CHAPTER 3

DATA Step Programming

The DATA Step programming language in the Base SAS software is a powerful and full-featured programming language. Whether you’re a novice or advanced user, the tips covered in this chapter are designed to show you how to use it to your advantage. This chapter covers important keywords, structured coding techniques, date and time processing, operators and modifiers, SAS functions, program testing and debugging techniques, large file-processing techniques, and concepts such as observation loops, Program Data Vector (PDV), variable and value assignments, conditional processing, and logic branching—all to enable you to write better programs.

In this chapter, you’ll learn how to:

- Direct control to the top of an observation loop
- Retain values across observation loops
- Assign a variable’s length
- Perform conditional logic
- Implement best practices coding standards
- Perform date and time processing
- Use date formats and informats
- Apply operators and modifiers and their order of evaluation
- Use arithmetic, character, date and time, and array functions as part of SAS statements
- Test and debug SAS program code
- Process large file efficiently and quickly
- Document and educate other users on complex processes, pieces of code, and other abstract coding constructs

The DATA Step Programming Language

This section includes tips that will show you how to get the most from the DATA step programming language. From understanding the DATA step’s many features and capabilities to visualizing how the observation loop works, you’ll find ways to take command of this powerful programming language. You’ll also learn valuable techniques that can be applied to your own program code quickly and easily.

Understanding the DATA step language

The DATA step language is at the heart of the SAS System. It’s a full-featured programming language designed to perform complex and powerful computing operations. In contrast to other compiled languages, a SAS program is structured in a top-down fashion. What this means is that a multi-step program is grouped...
into steps with each step running one step at a time, one after the next.

The DATA step language permits structured coding, algorithm, and data-processing constructs to be expressed using a style that is relatively easy to read and understand by other SAS users. An important feature of the DATA step language is that programs written on one machine are portable to other machines that fully support the SAS System (with the exception of operating system differences).

2 Understanding DATA step features and capabilities

The DATA step is a comprehensive, powerful, high-level language. As with all powerful languages, it possesses intrinsic features that enable it to be used for a variety of tasks and applications. The following list highlights some advantages of the DATA step.

• It has the ability to apply structured programming techniques, including the use of expressions and operators.

• It’s a well-defined language that is free of ambiguities. It’s considered self-documenting by some (a later section in this chapter presents useful tips to improve this important aspect of coding).

• It supports a variety of structures including strings, arrays, indexes, tables (formats), and more.

• It supports the ability to use a modular approach and can communicate with the calling program by way of parameters or global variables.

• It supports a variety of input-output access methods including sequential, fixed and variable length, delimited, hierarchical, random access, and third-party application (including MS-Word, Excel, and Access) files. SAS offers other licensed products that interface popular database products such as DB2, IMS, Oracle, and others.

• It is machine independent as well as portable.

• It quickly compiles and executes on machines where it is implemented.

• Its features and constructs are easily taught and learned without difficulty.

• It is useful in a wide range of programming applications and isn’t designed for a narrow purpose or audience.

3 DATA step processing begins with the DATA statement

DATA step processing begins with the DATA statement. As the first statement in a user-written program, the DATA statement signals the beginning of a user-written step (as opposed to a canned procedure) and frequently also the creation of a SAS data set (unless a _NULL_ data set is specified). As the first executable DATA step statement, the DATA statement triggers the beginning of a DATA step and activates the DATA compiler. Once the step is successfully compiled it’s automatically executed.