Calibrating A Monitor 2

In This Chapter

Chapter 1 introduced the major software components involved in calibration. This chapter describes calibration from the end-user perspective. The end user starts calibration and provides the necessary parameters (such as the device type, what module to load, and various other options for performing the calibration).

Note – The SDK provides two “loadable” source modules that work with Calibrator Tool to gather measurement data: `colorsense` and `XSolarisVisualGamma`. To use the `colorsense` module, the end user must purchase the `colorsense` colorimeter device. `XSolarisVisualGamma` presents the same GUI but does not require a colorimeter device.

The calibration process this chapter describes is how the sample loadable module works with Calibrator Tool. This method of obtaining measurement data may or may not apply to the design of your loadable module.

The chapter is meant to provide you with a better understanding of the sample loadable module. To accomplish this, it refers back and forth between the end user actions and those performed by the underlying software.

2

Before You Read Further

Before you read this chapter, you should review the section entitled "Calibrating Your Monitor" in Chapter 10, "Customizing Your Work Environment," in the *Solaris Advanced User's Guide*. That section provides additional information on calibration not repeated in this chapter, including

- General calibration concepts
- How to adjust the viewing environment before calibrating a monitor
- How to run Calibrator Tool

Revisiting the Calibrator Tool User Interface

Recall that when the end user starts Calibrator Tool, a setup window is displayed. This setup window is shown in Figure 2-1.

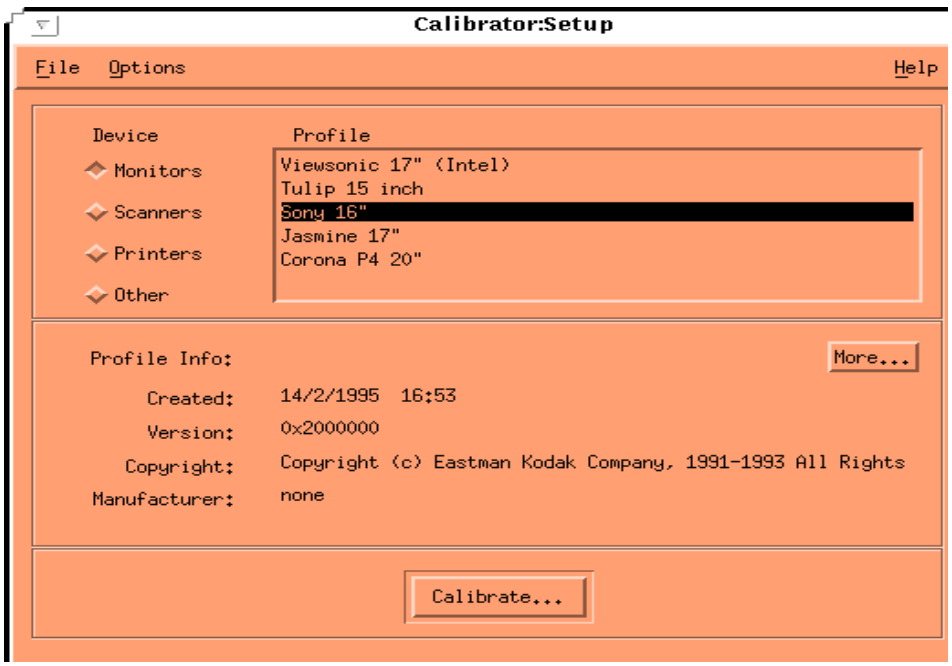

Figure 2-1 Calibrator Tool Setup Window

Table 2-1 summarizes the end-user actions and the actions of the underlying software that lead up to loading the `XSolarisVisualGamma` or `colorsense` module into memory.

Table 2-1 Events Leading Up To Loading Module

End-User Actions	Underlying Software
Starts Calibrator Tool by typing **`kcms_calibrate`** at the command-line prompt and pressing Return.	Calibrator Tool displays the window in `/usr/openwin/bin`. (See Figure 2-1.)
Selects Monitors as the Device type. (See Figure 2-1.)	By default, Calibrator Tool searches the following predefined libraries for monitor profiles: `/usr/openwin/etc/devdata/profiles` `/etc/openwin/devdata/profiles`
Selects the monitor profile (for example Sony 16). (See Figure 2-1.)	KCMS library functions open and obtain the generic device description of the user-selected monitor. For each X11 Window System visual supported by the frame buffer, KCMS library functions make a copy of the generic profile from `/usr/openwin/etc/devdata/profiles` or `/etc/openwin/devdata/profiles`.
Clicks on Calibrate. (See Figure 2-1.)	Calibrator Tool displays the Calibrator:Devices popup window. (See Figure 2-2.)
Selects the `XSolarisVisualGamma` device, for example, and clicks on Load.	The server dynamically loads the `XSolarisVisualGamma` module into memory shared with the Calibrator Tool main module. (The module is expected to be installed in the directory `/usr/openwin/etc/devhandlers`.) Then the loadable module displays its own GUI. (See Figure 2-3.)

The `/usr/openwin/etc/devdata/profiles` directory contains the names of files describing the color characteristics of the generic devices. Currently only monitors are supported by Calibrator Tool. Monitor profile names typically have the extension `.mon`. Profiles exist for other device types as well,

such as scanners (profile names with the extension `.inp`) and printers (profile names with the extension `.out`). The Calibrator Tool window in Figure 2-1 on page 8 also lists another category called Other for additional device types.

Figure 2-2 shows the Calibrator Tool popup window that lists the names of the loadable modules. The SDK includes the source code for sample modules shown in the figure. As previously mentioned, the `colorsense` module listed in the figure uses a colorimeter (hardware puck connected to an RS-232C serial port on the monitor) that the end user would need to purchase to obtain measurement values. `XSolarisVisualGamma` doesn't require this hardware. In either case, the measurement values obtained are used to build a table of color correction data for the particular visual.

The source code for the sample modules is provided in the `/opt/SUNWsdk/kcms` directory. You can use either of these modules as a template to create your own customized version of a loadable module.

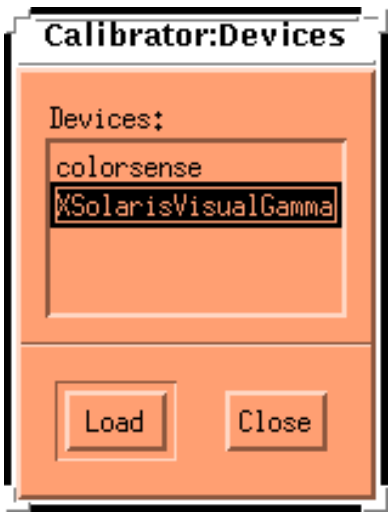

Figure 2-2 Selecting a Loadable Module

When the end user clicks on Load, Calibrator Tool uses the `OWconfig` database file to extract the names of the available loadable modules. You must edit `OWconfig` to add an entry for each loadable module you want to use with

Calibrator Tool. `OWconfig` is the database from which a module is located and later dynamically loaded. For details on `OWconfig` and the format of calibration module entries, see Chapter 3, “OWconfig Database.”

Figure 2-3 shows the GUI that `XSolarisVisualGamma` (or `colorsense`) displays when the end user selects the module from Calibrator Tool’s Calibrator:Devices window.

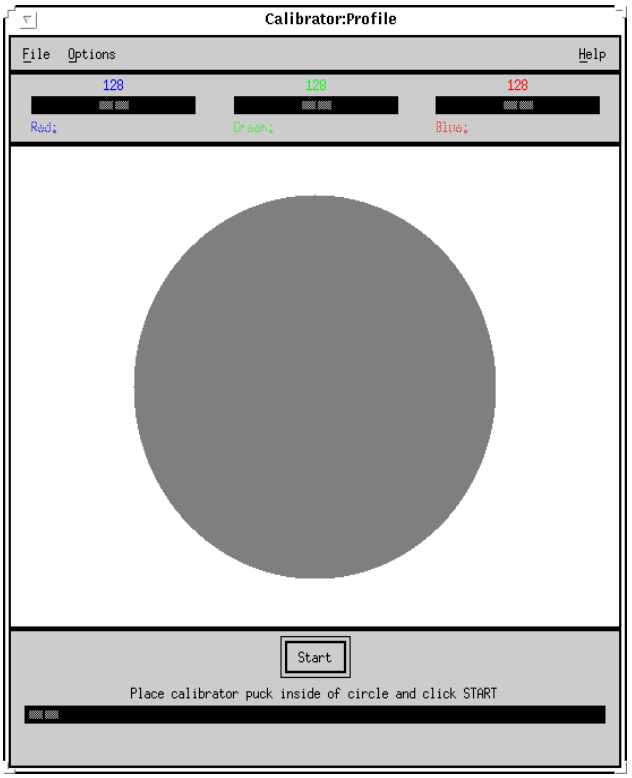

Figure 2-3 Interface to a Loadable Module

The significant point here is that, if your module makes use of a colorimeter, you need to consider a user interface design. The sample modules implement a color patch to obtain data from the monitor and, at the same time, to provide feedback to the user that calibration is actually taking place. Visibly each

module provides feedback on its data gathering activity by displaying the color patch, which changes from cyan to magenta to yellow as the luminance values of each color are being read.

Updating Profiles

Calibrator Tool uses the data it collects from the loadable module to update the X Window System profiles.

Error and Status Messages

See Chapter 10, "Customizing Your Work Environment," in the *Solaris Advanced User's Guide*, for error and status messages that Calibrator Tool displays. For example, a message would be displayed if the loadable module did not finish collecting data. This situation would occur if, for example, the hardware failed to complete measurement or the end user cancelled the measurement. In such a case, the loadable module returns a nonzero status to Calibrator Tool and the entire measurement process must be restarted.

Note – It is up to the writer of the loadable module to provide the end user with any special status or error messages.