

CHAPTER

1

Introduction to SAS/AF Software

<i>Overview of SAS/AF Software</i>	3
<i>Purpose of This Document</i>	3
<i>Getting More Information</i>	4
<i>Help</i>	4
<i>Documentation Available on the Web</i>	4
<i>Software Requirements</i>	4
<i>Mainframe Support</i>	4

Overview of SAS/AF Software

SAS/AF software is a set of tools for developing applications. Central to the SAS/AF development environment is the *frame*. You can think of a frame as an application window that contains the interface (the fields and buttons) of your application. With SAS/AF frame application development, you can build much of your application visually, using drag-and-drop components. And because SAS/AF applications are stored in SAS catalogs, they are portable to all SAS software platforms.

Purpose of This Document

This document is an introduction to the SAS/AF development environment. It guides you through the basic skills that you need to build a simple frame application. It also gives you a foundation with which you can transition to the larger reference manuals that fully cover SAS/AF software.

Although this document is intended for new users of SAS/AF, you should be familiar with basic SAS concepts such as libraries, catalogs, and catalog entries. You do not need object-oriented programming experience to benefit from this document, but familiarity with object-oriented concepts will certainly help.

Although specific to SAS/AF in SAS®9, the overall processes that are presented in this document also apply to versions of SAS/AF software starting with SAS 8.1.

Getting More Information

Help

Help is always available when you are using the SAS/AF development environment. To access help, select **Help ► SAS Help and Documentation**, navigate to **SAS Products** and then navigate to **SAS/AF**.

You can also get help on most windows and dialog boxes inside SAS/AF by pressing the F1 key when the window or dialog box is the active window, or by selecting **Help ► Using This Window**.

To access Help on a component in the Components window, right-click on the component, and then select **Help on Class**.

Documentation Available on the Web

SAS documentation, available in HTML or PDF, is available on the Web at <http://support.sas.com/documentation/onlinedoc/>.

The following books offer more information about developing applications using SAS/AF software:

- *SAS Guide to Applications Development, Second Edition*
- *SAS Component Language 9.1: Reference*
- *SAS/AF 9.1 Procedure Guide*

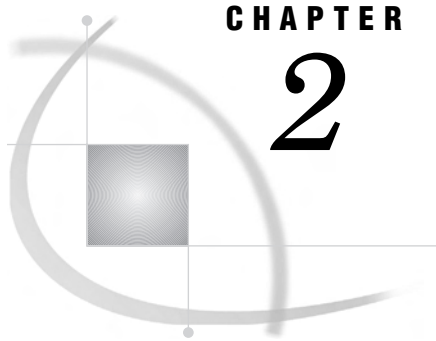
The text of all three of these books is also available in the SAS/AF help.

Software Requirements

To build the frame applications in this document, you must have SAS/AF software installed, and you must have a monitor that is capable of displaying graphics. To run the frame applications in this document, you must have Base SAS software.

Mainframe Support

SAS/AF does not support frame application development on a mainframe. However, you can build a frame application on another platform and then port that application to a mainframe platform (see “Native Controls” on page 5).



CHAPTER

2

The Building Blocks of Frame Applications

<i>Components, Controls, and Models</i>	5
<i>Native Controls</i>	5
<i>The SAS/AF Development Environment</i>	6
<i>The Frame</i>	6
<i>The Components Window</i>	7
<i>The Properties Window</i>	7
<i>The Source Window</i>	8
<i>A Simple Methodology for Frame Development</i>	9
<i>Using Models</i>	9

Components, Controls, and Models

Components are pieces of software that you can use to build applications. SAS/AF provides several components that enable you to build graphical user interfaces and then link those interfaces to data. There are two basic types of components: controls and models.

Controls constitute the graphical user interface, and include interface elements that you have seen in Web forms like Check Boxes, List Boxes, and Entry Fields. There are also controls that are specific to SAS/AF such as the Table Viewer (which displays SAS table data).

Models are another type of component. In contrast to the controls that are displayed to the user in the interface, models work behind the scenes to distribute data to controls. For example, to get a List Box to display a list of SAS libraries, you would attach a Library List model to the List Box.

Controls are sometimes called *visual components*, and models are sometimes called *non-visual components*. Controls and models are also generically called *objects*, especially in the context of object-oriented programming.

Native Controls

The controls that are supplied by SAS always appear as *native controls* on a platform, even if you ported your application to that platform. This means that if you wrote an application on Solaris, and then ported it to Windows XP, the application would look exactly like other Windows XP applications.

However, if you run a graphical user interface application on a character-based display (usually on mainframes), the controls (for example, the entry fields and list boxes) are represented as characters, which means the controls will look different from the examples in this document.