Introduction to the XIL Library 1

This introduction discusses three subjects:

- What functionality the XIL library provides
- How the library is meant to work with other Sun, and third-party libraries
- How to verify that the XIL software developer's kit (SDK) has been installed correctly and to set some environment variables that will make using the library easier

Functions in the XIL Library

The functions in the XIL library, which you call as C subroutines, fall into two main categories: image processing and image compression. The next two sections introduce these groups of functions, and a third section summarizes the remaining library functions.

Image-Processing Functions

The image-processing functions in the XIL library can be broadly grouped under the following headings: arithmetic, relational, and logical functions, geometric functions, and some miscellaneous functions. The arithmetic, relational, and logical functions include functions that enable you to add two images, take the maximum pixelwise values of two images, take the logical

AND of two images, multiply an image by a constant, and so on. The geometric functions include, among others, routines to scale, rotate, and transpose images. The miscellaneous functions enable you to:

- Find the minimum and maximum values in an image
- Produce a histogram for an image
- Threshold an image
- Fill an area in an image
- Filter an image
- Detect image edges
- Dilate or erode an image
- Pass an image through a lookup table
- Perform an interband linear combination
- Blend images
- Paint on an image
- Set and get the values of pixels in an image
- Copy a pattern to an image

Table 1-1 indicates where in this book to turn to find more information about the different classes of image-processing functions.

Table 1-1 Finding Information About Image-Processing Functions

Type of Image-Processing Functions	Discussed in This Chapter
Arithmetic, relational, and logical functions	Chapter 9, "Arithmetic, Relational, and Logical Functions"
Geometric functions	Chapter 10, "Geometric Functions"
Other functions	Chapter 11, "Miscellaneous Image Processing Functions"

Digital-Video Functions

The digital-video functions in the library enable you to compress and decompress images and sequences of images. Several compression formats are supported:

- Cell
- CellB
- JPEG baseline sequential
- JPEG lossless
- H.261

- MPEG-1
- CCITT Group 3 and Group 4

The Cell image compression technology, which was developed by Sun, has been optimized for the rapid decompression and display of images on simple hardware. Therefore, the Cell codec is able to achieve reasonable display quality on indexed-color frame buffers. The initial focus of the Cell technology is on Sun-to-Sun communications, where the benefits of fast decoding outweigh the benefits of standards. Possible areas of application include media distributions on CD-ROM and multimedia mail.

The CellB codec, which derives from its Cell counterpart, is intended for use primarily in videoconferencing applications. It features greater balance between the time spent compressing and decompressing images than the Cell codec. The CellB codec's strengths include

- Software compression at interactive rates
- Very fast decoding and display, especially on indexed-color frame buffers
- Low rates of CPU use
- Good quality output

The JPEG compression standards were developed by the Joint Photographic Experts Group to support the compression of still images, both grayscale and color. Although not specifically designed for the compression of sequences of images, or movies, JPEG compressors are also used frequently for that purpose. The JPEG baseline sequential compressor is a *lossy* compressor, which means that it compresses an image in such a way that when the compressed data is decompressed, the decompressed image and the original image may not match exactly. On the other hand, with a *lossless* JPEG compressor, the decompressed image does match the original image pixel for pixel.

The H.261 compression-decompression scheme was developed by the International Telegraph and Telephone Consultative Committee (CCITT). An H.261 video codec is intended to be used to compress and decompress video data sent over Integrated Services Digital Network (ISDN) lines. Thus, the codec is suitable for use in video telephony and videoconferencing applications. The current release of the XIL library includes an H.261 *decompressor*. A compressor can be obtained from a third party.

The MPEG-1 video compression standard was developed by the Moving Picture Experts Group. The group's goal was to compress full-motion video and the associated audio at the rate of about 1.5 Mbits/s. This is approximately the rate at which data can be read from a CD-ROM, so MPEG-1 compressed

video is a good choice for use in interactive multimedia applications. The current release of the XIL library includes an MPEG-1 *decompressor.* A compressor can be obtained from a third party.

The CCITT Group 3 and Group 4 compression standards were developed by the International Telegraph and Telephone Consultative Committee to enable facsimile machines to compress and decompress digitized documents. Now, Group 3 and Group 4 compressors and decompressors are also used for general document storage and retrieval.

Table 1-2 tells you where in this book to look for further information about compressing and decompressing images.

Table 1-2 Finding Information About Compression Techniques

Compression Type	**Discussed in This Chapter**
Cell	Chapter 14, “Cell Codec”
CellB	Chapter 15, “CellB Codec”
JPEG baseline sequential	Chapter 16, “JPEG Baseline Sequential Codec”
JPEG lossless	Chapter 17, “JPEG Lossless Codec”
H.261	Chapter 18, “H.261 Codec”
MPEG-1	Chapter 19, “MPEG-1 Codec”
CCITT Group 3 and Group 4	Chapter 20, “CCITT Group 3 and Group 4 Codecs”

Additional XIL Functions

The image-processing and digital-video functions mentioned above are the heart of the XIL library. However, the library also contains functions that perform the tasks listed in Table 1-3. Table 1-3 also indicates where in this book you can look for further information about the functions that perform these tasks.

Table 1-3 Finding Information About Additional XIL Functions

Additional Tasks You Can Perform	Discussed in This Section or Chapter
Copy an image to the display	"Copying an Image to the Display" on page 97
Rescale an image	"Rescaling an Image" on page 102
Cast an image from one data type to another	"Casting an Image from One Data Type to Another" on page 103
Dither an image	"Dithering an Image" on page 105
Convert an image from one color space to another	"Color-Space Conversion" on page 122
Perform undercolor removal	"Black Generation" on page 125
Handle XIL errors	Chapter 8, "Error Handling"

As you can see, the XIL library contains a broad range of functions. The next section explains why the XIL library was designed this way and how it relates to other Sun, and to third-party, libraries.

The XIL Library: A Foundation Library

The XIL library is what Sun calls a foundation library. Such a library is an application programming interface (API), but has special characteristics that distinguish it from other APIs. For example, a foundation library must deal with hardware dependencies, while most APIs are hardware independent. In addition, a foundation library is geared toward a broad area of application, such as imaging and video (the XIL library) or graphics (the XGL library), while most APIs have a narrower scope. For example, APIs based on the XIL library might address such areas as document imaging, facsimile applications, and digital video (see Figure 1-1).

1

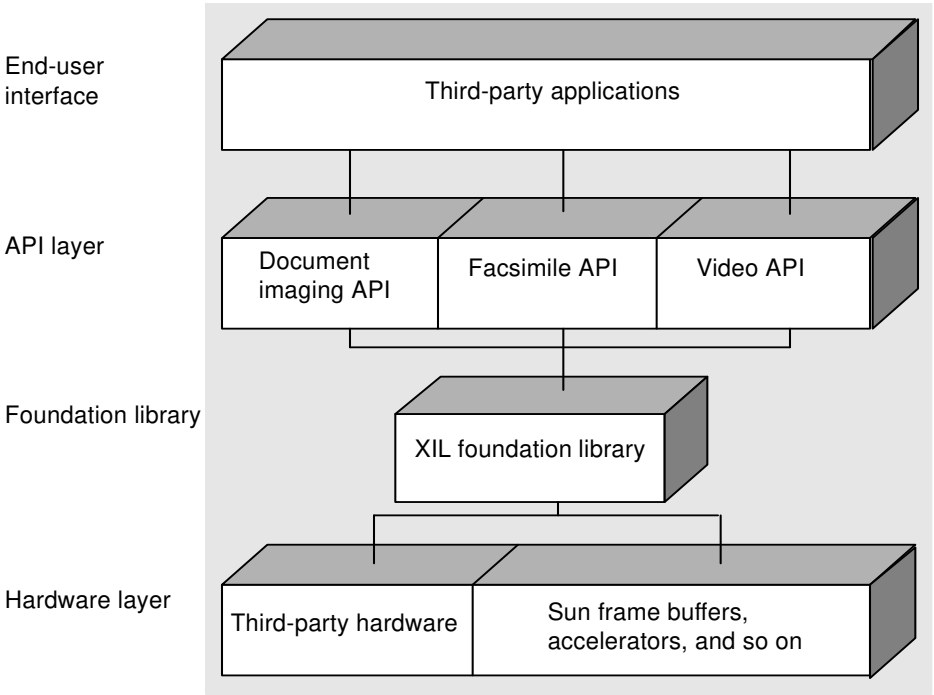

Figure 1-1 Foundation Libraries and Application Programming Interfaces

Why is this extra layer of software desirable? One reason is that a number of APIs—developed either by Sun or third parties—may require the same functionality. For instance, an API directed at the commercial document imaging market and another directed at the facsimile market both must be able to compress and decompress image data according to CCITT standards. If the code that handles this compression is part of a foundation library, the two APIs don't have to create two functions that do the same thing.

Second, a foundation library provides a single interface to the hardware with which the library interacts. For example, the XIL library provides an interface to video cards, scanners, imaging accelerators, frame buffers, and printers. Any applications or APIs written on top of the XIL library access these devices using XIL routines.

Third, a foundation library allows multiple APIs to share objects in a simple way. For instance, because XIL defines memory formats for different types of images, all APIs whose imaging functionality is based on XIL routines can operate on these images without having to convert the format of any image data.

Because an API may also need to provide capabilities that the XIL library does not support, it is also possible to *export* data from the XIL domain into *application space*. There, an application or API can do whatever processing of the data it needs to do. When this processing is complete, the data can be *imported* so that the program can process the data using only XIL functions.

The XIL Library is Multithread Unsafe

In this release, the XIL library doesn't support multithread programs and is therefore multithread-unsafe. Because this is true for all XIL functions, it is not stated in the man pages.

In the next release, the XIL library will support multithread programs. If you want to take advantage of that thread-enabled environment, you must link with the `thread` library when you compile your current XIL programs. Linking to the `thread` library when compiling a current XIL application won't provide any thread-enabled capabilities now, but it will ensure full binary compatibility with the thread-enabled environment later. Applications that don't link `-lthread` will run fine, but they won't take advantage of any thread support included in the release. Linking to the `thread` library now will not degrade your application's performance.

Note – "Building an XIL Program" on page 29 shows the Makefile delivered with one of the XIL example programs; the Makefile shows one way you can link with the `thread` library. The section also discusses the considerations you must make when linking to the library, depending on which compiler and debugger you use.

Verifying Installation and Setting Environment Variables

This section discusses

- The XIL packages that should have been installed
- What the XIL directory structure should look like

1

- The XIL environment variables you should set
- How to use the XIL man pages

XIL Packages

The default installation directory for the XIL SDK software package (`SUNWxilh`) is `/opt/SUNWits/Graphics-sw/xil`; the default installation directory for the XIL AnswerBook package (`SUNWAxi`) is `/opt/SUNWAxi`. In the `SUNWxilh` package, you will find the following subdirectories included at the top level: `examples`, `include`, and `man`. The content of these directories is described in the following sections.

examples/

The `examples` directory contains several examples that are discussed later in this book.

Table 1-4 Example Programs Provided with the XIL Developer's Release

Example Program	What It Does
`example1`	Reads a grayscale image from a file and displays the image in an Xlib™ window
`display`	Demonstrates how to display images of different data types and with different numbers of bands in an Xlib window
`encode`	Demonstrates how to create movies using XIL functions
`xilcis_example`	Demonstrates how to play back movies using XIL functions

include/

This directory has one subdirectory, `xil`, containing the include (`*.h`) files required by the XIL library.

`man/`

This directory has one subdirectory, `man3`, containing the on-line manual pages (accessible via the `man(1)` command) for the XIL library. These manual pages are also part of the XIL AnswerBook on-line documentation.

AnswerBook Package

The `SUNWAxi` package contains the XIL AnswerBook on-line documentation, including*:*

- *XIL Programmer's Guide*
- *XIL Reference Manual*

XIL Directory Structure

Figure 1-2 shows the directory structure of the XIL developer's kit. Directories shown in bold print are part of the SDK packages. The `lib` directory, containing the XIL Runtime Environment, is released on the Solaris™ CD-ROM disk.

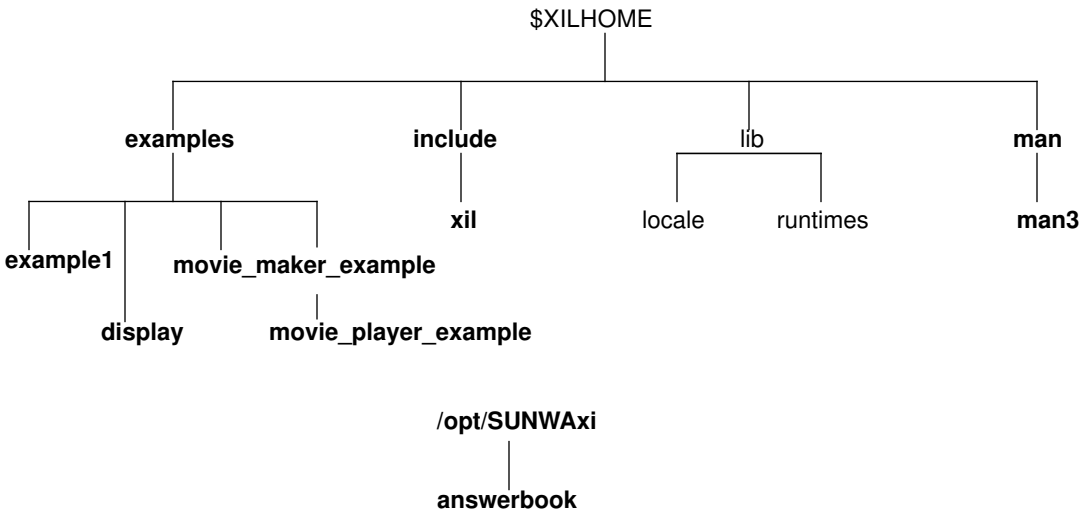

Figure 1-2 Directory Structure of XIL Developer's Kit.

1

Setting Environment Variables

You need to set the XILHOME and LD_LIBRARY_PATH environment variables as follows. (The example instructions assume that the XIL library is installed in the default directory.)

1. **Set the** XILHOME **environment variable to the directory containing the XIL libraries:**

```
% setenv XILHOME /opt/SUNWits/Graphics-sw/xil
```

2. **Set** LD_LIBRARY_PATH **to include** $XILHOME/lib**:**

```
% setenv LD_LIBRARY_PATH $XILHOME/lib:$LD_LIBRARY_PATH
```

Using the XIL Manual Pages

To access the XIL reference manual pages, you must modify the MANPATH environment variable to include the directory where the man pages are located. The optional catman(1M) command creates preformatted versions of the man pages and recreates the windex database.

1. **Set the** MANPATH **environment variable.**

   Assuming you have already set the XILHOME environment variable, set MANPATH as follows:

```
% setenv MANPATH $MANPATH:$XILHOME/man
```

2. **(Optional) Format the man pages.**

```
% su root
Password: <enter your root password>
# /usr/bin/catman $XILHOME/man 3
```