Preface

The *KCMS Application Developer's Guide* describes the Kodak Color Management System (KCMS) framework application programming interface. The purpose of the KCMS framework is to enable the accurate reproduction, and improve the appearance of, digital color images on desktop computers and associated peripherals. With this C API, you can write applications that perform correct color conversions and manipulations.

Who Should Use This Book

The intended audience of this manual is the professional programmer who is fluent in the C programming language and writing an application that:

- Uses color data
- Prints images
- Is an imaging tool
- Uses PhotoCD

Before You Read This Book

Check the following manuals for any corrections or updates to the information in this manual:

- *Solaris 2.5 Software Developer Kit Introduction*
- *Solaris 2.5 Software Developer Kit Installation Guide*

See the on-line SUNWrdm packages for information on bugs and issues, engineering news, and patches. For Solaris installation bugs and for late-breaking bugs, news, and patch information, see the *Solaris Installation Notes* (SPARC or x86).

For SPARC systems, consult any updates your hardware manufacturer provided.

Although you do not have to be a color scientist to write applications with the KCMS API, a certain amount of color literacy is helpful. Table P-1 lists two white papers that contain some basic information on color and KCMS. The files are located online in the `/usr/openwin/demo/kcms/docs/` directory.

Table P-1 KCMS White Papers

File Name	Title
`kcms-wp.ps`	*An Introduction to the Kodak Color Management System*
`kcms-wp-solaris.ps`	*Kodak Color Management System*

The KCMS framework this manual describes uses the International Color Consortium (ICC) format as the default format for color manipulation. For details on ICC, you should read the *International Color Consortium Profile Format Specification*. It is located by default in the `icc.ps` file in the `/opt/SUNWsdk/kcms/doc` directory. This manual refers to that document as the ICC specification.

Related Manuals

The following manuals will help you further understand the Driver Developer Kit (DDK) portion of the KCMS software product. These manuals are located in the DDK AnswerBook.

- *KCMS CMM Developer's Guide*
- *KCMS CMM Reference Manual*

The following manuals will help you further understand the Calibrator Tool portion of the KCMS software product.

- *Solaris Advanced User's Guide*

  In Chapter 10, "Customizing Your Environment," there is a section called "Calibrating Your Monitor." The section tells you how to adjust your viewing environment and how to calibrate your monitor with Calibrator Tool. This manual is in the Solaris 2.5 User AnswerBook.

- *KCMS Calibrator Tool Loadable Interface Guide*

  This manual will help you further understand the API to the Calibrator Tool. You can tailor the Calibrator Tool for your specific calibrator hardware and software with this API. This manual is in the KCMS AnswerBook.

Suggested Reading

It is highly recommended that you be familiar with, or have access to, the following manuals and manual pages to help you with topics discussed in this manual:

- `setlocale(3c)`
- *Solaris Developer's Guide to Internationalization*

How This Book Is Organized

This document consists of the following chapters and appendix:

- Chapter 1, "Introduction" explains the KCMS architecture and programming environment. In addition, it introduces you to several online sample programs that demonstrate the use of the KCMS API.
- Chapter 2, "Profiles" explains profiles, which are the focus of your programming efforts with the KCMS framework.
- Chapter 3, "Data Structures" describes the data structures of the KCMS framework.
- Chapter 4, "Functions" details each API function.
- Chapter 5, "KCMS Profile Attributes and ICC Tags" details each profile attribute and ICC tag.

- Chapter 6, "Warning and Error Messages" describes status codes (error and warning messages) returned by the KCMS framework functions.

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

Table P-2 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

API Naming Conventions

The naming conventions shown in Table P-3 are used throughout the KCMS framework and this guide.

Table P-3 API Naming Conventions

Item	Convention	Examples
Attribute names	ICC profile format attribute names begin with "`ic`"—`ic`<*AttributeName*>	`icHeader`
Data structures Typedefs Constants	ICC profile format data structures begin with "`ic`". All other data structures, typedefs, and constants are KCMS specific and begin with "Kcs"—`Kcs`<*TypeDefName*>	`icTextDescription` `KcsCalibrationData`

Table P-3 API Naming Conventions *(Continued)*

Item	Convention	Examples
Functions	Each significant word in a function name is capitalized. Intervening spaces are removed—`Kcs`<*FunctionName*>()	`KcsConnectProfiles()`
Macros	Macros are KCMS specific and are capitalized—`KCS_`<*MACRO_NAME*>	`KCS_DEFAULT_ATTRIB_COUNT`
Status codes	All status codes are capitalized and have the format `KCS_`<*STATUS_CODE*>	`KCS_PROF_ID_BAD`

Note – Historically KCMS was referred to by the acronym *KCS* (or *Kcs*). This acronym has been carried forward as the prefix in KCMS data type names, for example, KcsCalibrationData.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

Table P-4 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`