Introduction 1

This chapter introduces you to the Kodak Color Management System (KCMS) product. It describes each of the components of the KCMS architecture and tells you about programming requirements and hints when writing your KCMS application.

KCMS Architecture

The KCMS architecture provides a way to encapsulate specific color management functions in color profiles. Figure 1-1 illustrates the architecture of the KCMS environment. Each segment filled with gray is supplied by SunSoft; these are the default components. The other segments, filled with white, are components that you can add to your development environment.

Each component is discussed further in the following sections.

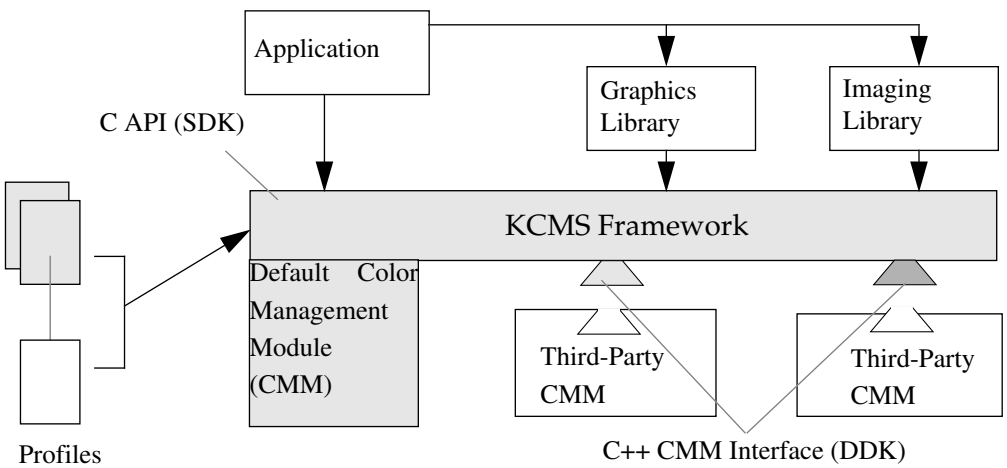

Figure 1-1 KCMS Architecture

Applications

At the top of the hierarchy are applications. With the KCMS framework you can write an application that:

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

Applications connect color profiles to provide a variety of new forms, thus minimizing the task of predefining all possibilities. With the 14 available KCMS API functions, your application can load, create and update profiles, connect and optimize profiles, and then process data through the result.

C API

The KCMS C API provides functions for your application to communicate with the KCMS framework and color management modules (CMMs). The C API is a portable programming interface that allows applications to manipulate color profiles and to use them to correct color data.

The C API consists of:

- A set of callable functions
- Header files
- A shared library and dynamically loaded code modules required for Solaris

KCMS Framework

The KCMS framework loads and saves profiles, gets and sets KCMS profile attributes, and directs requests for color management to the right CMM at the right time. It is particularly vital in calls that involve more than one CMM. The KCMS framework also maintains attributes and executes certain default behaviors and functionality.

Color management is performed by the framework and the CMMs. You can concentrate on dealing with profiles because the KCMS framework makes color management details transparent to the caller.

Profiles

Profiles are files that tell the KCMS framework how to convert input color data to the appropriate *color-corrected* output color data. They will be the focus of your programming efforts. For example, your application might load profiles, read profile attributes, connect profiles, optimize profiles, and apply profiles to color data.

See Chapter 2, “Profiles,” for detailed information.

Graphics and Imaging Libraries

Table 1-1 lists some of the imaging and graphics libraries available to use with the KCMS framework.

Table 1-1 Optional Imaging and Graphics Libraries

Library	Description
PEXlib	PHIGS Extensions to the X Library
XGL	Solaris 3D Graphics Foundation Library
XIElib	X Imaging Extension Library
XIL	Solaris Foundation Imaging Library
Xlib	X11 Window System Library

You can mix KCMS calls with any calls from these libraries. If the library you choose supports color management, your application may not need to make direct calls to the KCMS framework; the other library may already make those direct KCMS calls. See the documentation with the imaging and graphics library of your choice to see if that library already supports color management.

Color Management Modules

A color management module (CMM) is the component that ultimately does the color correction. Different CMMs use different techniques for evaluating color data, which can result in differences in quality, size, and speed of color correction.

Because CMMs are loaded at run-time and CMM interfaces are extendable, you can take advantage of the improvements in existing technologies and the latest color-correction technology, along with hardware acceleration, (without changing your code or rebuilding your application) by changing or adding new CMMs, or profiles, or both.

A Kodak-supplied CMM is the default CMM. You can write your own CMM (third-party CMM) or override portions of the default CMM. To write your own CMM you must purchase the Solaris Device Developer's Kit (DDK) that includes the following KCMS CMM manuals:

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

KCMS File System

The software product's directory structure indicates the types and locations of files. Table 1-2 shows you the top-level directories.

Table 1-2 KCMS Directories

Directory	Subdirectory	Content
`/usr/openwin`	`bin`	Configuration and networking binaries
	`demo/kcms`	KCMS demonstration programs
	`demo/kcms/images/tiff`	Sample TIFF images
	`demo/kcms/docs`	KCMS user white papers
	`lib`	`libkcs.so`; main KCMS library
	`share/etc/gpiutils`	CMM libraries
	`share/etc/devhandlers`	Dynamically loadable modules and third-party CMMs
	`share/etc/devdata/profiles`	Device profiles provided with KCMS
	`include/kcms`	Various library header files
	`man/man1`	KCMS command/utility manual pages
	`man/man6`	KCMS demo manual pages
`SUNWsdk/kcms`	`demo`	Sample programs
	`doc`	ICC specification
	`man/man3`	KCMS API manual pages
	`man/man6`	KCMS demo manual pages
	`src`	Sample source code
	`xi_lib`	XIL-based library to read and write TIFF files

1

Sample Programs

Several sample programs demonstrate how to use the API described in this manual. These programs are available online in the `SUNWsdk/kcms/demo` directory. The programs show you how to

- Check profile calibration (`kcms_update.c`)
- Test the loading of a scanner profile and a monitor profile, and correct the color image data (`kcstest.c`)
- Print header attributes in a profile (`print_attributes.c`)

The `/demo` directory also provides files used in the sample programs. These include

- `kcms_create.c`
- `kcmstest_tiff.c`
- `kcms_timer.c`
- `kcms_utils.c`
- `kcms_utils.h`
- `print_header.c`
- `print_montbls.c`

Check the `README_SDK` file for additional information.