

Introduction

Purpose	1
Research	2
Interpretation	2
Audience.....	2
Using This Book.....	3
Quick Tour	3

Becoming proficient in building software systems is not a function of perspiration, inspiration, or even experience for that matter. Building a complete system is like building a house from the ground up. It requires planning—a LOT of planning.

This publication is a culmination of the author's experiences as they relate to developing complete systems using SAS software. While the SAS programming language has a long history, many SAS programmers have not been exposed to systems development. Perhaps there is a historical perspective behind this phenomenon. SAS was originally developed as a powerful statistical language. However, the DATA step functionality has greatly expanded, including the addition of the macro language. Furthermore, multiple new SAS products were developed for virtually every type of business or scientific application. Becoming comfortable with all the new language elements was certainly enough of a challenge, but learning to write SAS code in an organized, modular way was not a high priority.

Purpose

The purpose of this publication is to demonstrate structured, modular programming techniques for creating complete systems using SAS software. However, readers who simply want to learn how to develop more complex programs will also find this publication to be invaluable. The focus of this publication is the development of software as it relates to the full software life cycle. Specifically, this publication walks through all phases of program development, from analysis to user acceptance.

The base SAS language was chosen to demonstrate these principles because it is the core of the SAS System. The base product is a rich language which contains all the constructs that you need to create complete, structured, modular programs.

Choosing material to be included in the text was a difficult task. However, as a general rule, the book tends to focus on "classical" techniques. These methods date as far back as the 1960's, when programmers were looking for guidance in developing software systems. All of these methods are still valid today for teaching programmers the fundamentals of software development.

The literature is exhaustive on topics such as modular and structured systems development, as well as the software life cycle. However, very little has been written relating these concepts directly to the SAS programming language. Hence, a significant portion of this publication weaves these elements together. Most chapters begin with a theoretical treatment of structured programming theory or key concepts that relate to a particular component of the software life cycle. After digesting the underlying principles, the reader is taught how to relate these principles to specific constructs in the SAS programming language.

Research

A number of different sources provided material for this publication. First and foremost is the author's experience relating to systems development. Another good source of material was a perusal of other software systems, written both in SAS and in other programming languages. Finally, a number of books and articles were reviewed that relate to all aspects of the software life cycle. This was invaluable for modeling the chapters in this book.

Interpretation

Much of the material presented here is the author's interpretation of modular programming theory as it applies to the SAS programming language. You may not agree with every idea, construct, or philosophy presented here. What is really important is to urge you, the reader, to *think* about what you are doing when you are coding with the SAS language.

It is easy to become involved in rote coding practices. Take a chance, and try something different. Play with the SAS language as much as you can. Undoubtedly, you will discover that experimentation is the programmer's best teacher.

Audience

Much of the material in this book focuses on philosophy rather than specific programming technique. Hence, this book can be beneficial to a wide range of programming experiences. Furthermore, the platform that you work on is irrelevant. So, if you can confidently write simple SAS programs, then you can learn from the instruction in this book. In fact, this publication is useful for anyone who has six months experience up to advanced programming ability.

Although the information in this book focuses on the base product, users who work with other SAS products can certainly benefit from the principles of systems development. It should benefit anyone who wants to move to the next level of systems development using SAS software.

Finally, this publication does not teach elementary SAS programming constructs, nor does it attempt to teach syntax. It assumes that the reader understands the basic syntax for the SAS language.

Using This Book

All source code was tested on the SAS System, Release 6.12, for the Windows environment (TS020). While there may be an occasional reference to code specific to Windows, most of the code is portable to a wide variety of operating environments. This feature is intended to allow readers from different environments to benefit from the material in this publication.

You may find that you read the book sequentially as a project develops, or that you may skip around and read only chapters of interest. In either case, the material in each chapter is highly independent, so you won't lose the flow if you decide not to read the book in sequential order.

Much of the code in this book is presented as small code fragments. This should allow you to grasp the salient concepts and then apply them to their own situations. You may see ellipses (...) embedded within the code snippets. This means "other SAS code here." This omitted code is immaterial for the discussion at hand.

Also, for brevity, certain constructs are omitted from code fragments. For example, using KEEP and RETURN statements is highly recommended in other parts of the text. However, they are often omitted from the code fragments only because they tend to obscure the salient points of the demonstration code. Similarly, defensive coding practices and comprehensive error checking are encouraged throughout the book. However, due to space limitations, these constructs may be omitted in code segments and in Appendix 1, "Coding Samples."

Quick Tour

Here is a brief overview of each chapter:

Chapter 2, "Overview of the Software Life Cycle," provides a preliminary, tongue-in-cheek look at the entire software development process. This chapter should be useful for preparing you for the long (though relatively painless) journey ahead. The content presented here should be particularly useful for those who may not be familiar with the range of terminology relating to systems development and the software life cycle.

Chapter 3, "Managing the Project," introduces you to the general concerns required to handle large projects, from assigning programming tasks, to providing external documentation.

Chapter 4, "Analyzing the Problem," describes the beginning of the software life cycle. This chapter discusses several key concepts regarding problem definition, heuristics, and analysis.

Chapter 5, "Laying the Foundation with Good Design," allows for the transition between analysis and coding. The key elements presented include modularity and the means of communicating data between modules. Also included is a formal discussion of building SAS modules with an emphasis on both intramodule and intermodule strength.

Chapter 6, "Improving Readability and Providing Internal Documentation," explores two areas that are often ignored in program development. The first section discusses principles for improving the readability of a program. The second section suggests ways to improve source code documentation.

Chapter 7, “Structured Programming Techniques,” examines the principles of building program structure. Little details in coding can have profound effects in systems development. This chapter explains details about structured methods and the theoretical framework behind them.

Chapter 8, “System Coding Principles,” addresses coding concerns involved in building a complete software system. In addition, this chapter explores the development of a modular reporting system using the DATA _NULL_ step.

Chapter 9, “Program Verification and Testing Methodology,” describes testing, an often overlooked part of the life cycle equation. You will learn about testing techniques that go beyond the random nature of typical systems testing.

Chapter 10, “Digging Out with Debugging Techniques,” provides useful techniques for finding and repairing program bugs. In addition, this chapter contains an introduction to the SAS DATA step debugger.

Chapter 11, “Putting It All Together with a Demonstration: The Site Map System,” is a concise description of a project developed for the Environmental Protection Agency. This chapter briefly walks through all phases of the software life cycle so you can appreciate the process of systems development.

Chapter 12, “Answers to Exercises,” provides answers to all of the problems presented at the end of each chapter.

Appendices: Appendix 1, “Coding Samples,” is a collection of SAS code that is referenced within the chapters. These are usually small programs that are too large to be embedded within the text. The remaining appendices (2 through 5) are useful for those who want to explore the details of the project discussed in Chapter 11. These appendices provide definitions and details about algorithms used to generate the code.

References are provided at the end of each chapter, including both the bibliography and cited references.