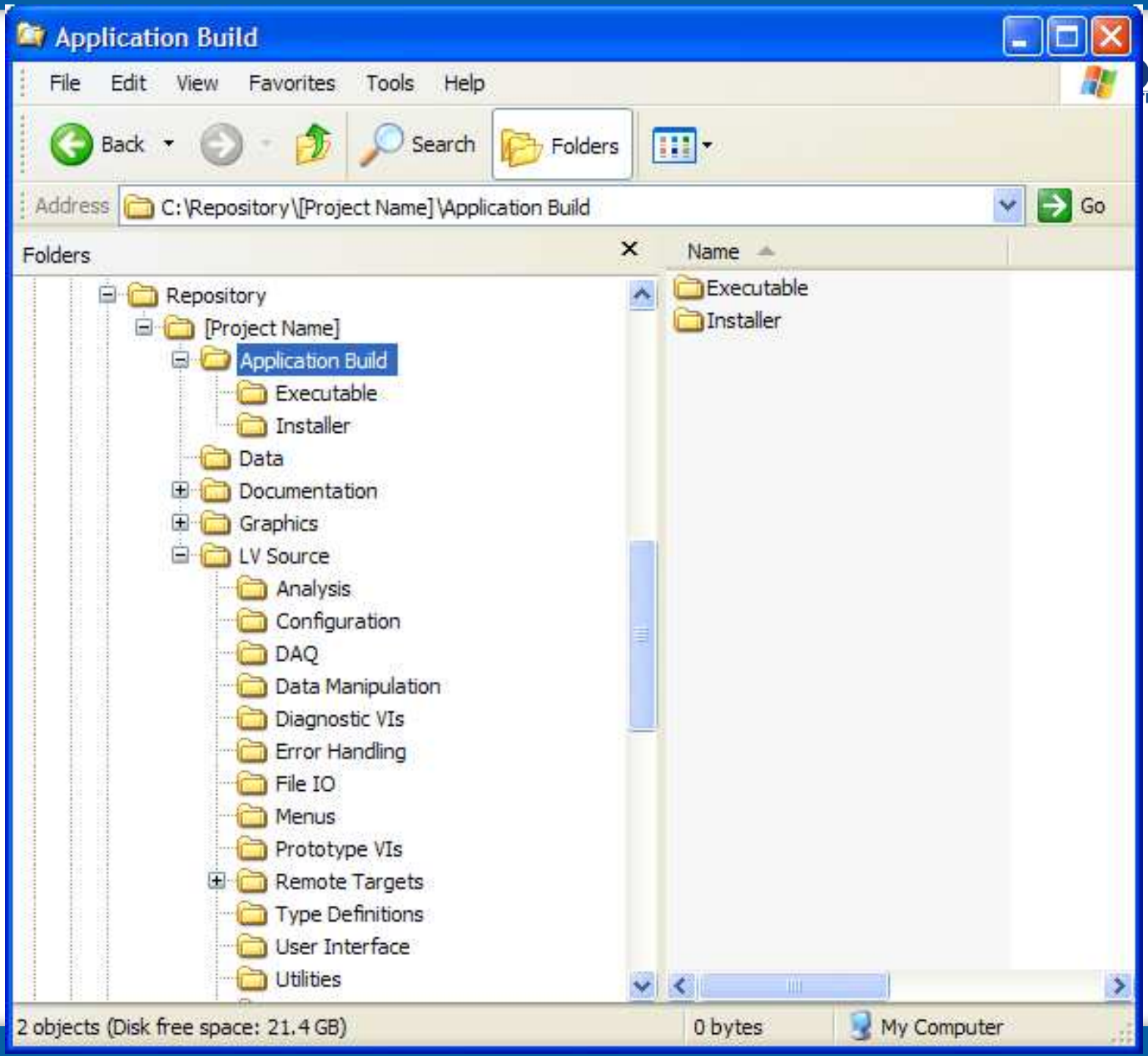
Appendix D: Product Specifications

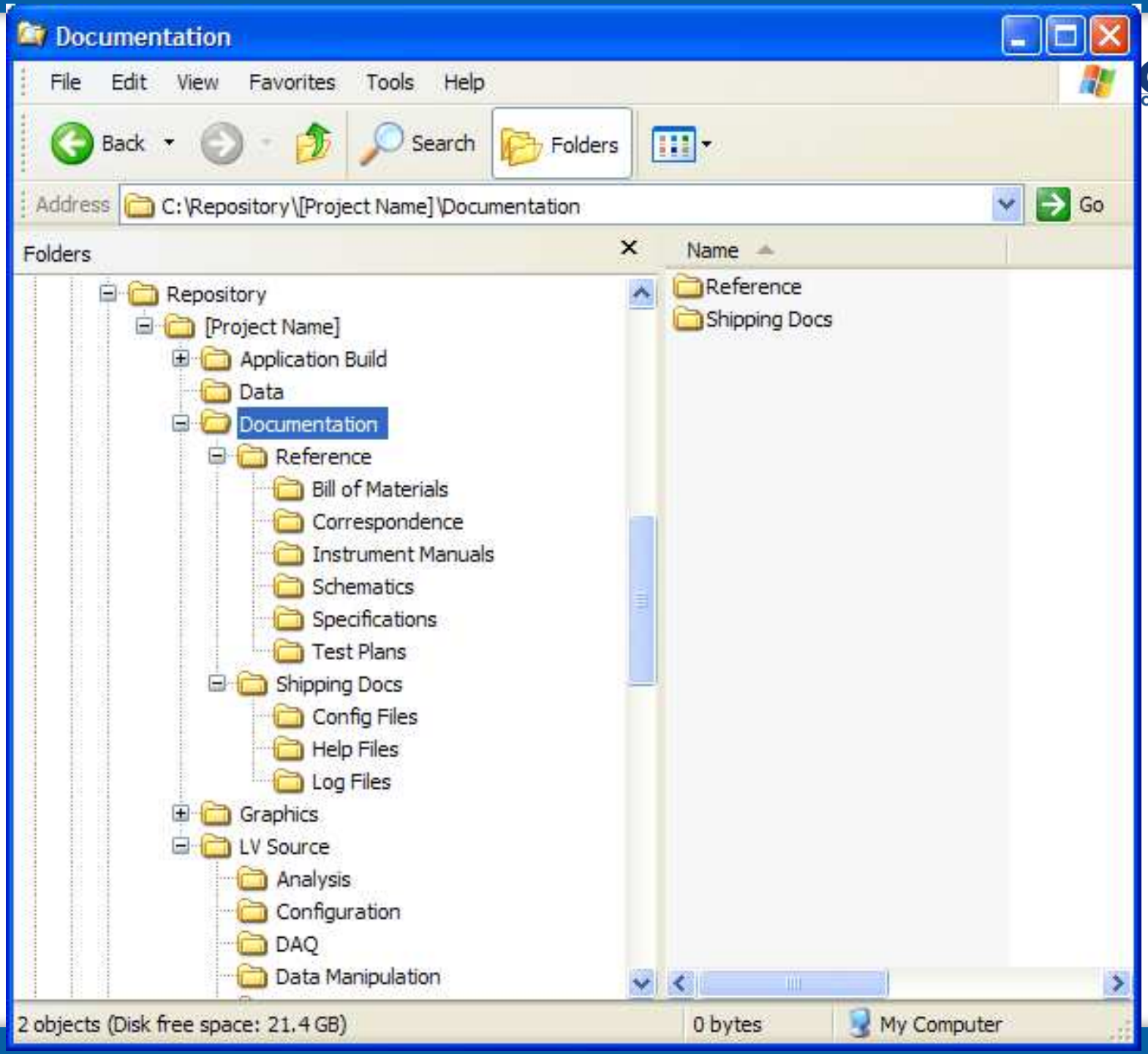
[Include the manufacturer's specifications of any 3rd party hardware and software products that are discussed within the specification.]

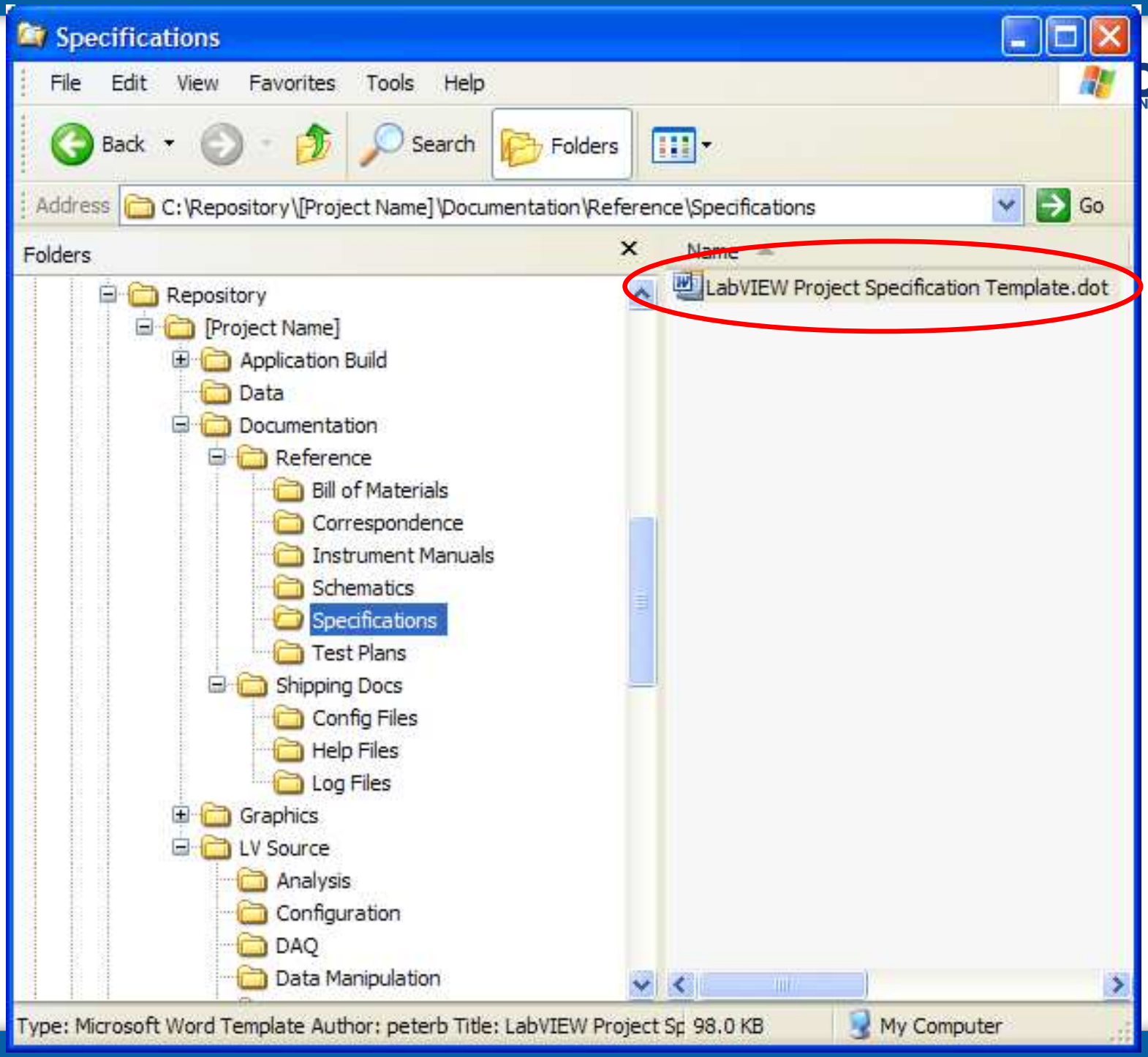
Project File Organization

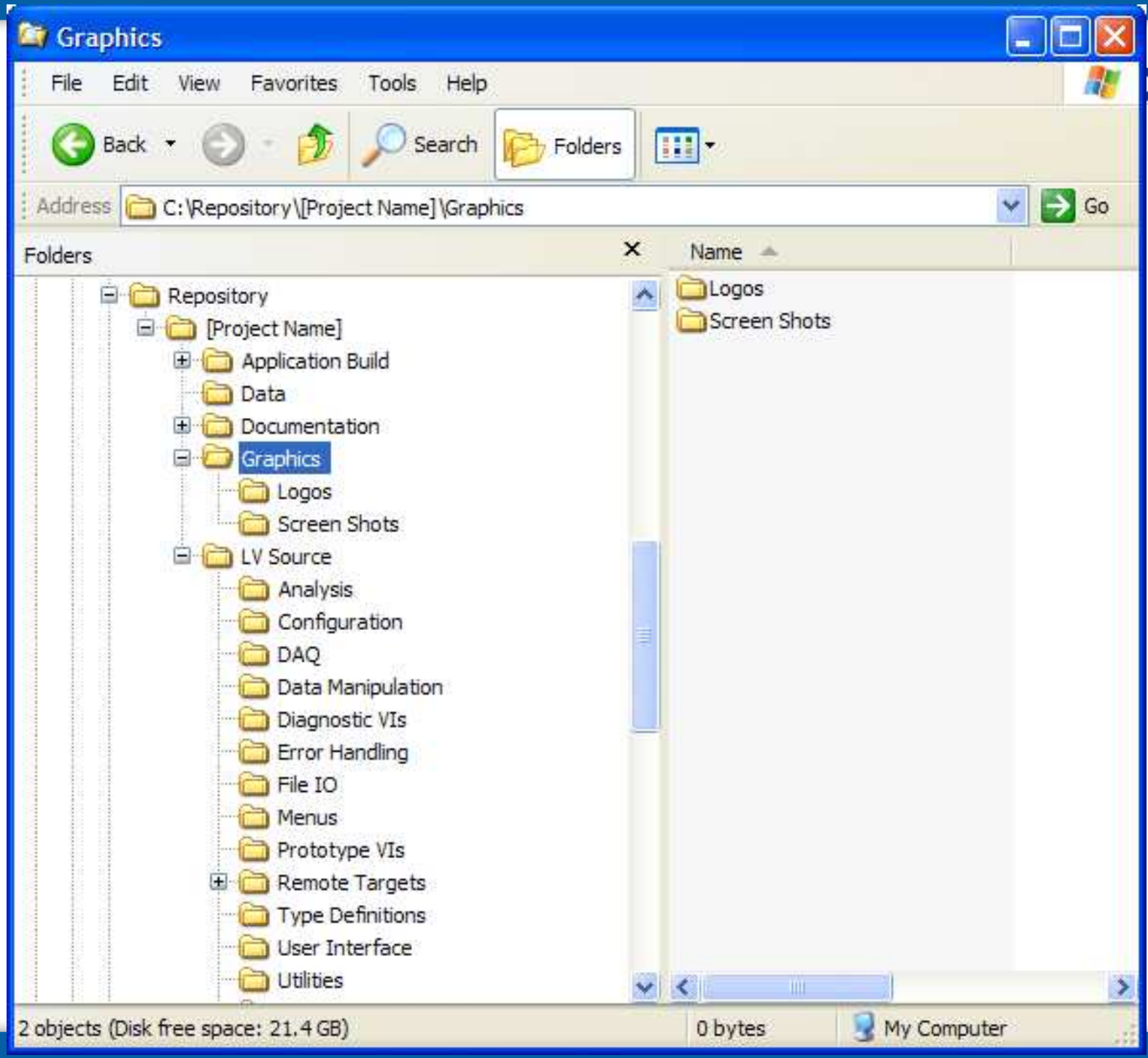
- ***Rule 2.8*** *Maintain an organized repository on disk*
- ***Rule 2.9*** *Create a LabVIEW source folder hierarchy that reflects your application's architecture*
- ***Rule 2.10*** *Create the folder hierarchy before you begin coding*

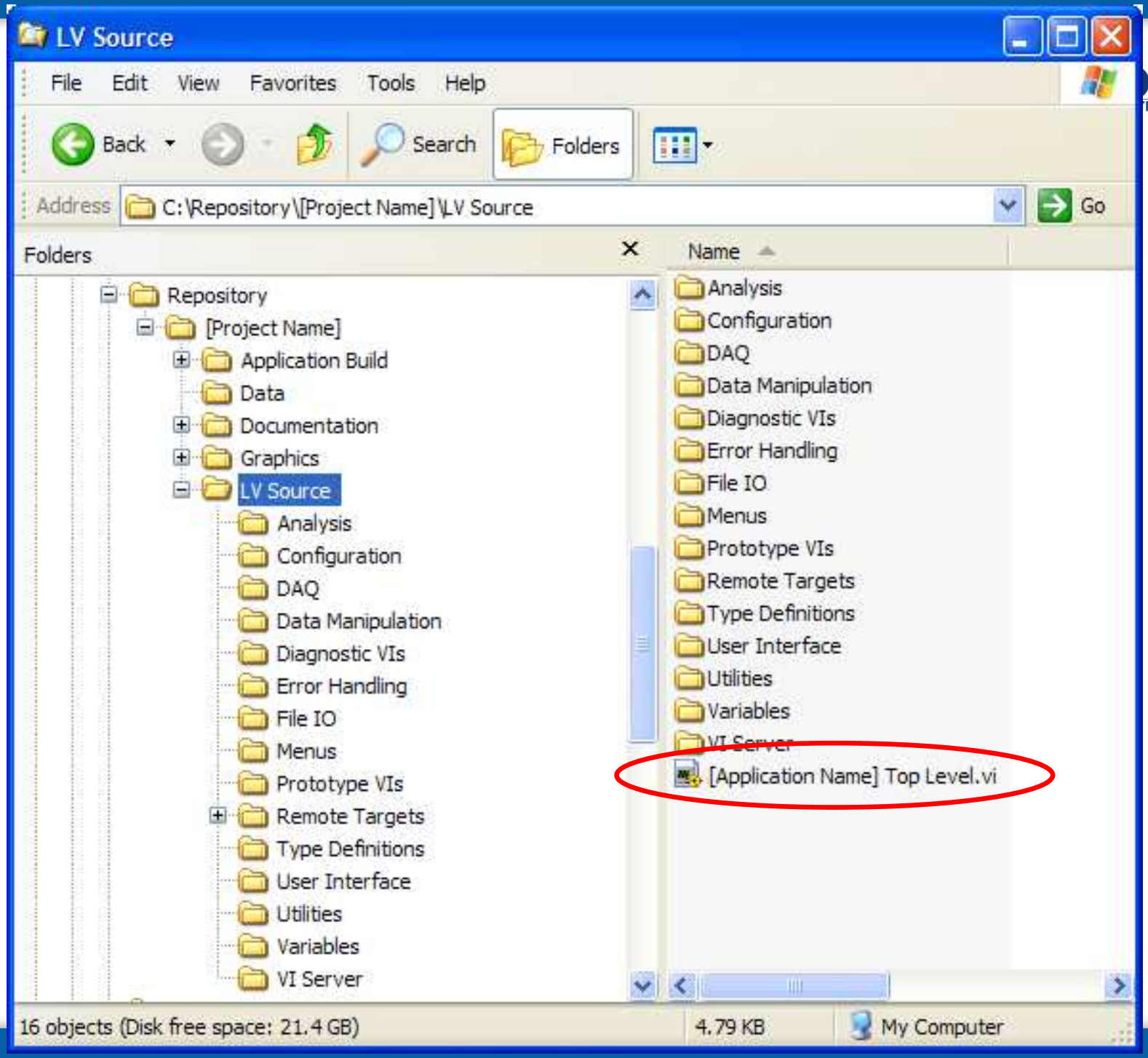

[Project Name]
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\Repository\[Project Name]
Go
Folders
Repository
[Project Name]
Application Build
Data
Documentation
Graphics
LV Source
Analysis
Configuration
DAQ
Data Manipulation
Diagnostic VIs
Error Handling
File IO
Menus
Prototype VIs
Remote Targets
Type Definitions
User Interface
Utilities
Variables
VI Server
Application Build
Data
Documentation
Graphics
LV Source
[Project Name].aliases
[Project Name].lvproj
7 objects (Disk free space: 21.4 GB)
3.71 KB
My Computer


Application Build
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Repository\[Project Name]\Application Build
Go
Folders
Name
Repository
[Project Name]
Application Build
Executable
Installer
Data
Documentation
Graphics
LV Source
Analysis
Configuration
DAQ
Data Manipulation
Diagnostic VIs
Error Handling
File IO
Menus
Prototype VIs
Remote Targets
Type Definitions
User Interface
Utilities
Executable
Installer
2 objects (Disk free space: 21.4 GB)
0 bytes
My Computer


Documentation
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Repository\[Project Name]\Documentation
Go
Folders
Name
Repository
[Project Name]
Application Build
Data
Documentation
Reference
Bill of Materials
Correspondence
Instrument Manuals
Schematics
Specifications
Test Plans
Shipping Docs
Config Files
Help Files
Log Files
Graphics
LV Source
Analysis
Configuration
DAQ
Data Manipulation
Reference
Shipping Docs
2 objects (Disk free space: 21.4 GB)
0 bytes
My Computer

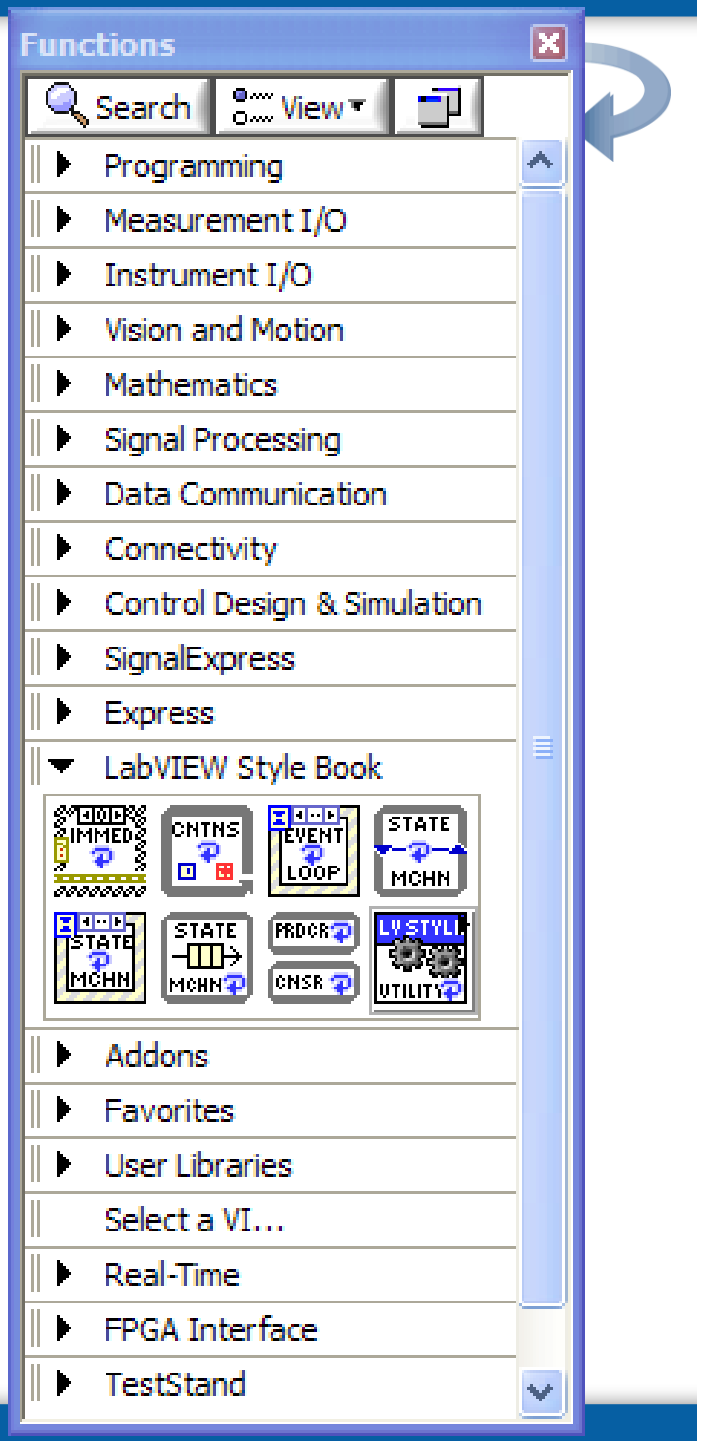

Specifications
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Repository\[Project Name]\Documentation\Reference\Specifications
Go
Folders
Name
LabVIEW Project Specification Template.dot
Repository
[Project Name]
Application Build
Data
Documentation
Reference
Bill of Materials
Correspondence
Instrument Manuals
Schematics
Specifications
Test Plans
Shipping Docs
Config Files
Help Files
Log Files
Graphics
LV Source
Analysis
Configuration
DAQ
Data Manipulation
Type: Microsoft Word Template Author: peterb Title: LabVIEW Project Sp 98.0 KB
My Computer


Graphics
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Repository\[Project Name]\Graphics
Go
Folders
Name
Repository
[Project Name]
Application Build
Data
Documentation
Graphics
Logos
Screen Shots
LV Source
Analysis
Configuration
DAQ
Data Manipulation
Diagnostic VIs
Error Handling
File IO
Menus
Prototype VIs
Remote Targets
Type Definitions
User Interface
Utilities
Logos
Screen Shots
2 objects (Disk free space: 21.4 GB)
0 bytes
My Computer


LV Source
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Repository\[Project Name]\LV Source
Go
Folders
Name
Repository
[Project Name]
Application Build
Data
Documentation
Graphics
LV Source
Analysis
Configuration
DAQ
Data Manipulation
Diagnostic VIs
Error Handling
File IO
Menus
Prototype VIs
Remote Targets
Type Definitions
User Interface
Utilities
Variables
VI Server
[Application Name] Top Level.vi
16 objects (Disk free space: 21.4 GB)
4.79 KB
My Computer

Design Pattern Templates

- Immediate SubVI
- Continuous Loop
- Event Handling Loop
- Classic State Machine
- Queued State Machine
- Event-Driven State Machine
- Producer-Consumer



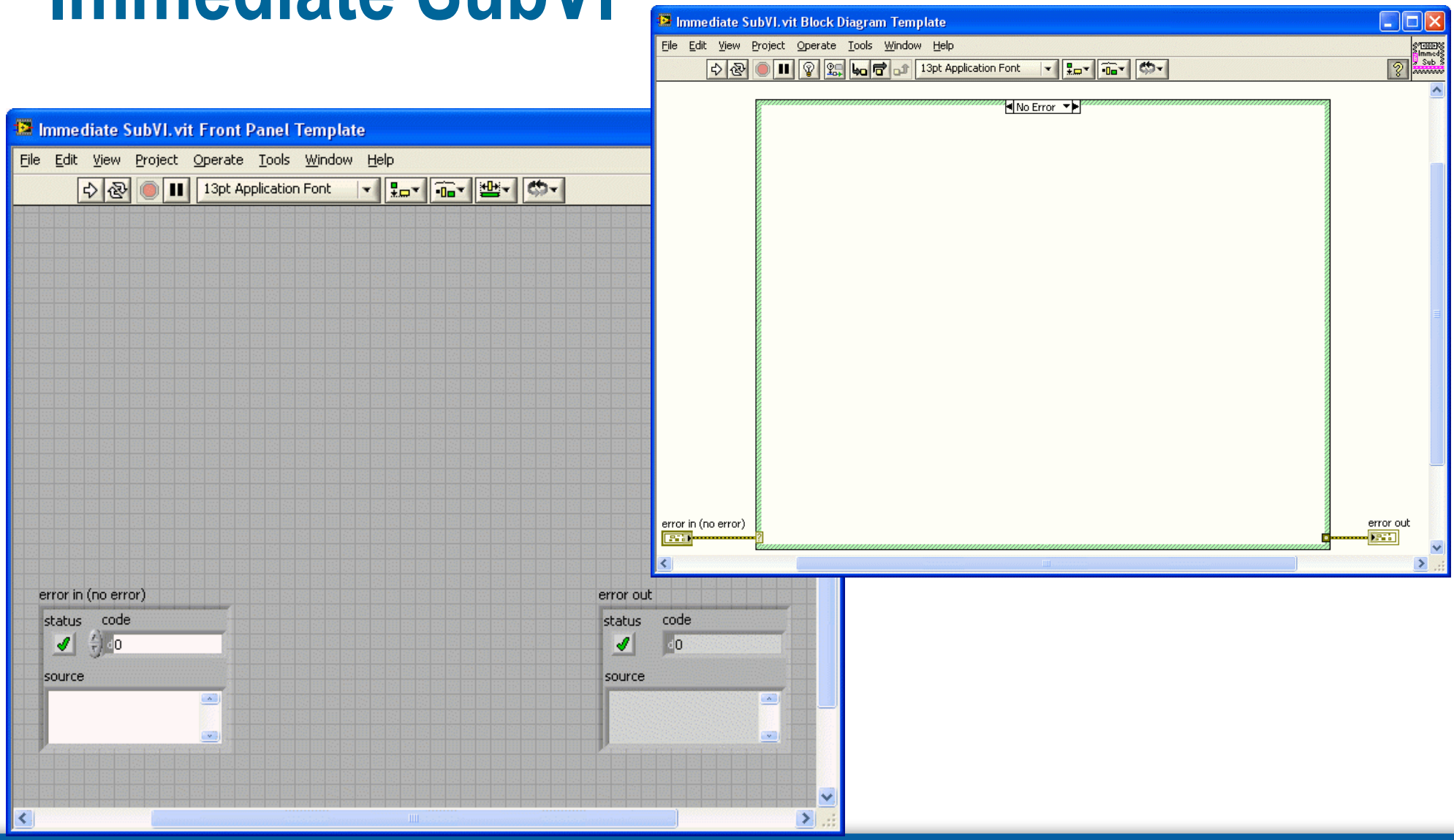
SubVI Error Handling

Rule 5.25 *Assign error clusters to bottom left and right terminals*

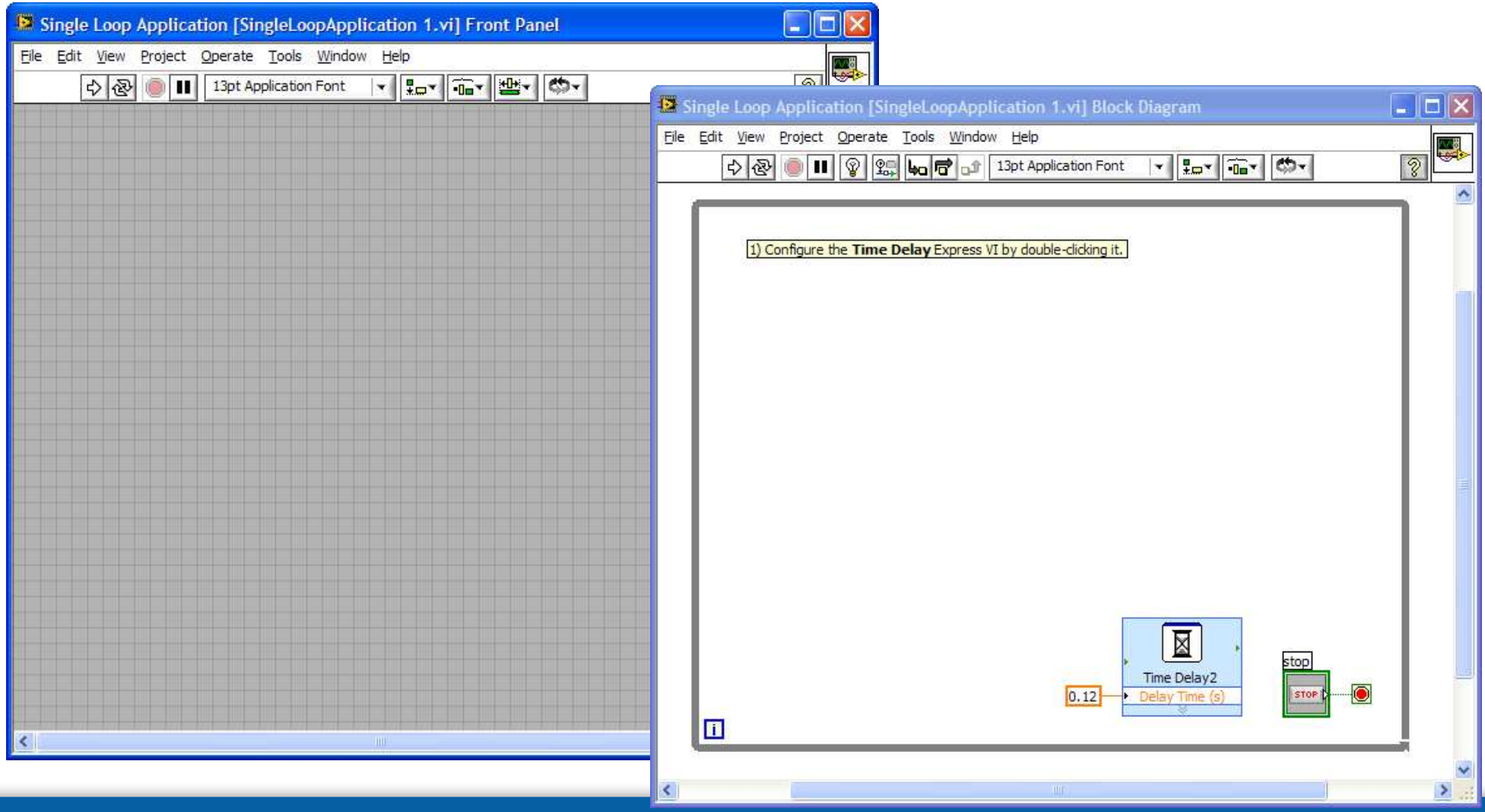
Rule 7.13 *Skip most subVI diagrams on error using an Error Case Structure*

Rule 7.15 *Use the SubVI with Error Handling template*

Immediate SubVI



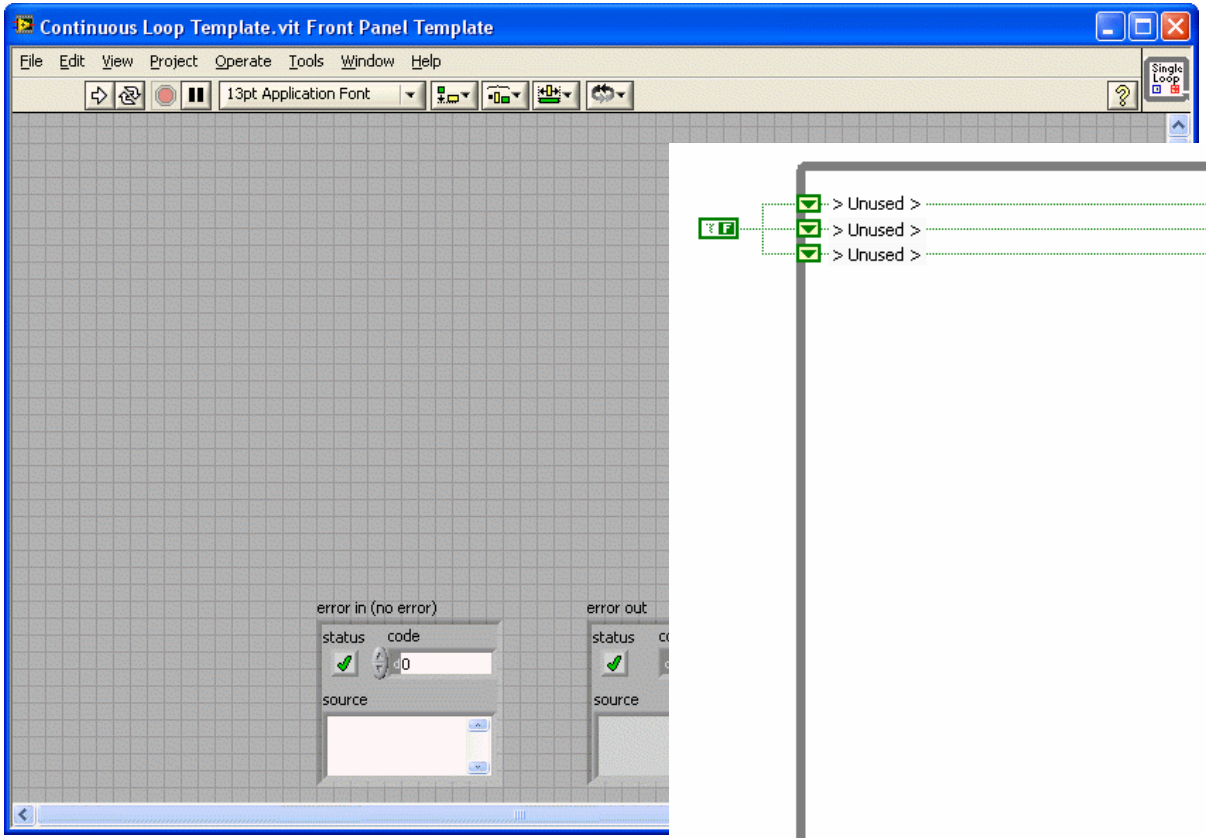
NI Single Loop Application

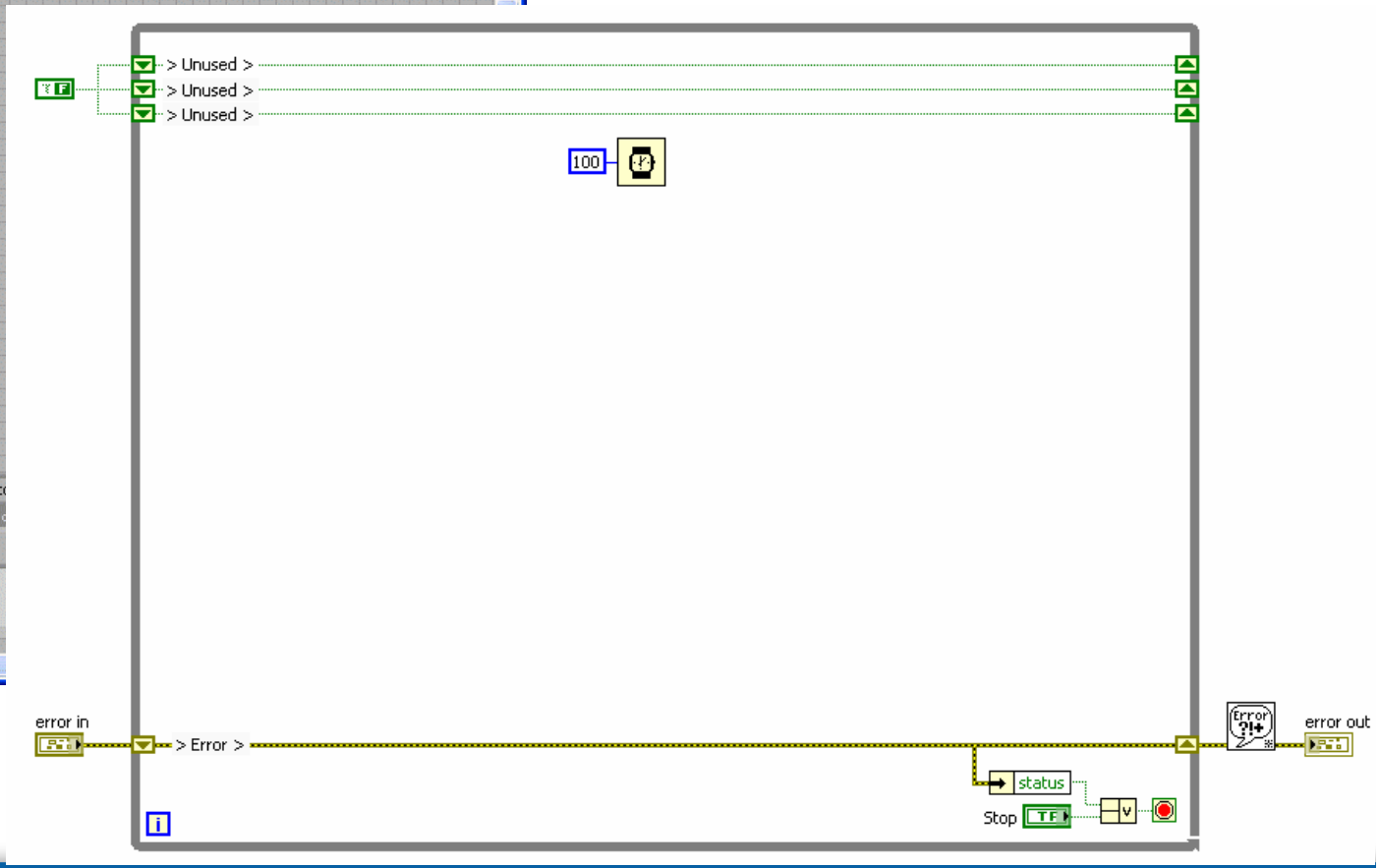


Continuous Loop Rules

- ***Rule 4.35*** *Use shift registers over local and global variables*
- ***Rule 4.36*** *Group most shift registers near the top of the loop*
- ***Rule 4.37*** *Label wires exiting the left shift register terminal*
- ***Rule 8.1*** *Use multiple criteria for the loop condition*
- ***Rule 8.3*** *Include a delay within continuous While Loops*

Continuous Loop

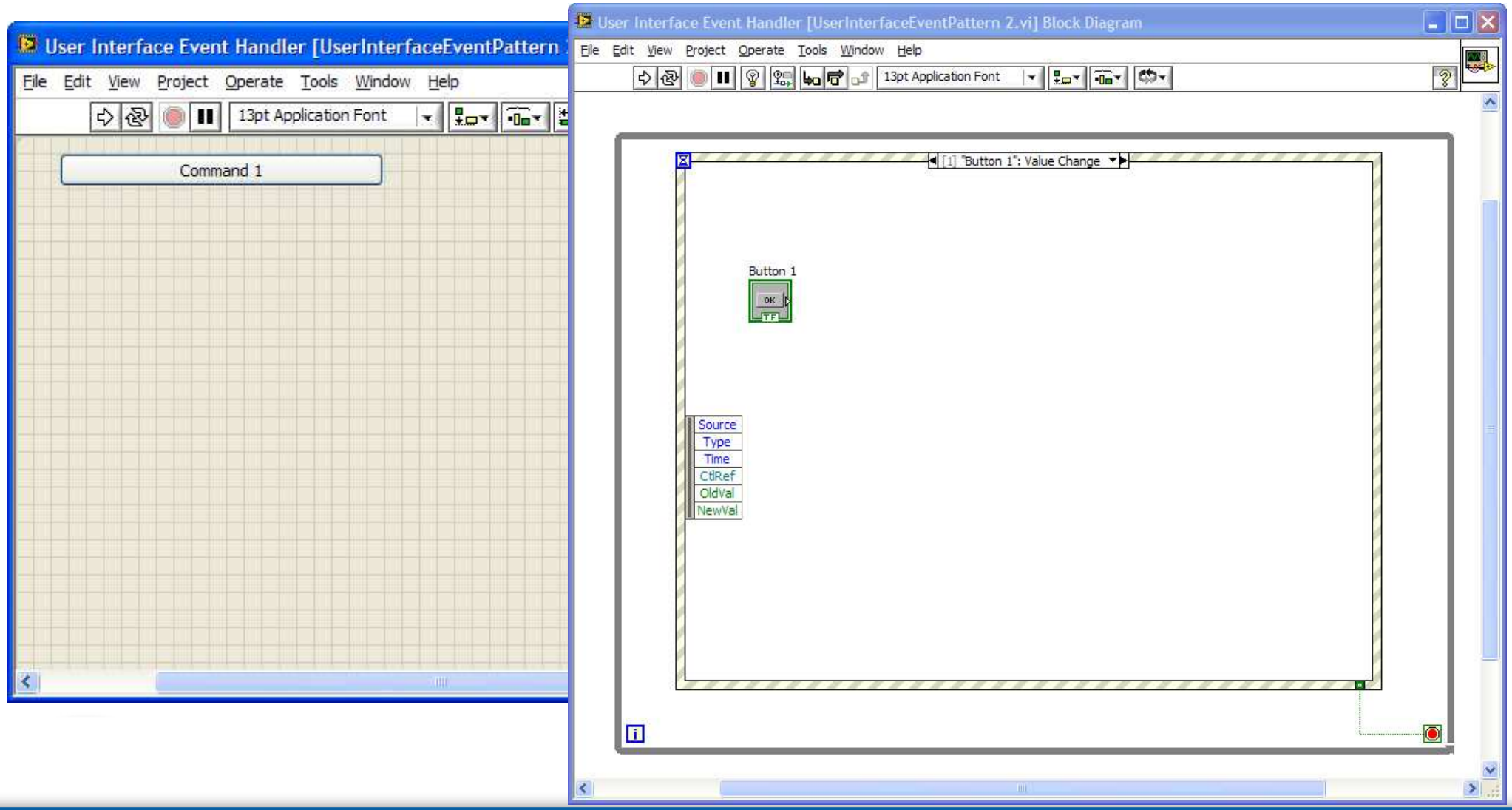




Limitations of Continuous Loop

- ***Rule 8.4*** *Avoid polling GUI objects*
- Use an Event structure

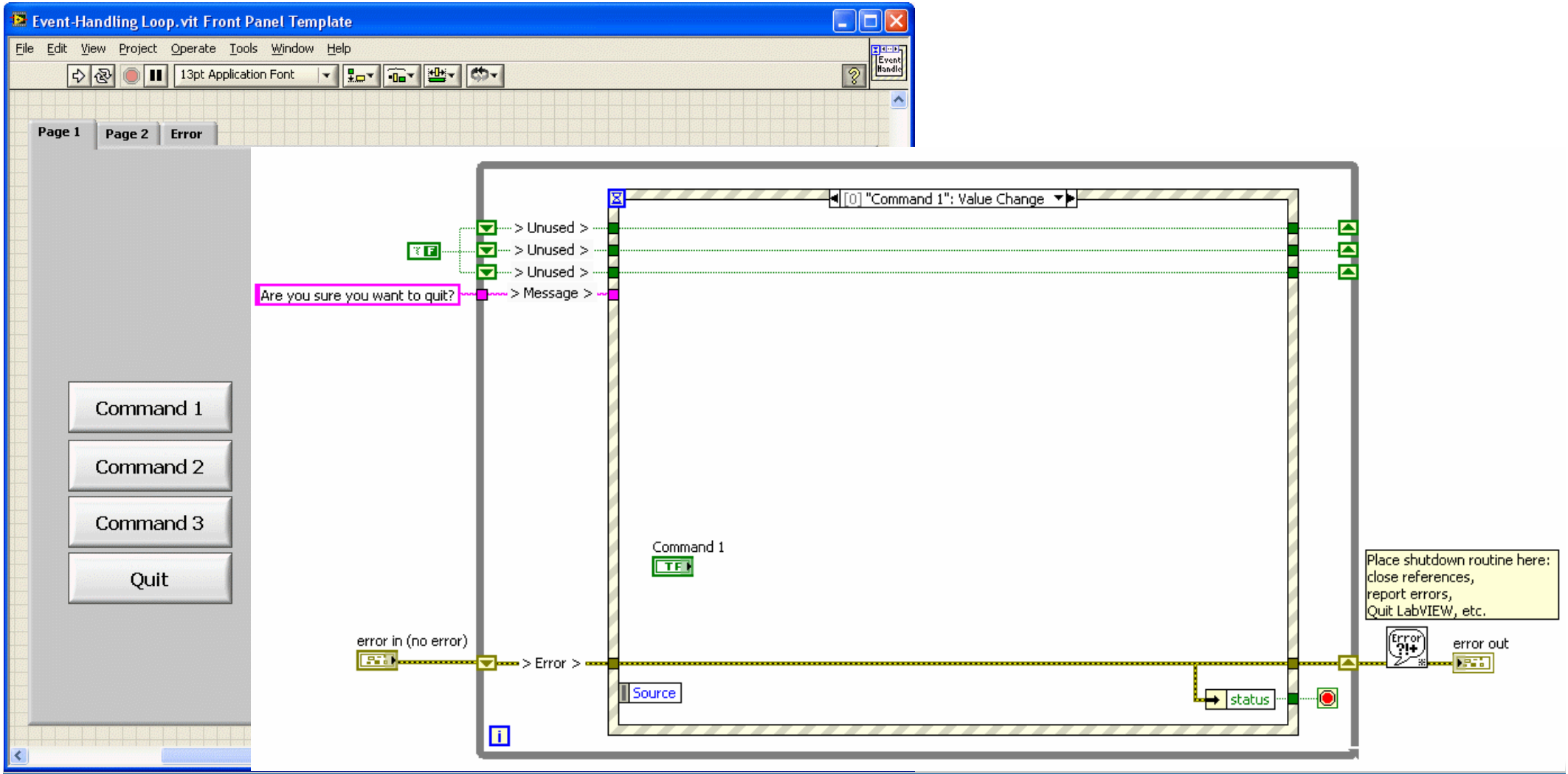
NI User Interface Event Handler



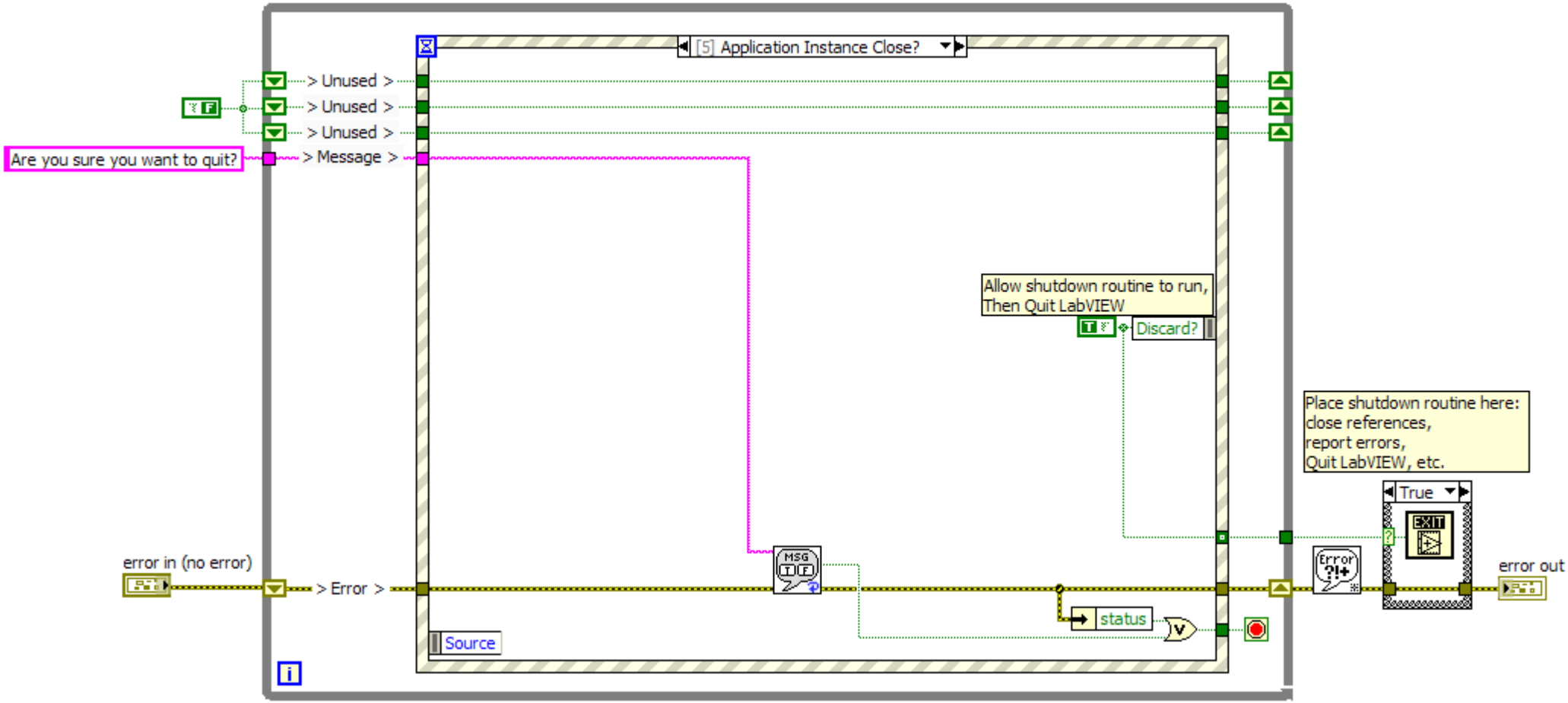
Event Structure Rules

- ***Rule 8.5*** *Use the Value Change event for most GUI controls*
- ***Rule 8.6*** *Place control terminals within their Value Change event case*
- ***Rule 8.7*** *Resize the Event Data Node to hide unused terminals*
- ***Rule 8.8*** *Avoid continuous timeout events*

Event-Handling Loop



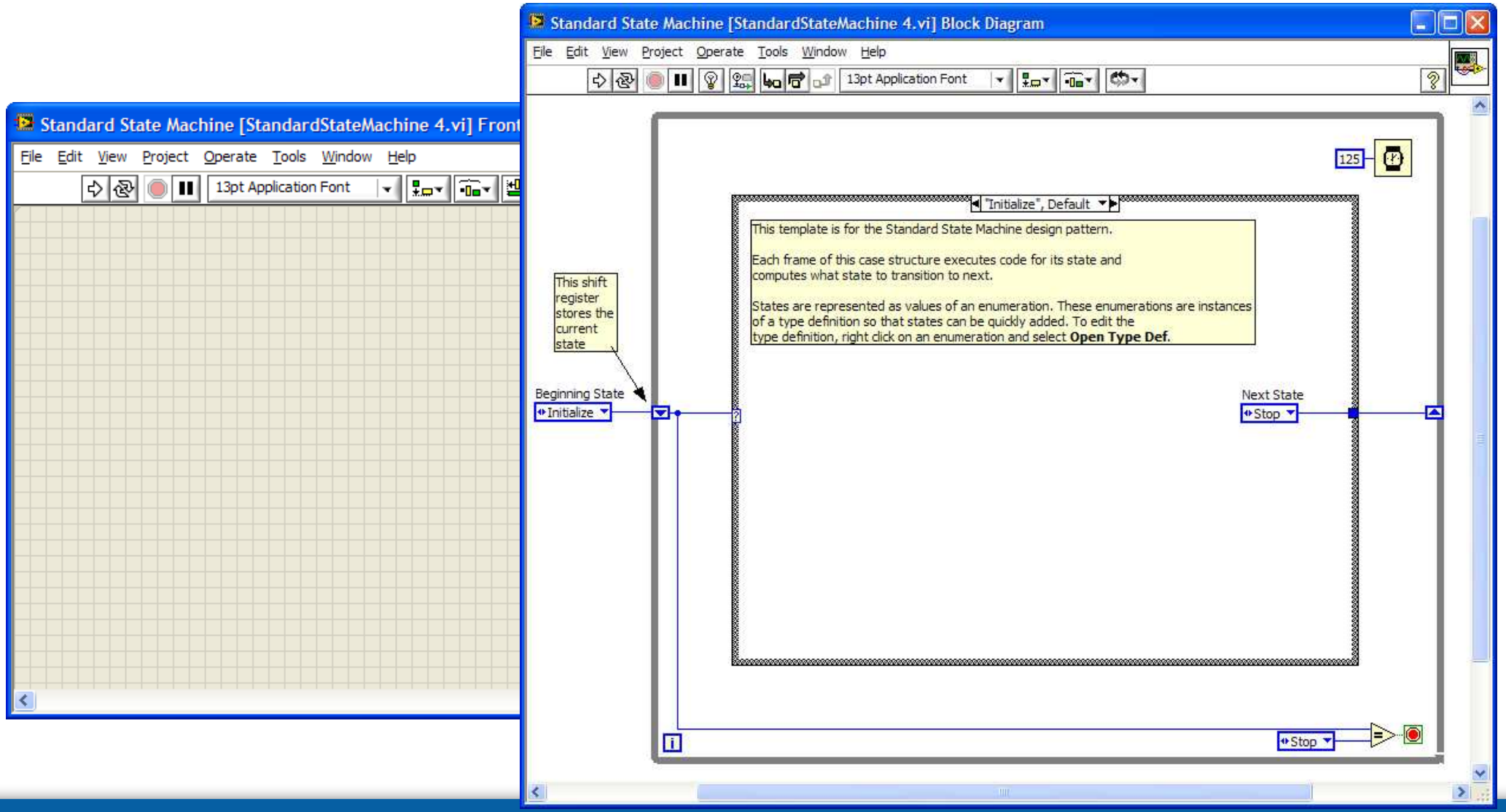
Event-Handling Loop - Shutdown



Limitations of Event-Handling Loop

- Only one subdiagram per GUI event
- ***Rule 8.9*** *Use a state machine design pattern in most VIs of medium or greater complexity*

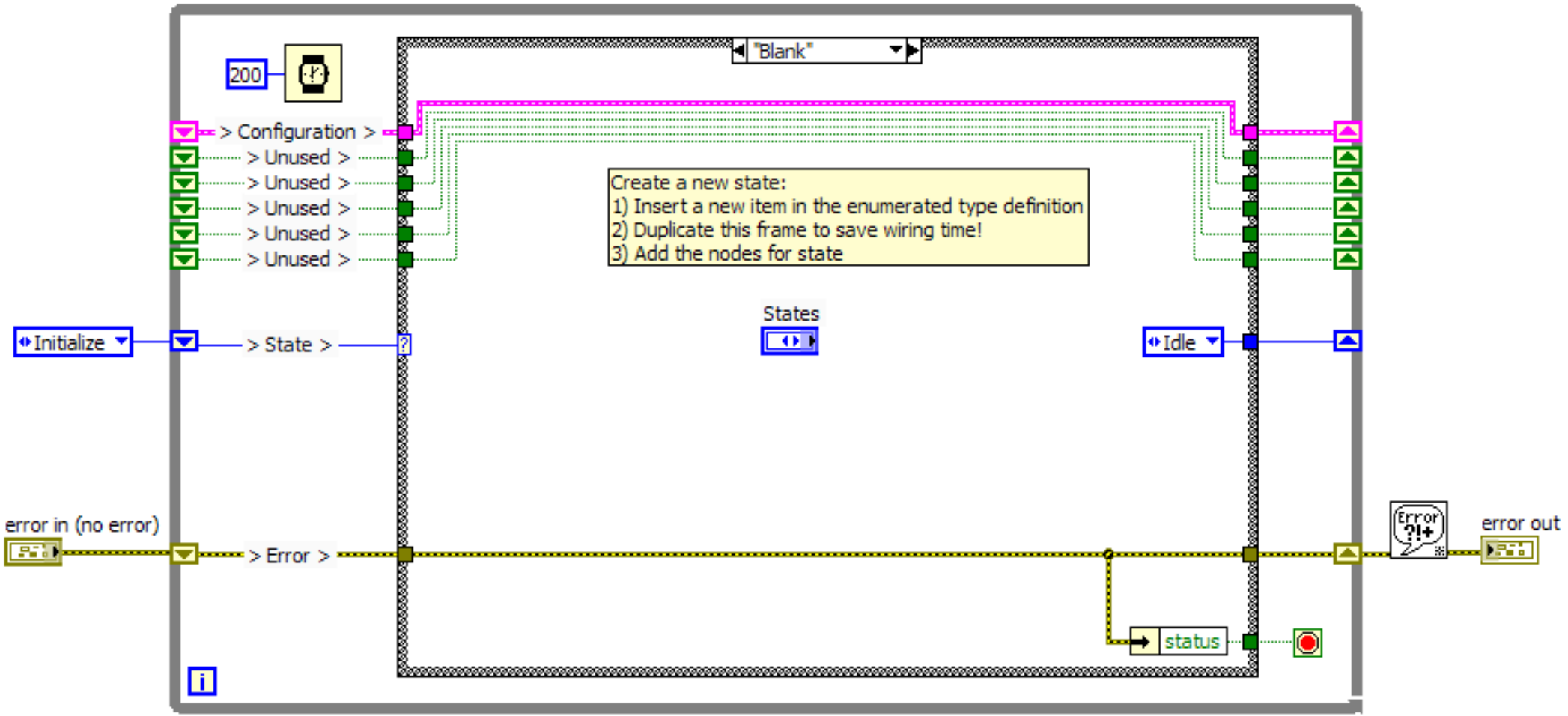
NI Standard State Machine



State Machine - Rules

- ***Rule 8.12*** *Use an enumerated type definition for the case selector*
- ***Rule 8.13*** *Minimize code external to the Case structure*
- ***Rule 8.14*** *Include states for Initialize, Idle, Shutdown, and Blank*

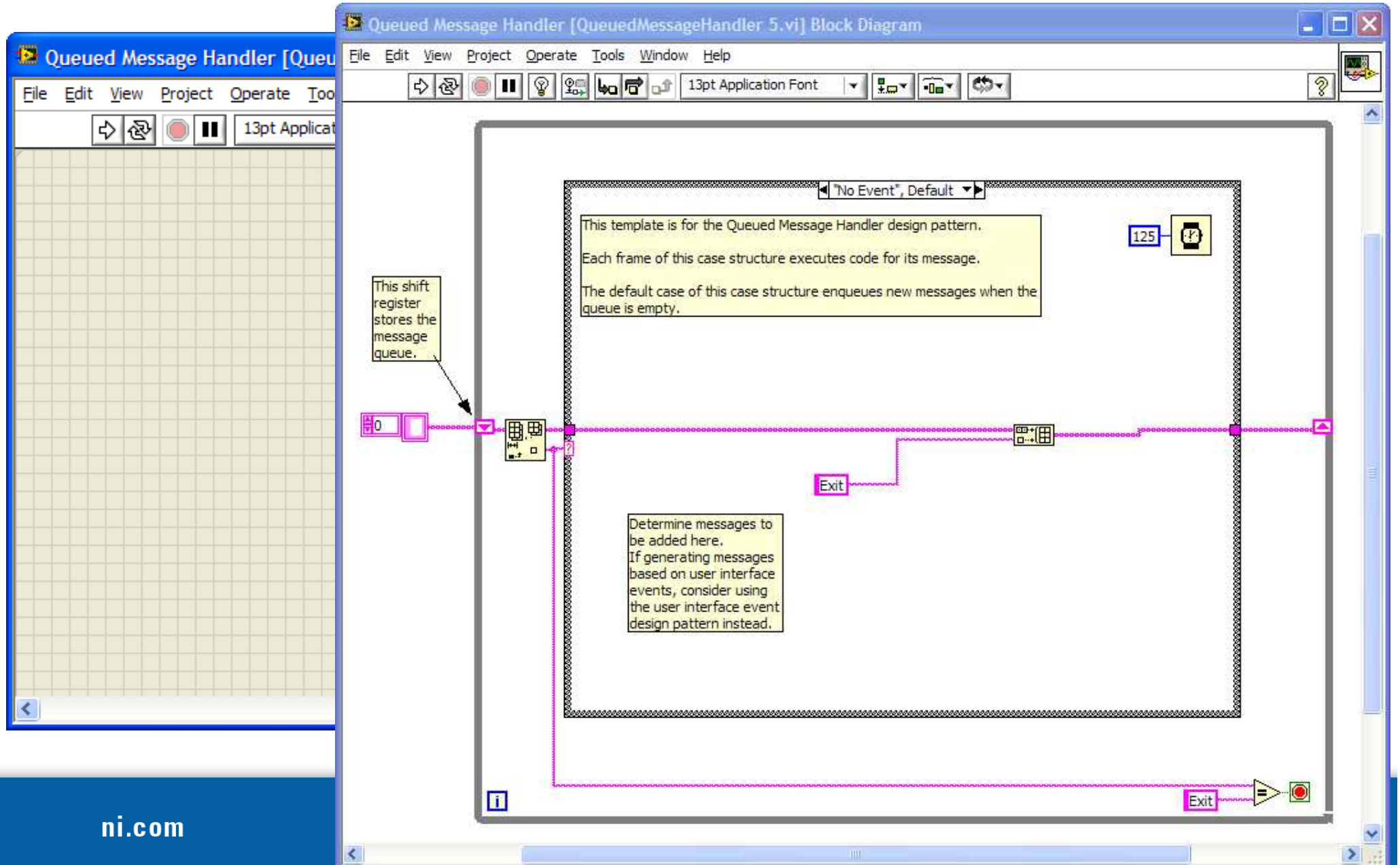
Classic State Machine



Limitations of Classic State Machine

- Can only store one state in shift register
- Not event driven
- Consider the Queued State Machine or Event Driven State Machine

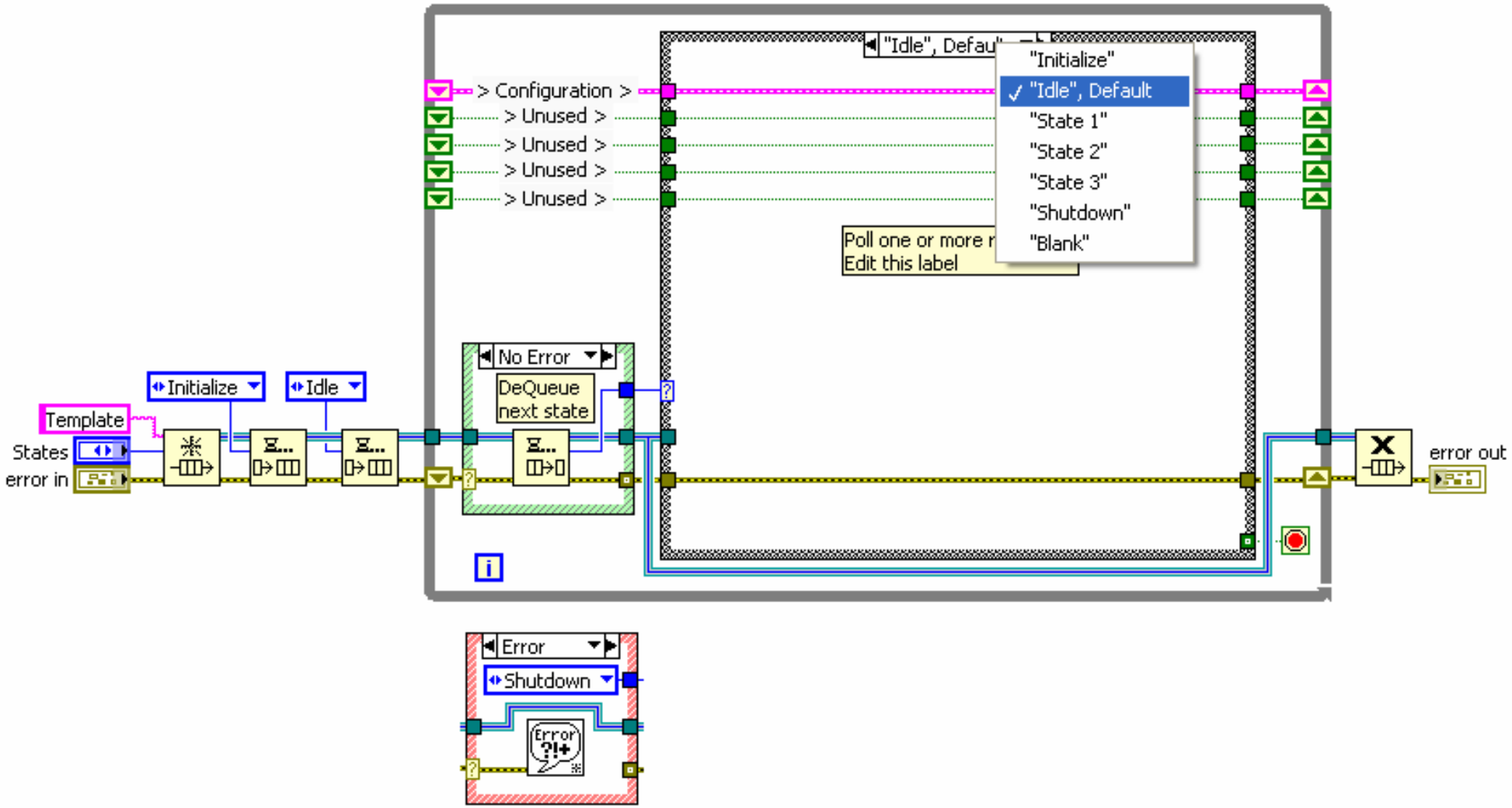
NI Queued Message Handler



Queued State Machine - Rules

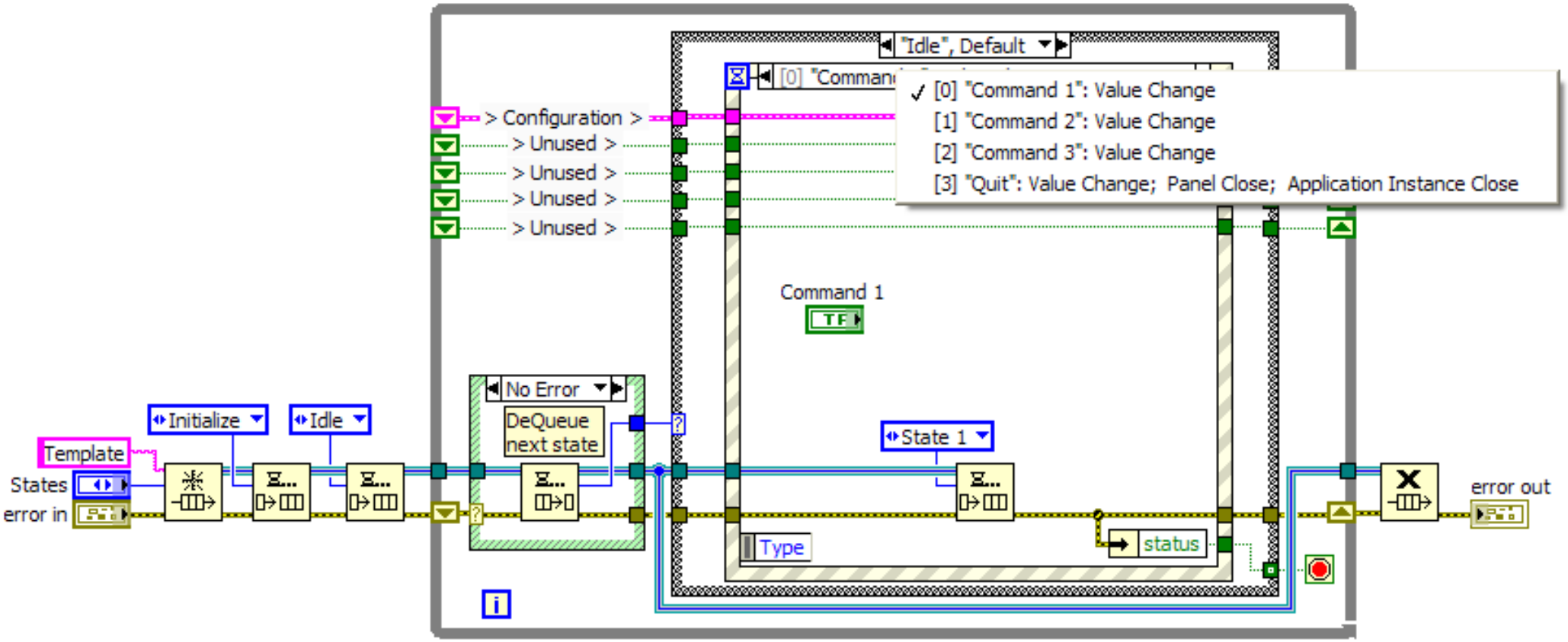
- Use a queue of enum or cluster of enum and variant
- ***Rule 8.15*** *Avoid timeout with Enqueue and Dequeue Element*

Queued State Machine



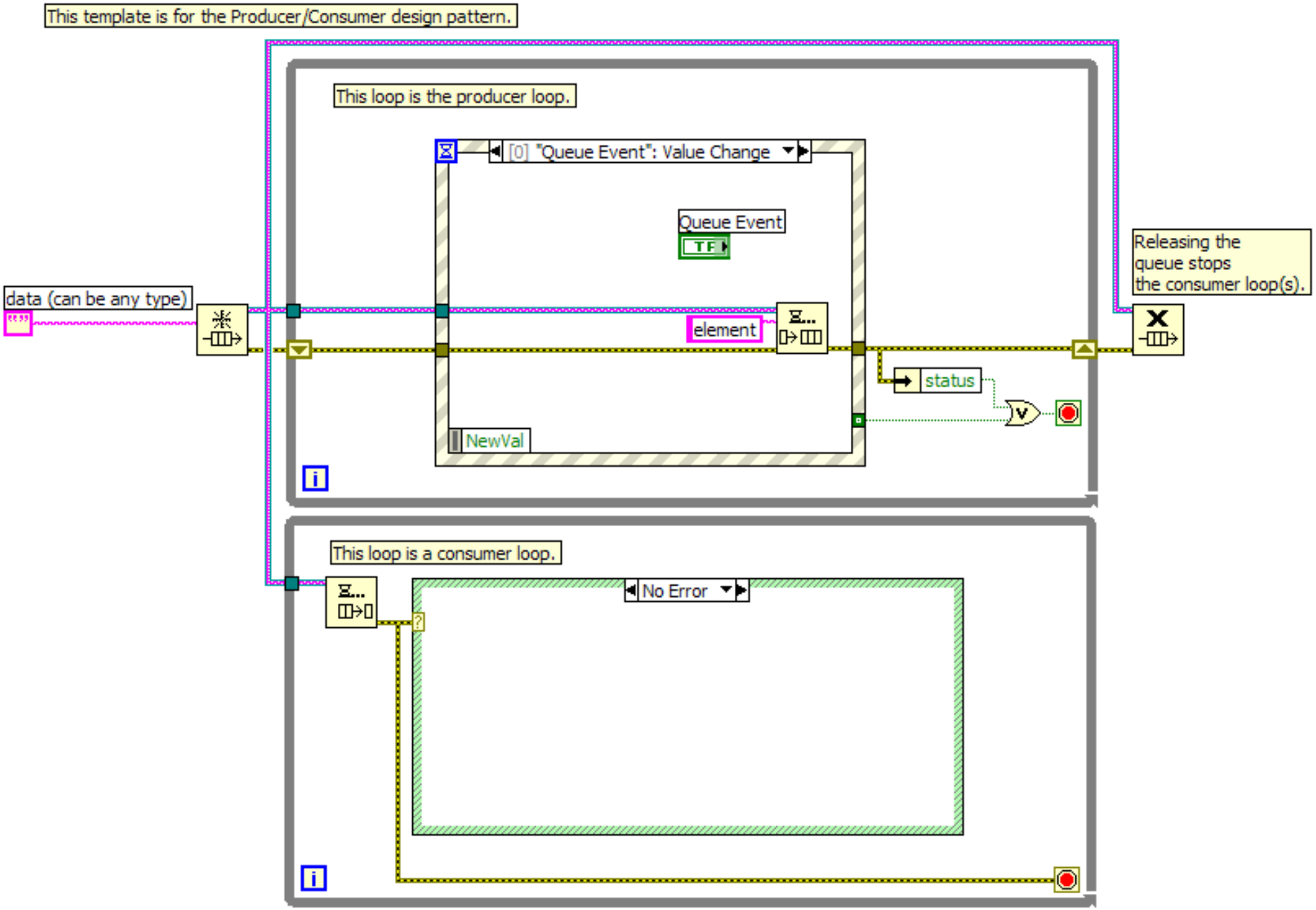
Event-Driven State Machine

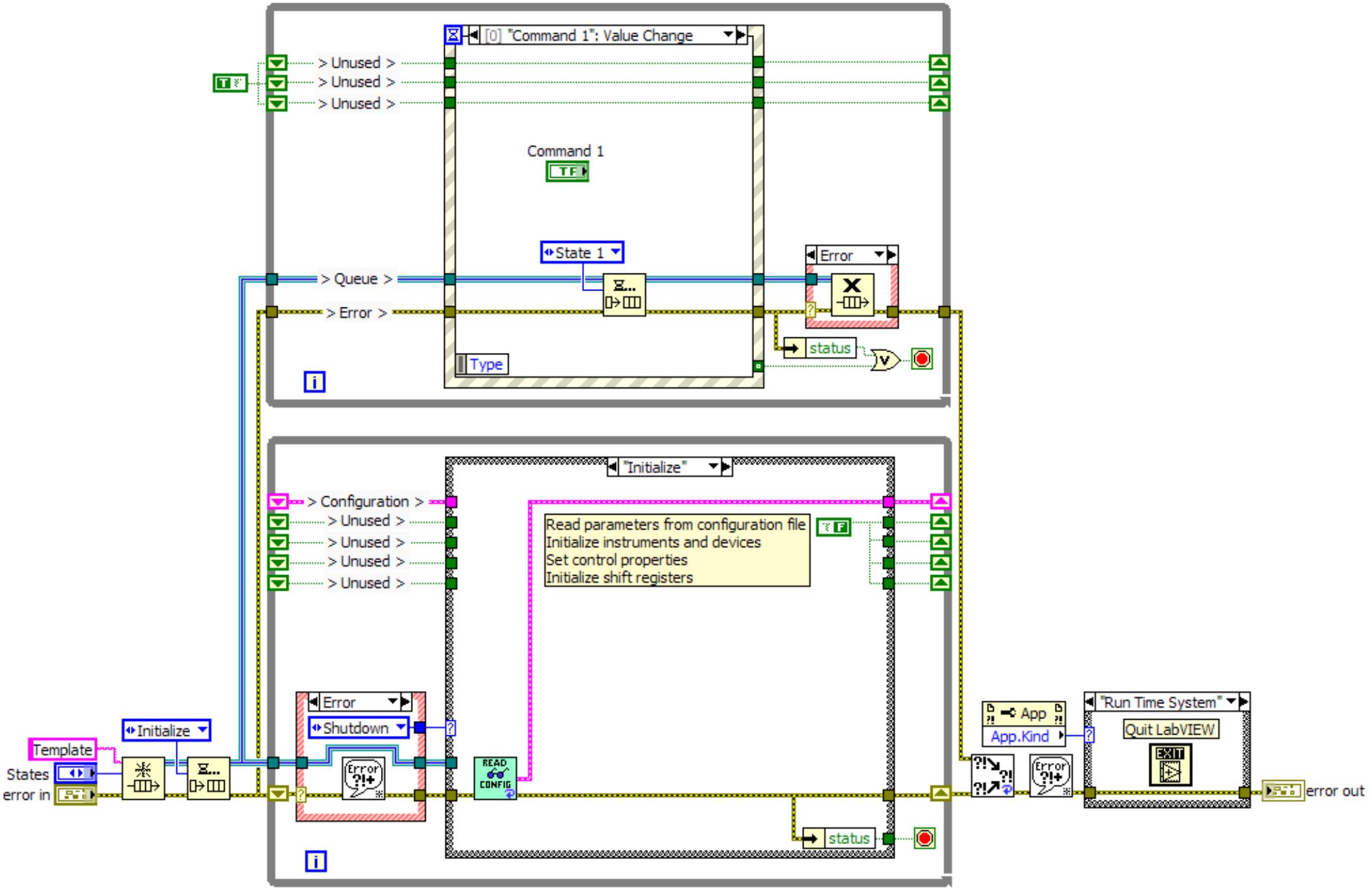
- Queued State Machine with Event structure in Idle state
  - Event structure captures GUI events
  - Event-driven performance


"Idle", Default
[0] "Command 1": Value Change
[1] "Command 2": Value Change
[2] "Command 3": Value Change
[3] "Quit": Value Change; Panel Close; Application Instance Close
> Configuration >
> Unused >
> Unused >
> Unused >
> Unused >
Command 1
TF
No Error
DeQueue next state
Initialize
Idle
Template
States
error in
State 1
Type
status
error out

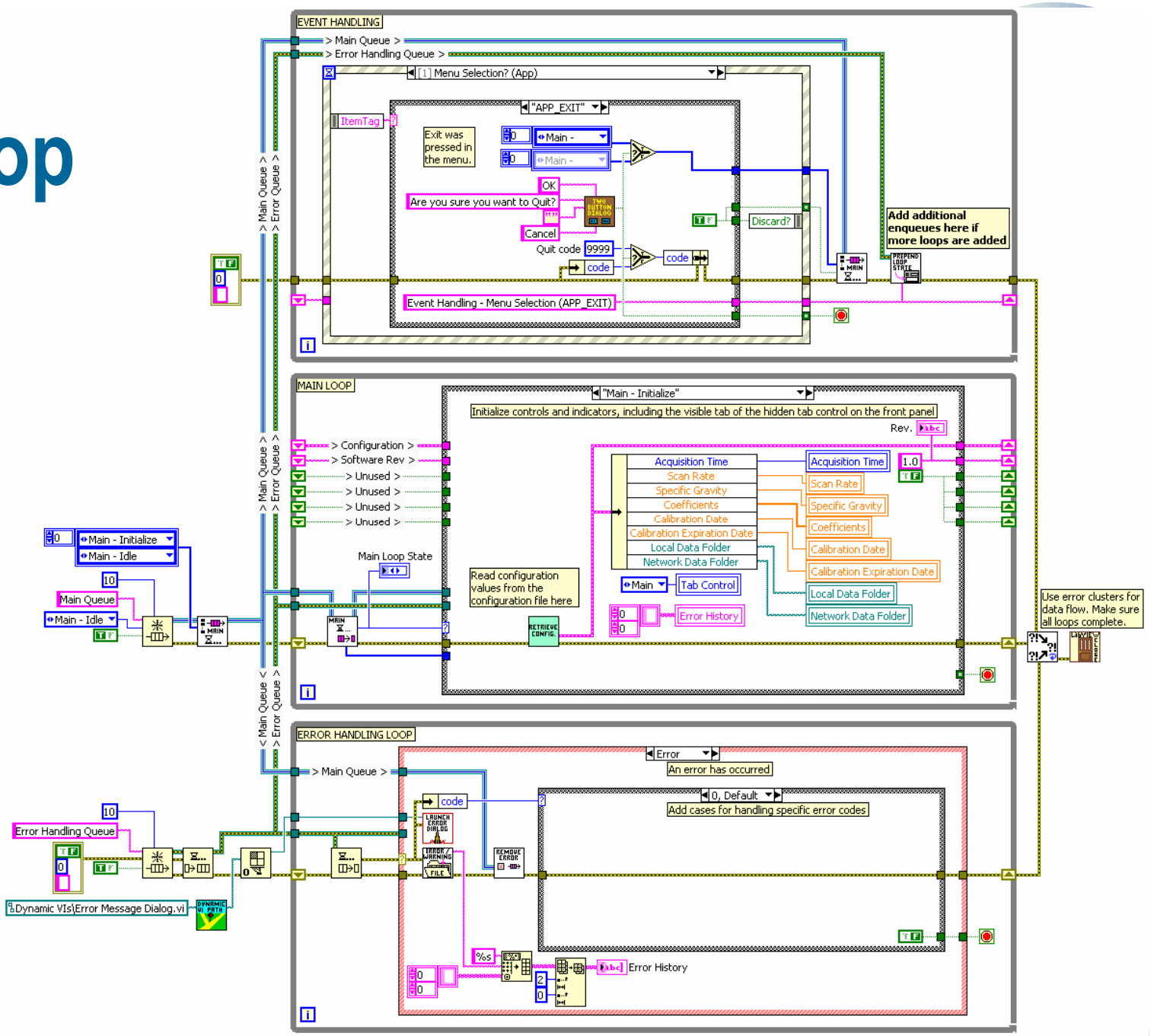
Limitations of Queued State Machines

- Can only process one state at a time
- Processing states freezes the GUI
- Consider separate loops for GUI event handling and state machine


This template is for the Producer/Consumer design pattern.
This loop is the producer loop.
[0] "Queue Event": Value Change
Queue Event
TF
data (can be any type)
element
status
NewVal
Releasing the queue stops the consumer loop(s).
This loop is a consumer loop.
No Error

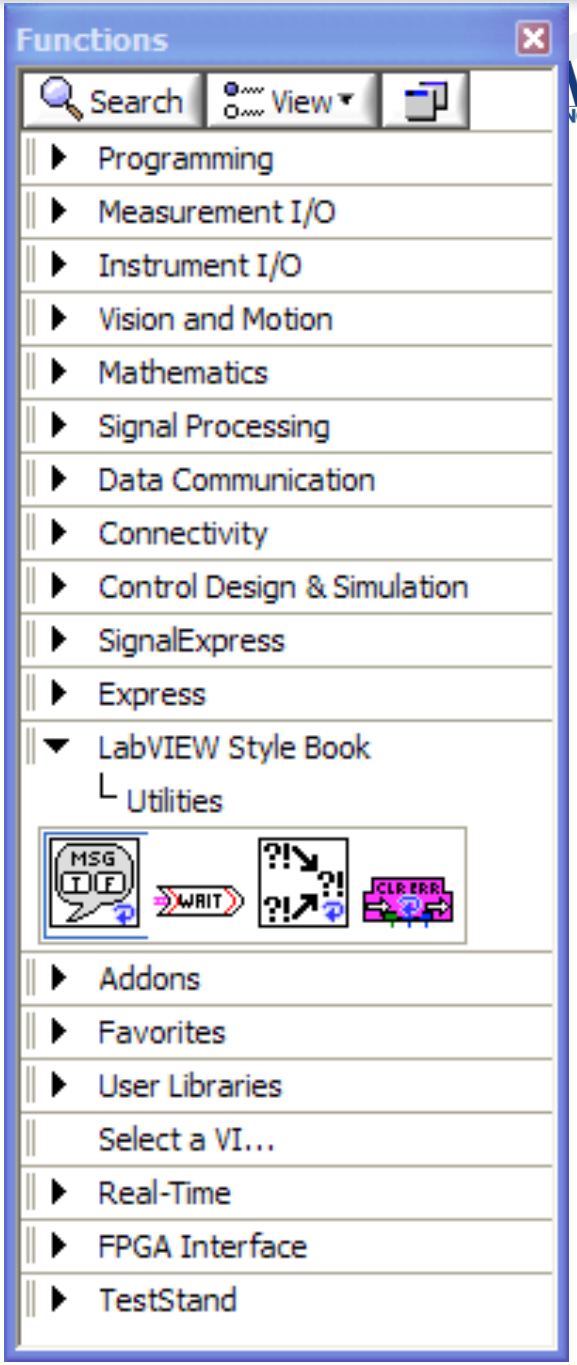

[0] "Command 1": Value Change
> Unused >
> Unused >
> Unused >
Command 1
TF
State 1
Error
> Queue >
> Error >
status
Type
"Initialize"
> Configuration >
> Unused >
> Unused >
> Unused >
> Unused >
Read parameters from configuration file
Initialize instruments and devices
Set control properties
Initialize shift registers
Error
Shutdown
READ CONFIG
status
Template
States
error in
Initialize
App
App.Kind
"Run Time System"
Quit LabVIEW
EXIT
error out

Multiple-Loop Application Framework



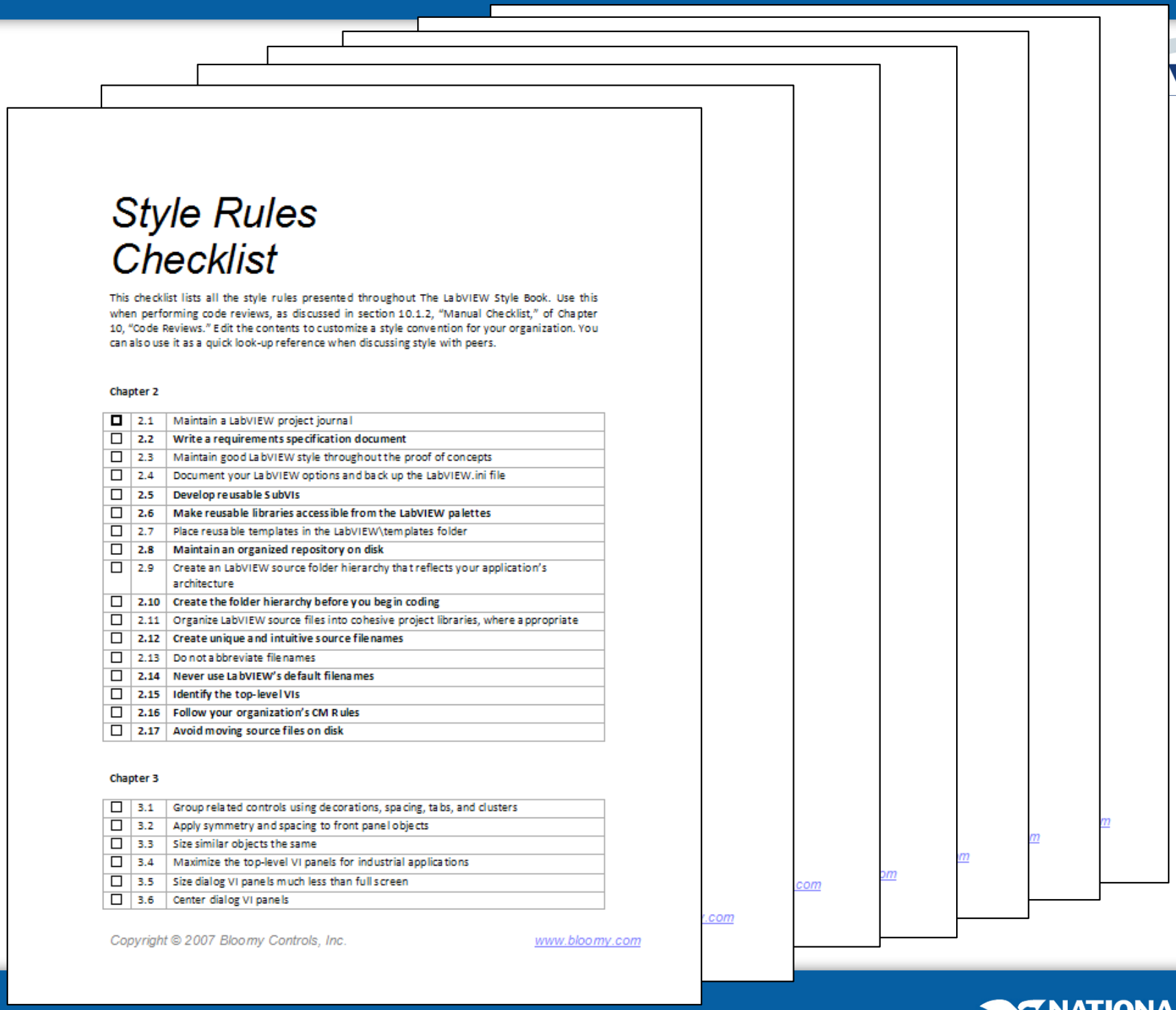
Utility VIs

- Two Button Dialog VI
- Wait n mSec VI
- Merge Multiple Errors VI
- Clear Error All or Specified VI



Style Rules Checklist

- ***Rule 10.1*** *Enforce your organization's style convention using code reviews*
- ***Rule 10.7*** *Use a manual checklist to perform a comprehensive style review*



Style Rules Checklist

This checklist lists all the style rules presented throughout The LabVIEW Style Book. Use this when performing code reviews, as discussed in section 10.1.2, "Manual Checklist," of Chapter 10, "Code Reviews." Edit the contents to customize a style convention for your organization. You can also use it as a quick look-up reference when discussing style with peers.

Chapter 2

☐	2.1	Maintain a LabVIEW project journal
☐	**2.2**	**Write a requirements specification document**
☐	2.3	Maintain good LabVIEW style throughout the proof of concepts
☐	2.4	Document your LabVIEW options and back up the LabVIEW.ini file
☐	**2.5**	**Develop reusable SubVIs**
☐	**2.6**	**Make reusable libraries accessible from the LabVIEW palettes**
☐	2.7	Place reusable templates in the LabVIEW\templates folder
☐	**2.8**	**Maintain an organized repository on disk**
☐	2.9	Create an LabVIEW source folder hierarchy that reflects your application's architecture
☐	**2.10**	**Create the folder hierarchy before you begin coding**
☐	2.11	Organize LabVIEW source files into cohesive project libraries, where appropriate
☐	**2.12**	**Create unique and intuitive source filenames**
☐	2.13	Do not abbreviate filenames
☐	**2.14**	**Never use LabVIEW's default filenames**
☐	**2.15**	**Identify the top-level VIs**
☐	**2.16**	**Follow your organization's CM Rules**
☐	**2.17**	**Avoid moving source files on disk**

Chapter 3

☐	3.1	Group related controls using decorations, spacing, tabs, and clusters
☐	3.2	Apply symmetry and spacing to front panel objects
☐	3.3	Size similar objects the same
☐	3.4	Maximize the top-level VI panels for industrial applications
☐	3.5	Size dialog VI panels much less than full screen
☐	3.6	Center dialog VI panels

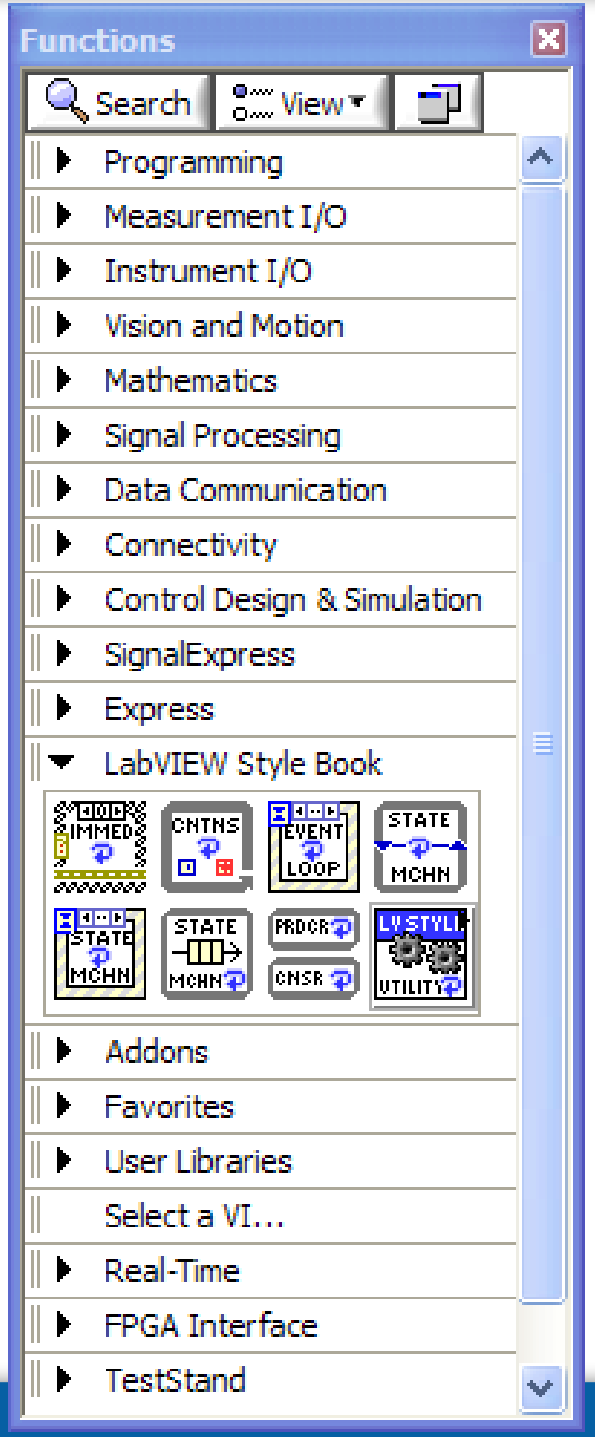
 www.bloomy.com

VI Package

- Requires VI PACKAGE MANAGER
  - Download from jkisoft.com
- Installs on LabVIEW 8.2, 8.5, and 8.6 independently
- Can install on Mac, Windows, and Linux
- Refreshes the LabVIEW palettes immediately after installation / uninstallation / upgrade
- Provides configuration management capabilities
  - Version control
- Easy to use

Conclusion

- Accessible via palettes
- Promote good style
  - Ease of use
  - Efficiency
  - Readability
  - Simplicity
  - Performance
  - Maintainability
  - Reliability
- Save time & money



Feedback & Discussion

- E-mail the author and/or send feedback to lvstyle@bloomy.com
- Download the templates from www.bloomy.com/lvstyle/
- Drop off your card to win a fee copy!

Contact Bloomy Controls

- Email info@bloomy.com
- Write or visit

Headquarters:	**MA Office:**	**NJ Office:**
839 Marshall Phelps Rd.	257 Simarano Dr.	2125 Center Ave.
Windsor, CT 06095	Marlborough, MA	Fort Lee, NJ
(860) 298-9925	(508) 281-8288	(201) 944-9890