Introduction

Who Should Read This Book

This book is suitable for all beginning and intermediate LabVIEW programmers. If you are already a Certified LabVIEW Architect, this book might be too basic for you. Everyone else should find it valuable. This book is unique in its approach. Rather than presenting detailed instructions on how to navigate the LabVIEW user interface, this book focuses on how to use the dataflow paradigm of LabVIEW to create effective programs that are readable, scalable, and maintainable. This book will teach you what good LabVIEW programs look like and, in the process, will teach you how to "think" in LabVIEW.

The first chapter covers everything you need to know to read and understand the programs used, so even absolute beginners can benefit from this book. When new LabVIEW functions are first introduced, they are accompanied by excerpts from the LabVIEW help files for ease of reference. Beginning and intermediate LabVIEW programmers will find design patterns that they can use right away in their own programs. Beginning programmers will have the greatest benefit, because the material in this book is not easily found in other sources. The beginner will "jump start" their understanding of dataflow and produce better code faster than they would otherwise. Intermediate programmers will benefit as well and hopefully will learn techniques that inspire them to revisit and revise their previous work. LabVIEW programmers preparing for the Certified LabVIEW Developer (CLD) exam will especially find this book useful, since it contains full solutions for all previously published CLD exams.

Intermediate and advanced users who have not yet made the leap to LabVIEW classes will also find this book useful. LabVIEW classes are used extensively in the last third of this book. Beginning programmers will perhaps see the greatest benefit because the material in this book is not easily found in other sources. This book provides examples of how to take advantage of data encapsulation, inheritance, composition, and dynamic dispatch when using LabVIEW classes. Intermediate and advanced users will also appreciate the discussion of parallel loop architectures and race conditions, as well as the example of how to use LabVIEW classes to extend the language.

How This Book Is Organized

This book follows a "teach by showing, learn by doing" approach. National Instruments offers multiple levels of certification for aspiring LabVIEW programmers. The Certified LabVIEW Developer (CLD) is the mid-level LabVIEW certification. Certified LabVIEW Developers must pass a four-hour programming exam in which they demonstrate an ability to create functional and well-documented LabVIEW code.

Over the years, National Instruments has published seven CLD exams. One of these is a controller for a car wash, which is used as a demonstration project throughout this book. The chapters in this book tend to follow a three-part pattern:

- A chapter introduces a new concept and demonstrates its use in a simple program (the `Multitest` VI)
- The next chapter shows how this concept can be used in a more complex program, such as the car wash controller
- A problem set is presented in the following chapters, allowing you to implement the new concept using problems based on the six other released CLD exams

This process is repeated throughout the book, focusing on core LabVIEW structures and functions that are the basis of all effective LabVIEW programs.

- Chapter 1 introduces all of the LabVIEW basics needed to read and understand the programs presented in this book. Chapter 2 introduces functional specifications and state diagrams. The functional specifications of the `Multitest` VI and of the car wash controller are presented. Problem Set 1 (Chapter 3) covers the basic LabVIEW material and asks you to develop sub-VIs used in the later problem sets.
- Chapter 4 introduces the classic state machine architecture and demonstrates it using the simple `Multitest` VI. Chapter 5 introduces action engines (also known as functional global variables). Chapter 6 uses an action engine to create a timer for the car wash controller. Problem Set 2 (Chapter 7) gives you an opportunity to create action engines for some of the other CLD exams. Chapter 8 demonstrates how to use the state machine architecture and an action engine to create a fully functioning solution for the car wash controller. Problem Set 3 (Chapter 9) gives you an opportunity to create state machine solutions for some of the other CLD exams.
- Chapter 10 introduces event structures and demonstrates their use in the `Multitest` VI. Chapter 11 upgrades the state machine implementation of the car wash controller to include an event structure. Problem Set 4 (Chapter 12) explores the use of event structures in some of the other CLD exams.
- Chapter 13 introduces queues and demonstrates their use in the `Multitest` VI. Chapter 14 upgrades the event-driven car wash controller to include the use of a queue.
- Chapter 15 applies queues, event structures, and the state machine architecture to a data acquisition program. This chapter discusses how data acquisition tasks

are created and used in a LabVIEW program. A program is developed that allows the user to null, calibrate, and read a strain gage in a quarter-bridge circuit. Problem Set 5 (Chapter 16) explores the use of queues in some of the other CLD exams.

- Chapter 17 explains how to use the material presented in the previous chapters to prepare for the Certified LabVIEW Developer exam. Problem Set 6 (Chapter 18) and Problem Set 7 (Chapter 19) provide full walk-throughs of two of the published CLD exams that were not used in the prior problem sets.
- Chapter 20 expands on the topic of state machines with arguments and string-based queues. These are demonstrated using the `Multitest` VI. Chapter 21 upgrades the queued car wash controller by adding arguments and a string-based queue. Problem Set 8 (Chapter 22) explores the use of arguments in some of the other CLD exams.
- Chapter 23 introduces LabVIEW classes and discusses data encapsulation using classes. The car wash controller with arguments is upgraded by replacing the action engine with a LabVIEW class. Problem Set 9 (Chapter 24) explores the use of LabVIEW classes to encapsulate data in some of the other CLD exams.
- Chapter 25 introduces the concepts of inheritance and composition in LabVIEW classes. The car wash controller with LabVIEW classes is upgraded to take advantage of inheritance and composition. Problem Set 10 (Chapter 24) explores the use of inheritance and composition in some of the other CLD exams.
- Chapter 27 introduces parallel loop architectures and demonstrates the producer–consumer design pattern using the `Multitest` VI. Chapter 28 discusses race conditions in parallel loop architectures and how to combat them, demonstrating these techniques using the car wash controller. Chapter 29 uses LabVIEW classes to extend the language and create a new form of queue that can be used to defeat the race conditions shown in Chapter 28. Problem Set 11 (Chapter 30) explores the use of parallel loop architectures in some of the other CLD exams.
- Chapter 31 provides a look back at the material covered in this book. The appendices present the functional specifications of the CLD exams used in the problem sets, as well as providing full solutions to all of the exercises in the problem sets.

The website that accompanies this book (http://www.ntspress.com/publications/effective-labview-programming/user-resources/) contains all of the programs presented in the main body of the text as well as solutions to the exercises presented in the problem sets. The programs (or VIs as they are called in LabVIEW) can be found in the "All VIs" folder on the site. This folder contains subfolders dedicated to all of the major projects used in this book: the ATM controller, the Boiler controller, the Car Wash controller, the Car Wash 2 controller, the DAQ State Machine, the Multitest State Machine, the Security System controller, the Sprinkler controller, and the Traffic Light controller. Each of these folders contains all of the files related to the various versions of these projects presented in this book.

LabVIEW Projects also have been provided to make it easier to navigate the codes in the "All VIs" folder. A LabVIEW Project is a library that can be used to provide a virtual organization of VIs and other material without changing the location of the codes. The LabVIEW Projects are used to present selected VIs and all of their supporting sub-VIs and type definitions in one easily navigated package.

The website contains four folders of LabVIEW Projects. The LabVIEW Projects in the "Problem Set Solutions" folder contain the solutions to the exercises presented in this book. There is one LabVIEW Project for each problem set. The individual exercise solutions are contained in labeled virtual folders in the project. Clicking on the entries in these folders will automatically open the solutions to the corresponding exercises.

The website also contains a folder of LabVIEW Projects called "VIs Organized By Chapter." The LabVIEW Projects in this folder are each dedicated to a separate chapter in the book. If you would like to examine a VI described in a chapter, navigate to the LabVIEW Project for that chapter and double-click on the desired VI to open it.

There is also a folder called "VIs Organized By Topic." This folder groups all of the example codes and the problem set solutions by topic. For example, the "Classic State Machine" subfolder contains LabVIEW Projects dedicated to classic state machine implementations of the Boiler controller, the Car Wash controller, the Car Wash 2 controller, the Multitest State Machine, the Security System controller, and the Traffic Light controller. This set of LabVIEW Projects is very useful to see how a specific feature or pattern can be used in a variety of programming projects.

The website also contains a folder of LabVIEW Projects called "Practical CLD Exam Solutions." These projects are dedicated to VIs that satisfy the requirements of the Certified LabVIEW Developer exam at a level of complexity that is appropriate to the four-hour time limit of the exam. The original CLD exam functional specification documents also can be found on the website in the "Original CLD Exam Documents" folder.