

Solaris 2.4 Software Developer Kit Introduction

2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.



SunSoft
A Sun Microsystems, Inc. Business

© 1994 Sun Microsystems, Inc.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX[®] and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's font suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, the Sun logo, Sun Microsystems, Sun Microsystems Computer Corporation, SunSoft, the SunSoft logo, Solaris, SunOS, OpenWindows, DeskSet, ONC, ONC+, SunExpress, SunSolve Online, SunPro, ProCompiler, ProWorks, OpenBoot, XView, ToolTalk, XGL, XIL, Solaris Visual, AnswerBook, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. CatalystSM and SunSpectrumSM are service marks of Sun Microsystems, Inc.. UNIX is a registered trademark of Novell, Inc., in the United States and other countries; X/Open Company, Ltd., is the exclusive licensor of such trademark. OPEN LOOK[®] is a registered trademark of Novell, Inc. PostScript and Display PostScript are trademarks of Adobe Systems, Inc. Motif[®] is a registered trademark of Open Software Foundation, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCstorage, SPARCware, SPARCcenter, SPARCclassic, SPARCcluster, SPARCdesign, SPARC811, SPARCprinter, UltraSPARC, microSPARC, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and SunTM Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a product of the Massachusetts Institute of Technology.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.



Contents

1. Introduction	1
Software Developer Kit Overview	1
How the SDK Fits Into a Solaris Development Environment .	3
New SDK Features	3
On-Line Demonstration	3
To Display the Demonstration	4
From File Manager	4
From the Command Line	6
C and C++ Compiler Try-and-Buy CD-ROM Disc	7
Internationalization	7
2. SDK Components	9
Multithreaded Programming	9
STREAMS Framework	9
Network Applications	10
New Features	10

ToolTalk Messaging Services	11
VISUAL for Solaris	12
Solaris XGL 3.1 Graphics Library	13
Solaris PEX 2.2	13
Solaris XIL 1.2 Imaging Library	14
Open Windows Libraries and OpenWindows X Server 3.4	15
New Features	16
Windows Graphical User Interface Toolkits	16
Motif Toolkit	16
New Features	18
XView Toolkit	18
OPEN LOOK Intrinsic Toolkit	18
OpenWindows 3.0.1 Developer's Guide	19
Devguide Motif Utilities	20
Supplemental On-line Documentation	20
3. Technical Support, Training, and Documentation	23
Technical Support	23
Sun Educational Services	24
A Complete List of SDK Documentation	25
Documents Available Through the AnswerBook Product	25
Solaris 2.4 Software Developer AnswerBook	25
OpenWindows Developer Guide AnswerBook	26
Solaris XGL 3.1 AnswerBook	26

Solaris XIL 1.2 AnswerBook	26
Solaris 2.4 Supplemental Developer AnswerBook. . .	26
Solaris 2.4 Reference Manual AnswerBook	27
Documents Available Through PostScript Files.	27
Documents Available in Hard Copy Only	28
On-Line Documents Also Available in Hard Copy	28
Third-Party Documents	29

Preface

The *Software Developer Kit Introduction* gives an overview of the Solaris™ 2.4 Software Developer Kit (SDK). It also:

- Tells you how the SDK fits into a Solaris development environment.
- Lists the new SDK features in this release of Solaris.
- Describes each component of the SDK.
- Lists and gives a brief description of SDK on-line documentation.
- Tells you how to obtain technical support and technical training.

Who Should Use This Book

If you are a software developer who is creating applications for Solaris 2.4, you should read this book. Typical software developers are independent software vendors (ISVs) or original equipment manufacturers (OEMs) who want their applications to run in a Solaris environment.

Before You Read this Book

Before you install the SDK, you must have one of the following versions of Solaris 2.4 installed on your x86 or SPARC platform:

- x86 Enterprise Server
- x86 Workgroup Server
- x86 DeskTop
- SPARC DeskTop
- SPARC Workgroup Server
- SPARC Enterprise Server

You should also have the AnswerBook® product (the on-line version of the documentation) installed. The AnswerBook product is provided with any version of Solaris 2.4. Once you have a version of Solaris 2.4 installed, read the manuals that are provided in the SDK box. These include:

- *Software Developer Kit Introduction* (this manual)
- *Software Developer Kit Open Issues and Late-Breaking News*
- *Driver Developer Kit Installation Guide*

You can then install the SDK using the *Software Developer Kit Installation Guide*.

What Typographic Changes and Symbols Mean

The following table describes the type changes and symbols used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. system% You have mail.
AaBbCc123	What you type, contrasted with on-screen computer output	system% su Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You must be root to do this.

Code samples are included in boxes and may display the following:

%	UNIX C shell prompt	system%
\$	UNIX Bourne and Korn shell prompt	system\$
#	Superuser prompt, all shells	system#

Introduction

1 

Welcome to the *Solaris 2.4 Software Developer Kit* (SDK). This chapter introduces you to the SDK and tells you how it fits into a Solaris development environment. It also lists features that are new to the SDK in the Solaris 2.4 release. In addition, you are introduced to the SDK on-line demonstration and the C and C++ *Try-and-Buy* CD-ROM disc. This chapter also gives a brief overview of Solaris internationalization.

Software Developer Kit Overview

The SDK provides the software tools, documentation, and training and support information you need to develop software applications for the Solaris 2.4 release.

Solaris 2.4 technologies, such as multithreaded programming and *ToolTalk™* *Messaging Services* along with SDK technologies, such as the XGL™ graphics library. The SDK on-line documentation make up the *components* of the SDK.

These components are provided either on the Solaris CD-ROM disc or on the SDK CD-ROM disc. The SDK components and the CD-ROM disc that they are delivered on are listed in Table 1-1.

Table 1-1 SDK Components

SDK Component Name	CD-ROM Disc Location
Multithreaded programming	Solaris
STREAMS framework	Solaris
Network applications	Solaris
ToolTalk Messaging Services	Solaris
Software developer support for <i>VISUAL™ for Solaris</i> :	
•XGL graphics library	SDK*
•Solaris PEX™	SDK*
•XIL™ imaging library	SDK*
•Display PostScript™ libraries	Solaris
•OpenWindows™ X Server and libraries	Solaris
Windows Graphical User Interface Toolkits:	
•Motif®	SDK*
•XView™	Solaris
•OPEN LOOK® Intrinsic Toolkit (OLIT)	Solaris
On-line documentation	SDK
SDK on-line demonstration	SDK

*The runtime environments for these SDK components are on the Solaris CD-ROM disc.

Except for the SDK on-line demonstration, which you may want to read about in “On-Line Demonstration” on page 3, the SDK components are explained further in Chapter 2, “SDK Components.”

How the SDK Fits Into a Solaris Development Environment

Solaris developers produce applications, drivers, and graphics handlers that are ready for the end-user Solaris runtime environments. A Solaris development environment may be constructed using the Solaris runtime environments (available with any version of the Solaris 2.4 release), developer kits (the Solaris 2.4 Software Developer Kit and Solaris 2.4 Driver Developer Kit), and compilers (the ProCompiler™ and SPARCompiler™ C and C++).

The SDK contributes to this environment by providing the on-line documentation, development software libraries, productivity tools, sample code, and testing tools you need to develop software applications for the Solaris runtime environments. For more information on the Solaris 2.4 release, see the *Solaris 2.4 Introduction*.

New SDK Features

The following features are new to the Solaris 2.4 SDK:

- The SDK software is now available for both the x86 and SPARC architectures.
- The SDK documentation is now merged so that it provides uniform support for both the x86 and SPARC architectures.
- The SDK now has an on-line SDK demonstration (see the following section “On-Line Demonstration”).

New features for each component of the SDK are listed in Chapter 2, “SDK Components.”

On-Line Demonstration

The on-line SDK demonstration is a series of on-line technology demonstrations and documentation that feature the SDK components. This demonstration includes:

- Technology demonstrations
- Source code for each demonstration
- Technical white papers
- Technical newsletters
- Referrals to helpful third-party books.

Please preview this on-line demonstration for a better understanding of features and functions offered by the SDK components. A Motif graphical user interface (GUI) provides you with easy access to each demo, white paper, and newsletter.

To Display the Demonstration

1. **Insert the caddy containing the SDK CD-ROM disc into the drive.**

Note – If you have the volume management software enabled (the default with the Solaris 2.4 release), inserting the CD-ROM disc mounts it automatically. If you have disabled volume management, mount the CD-ROM disc using the `mount(1M)` command.

Note – If you are installing from a remote CD drive, refer to the *Software Developer Kit Installation Guide* for instructions on installing the SDK from a remote CD drive.

Note – To run the SDK on-line demo, your system must support Solaris audio output (that is, `/dev/audio` must point to a valid audio device).

File Manager opens, displaying the contents of the CD-ROM disc, including the `START_INTRO` icon.

From File Manager

2. **Click twice on the `START_INTRO` icon**

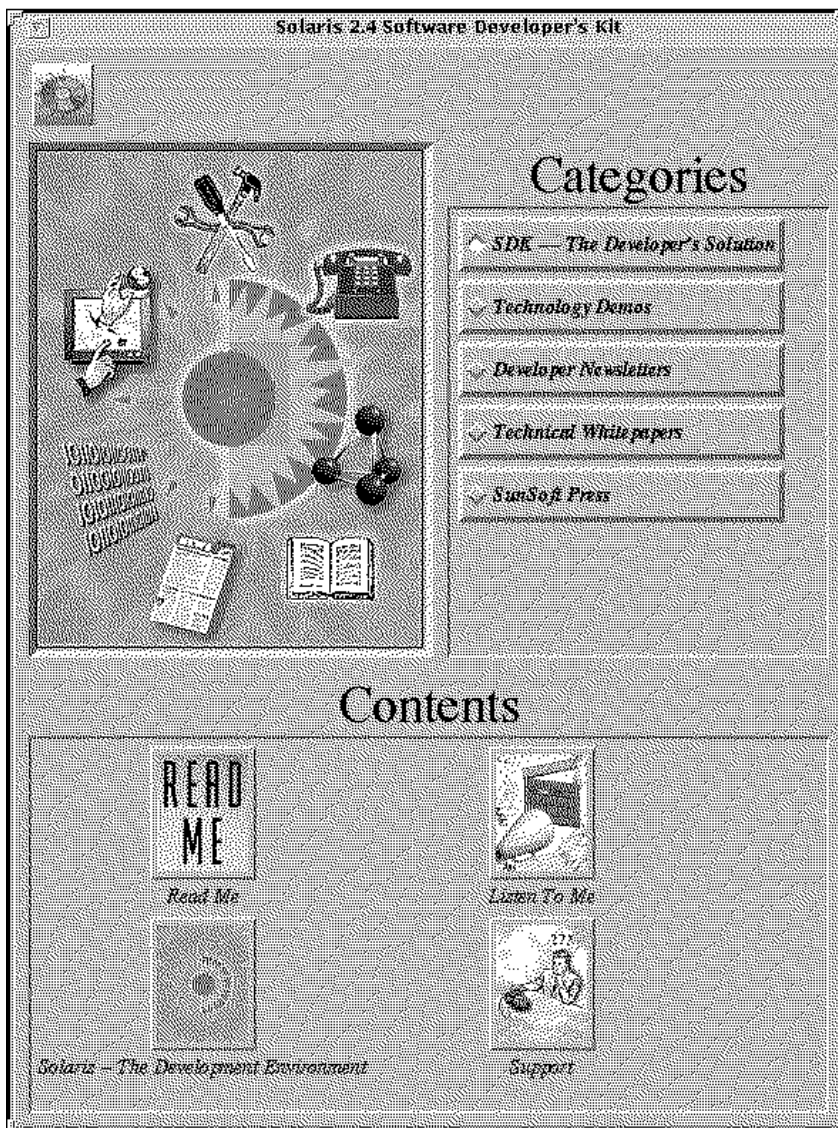
A dialog box opens and asks if you want to run the demonstration in a shell tool or without one.

3. **Choose “No Shell Tool”**

In a few moments, you will see a window entitled *Welcome to the Solaris 2.4 Software Developers’s Kit*. Wait a few more moments and the window displaying the SDK-demonstration *Categories* and *Contents* appears (see the following figure).



Note - If you get the error message `sh: START_INTRO not found`, it's because you don't have "." in your path. You must run the demonstration from the command line. See the following section "From the Command Line."



From the Command Line

1. From the command line enter:

```
/cdrom/cdrom0/START_INTRO
```

In a few moments, you will see a window entitled *Welcome to the Solaris 2.4 Software Developer's Kit*. Wait a few more moments and the window displaying the SDK-demonstration *Categories* and *Contents* appears.

From here you can choose to look at:

- The SDK overview, called *SDK—The Developer's Solution*
- Technology demos
- Developer newsletters
- Technical white papers
- SunSoft Press

As you choose a category, the *Contents* section of the window changes to reflect the category that have chosen. If you choose the *Technology Demos* category, you will always see the following icons within a component selection, for example, if you click on the XGL icon, you will see the following icons:

- The executable file that starts the demo, for example *XGL Demo*.
- The *Listen To Me* icon—An audio description of the technology.
- The *Read Me* icon—Describes the system requirements and gives an explanation of the demonstration.
- The *What's Next* icon—Tells you what to do next, for example, install software, read certain white papers, read third-party books, location of source code for the demonstration.

You will also see, at the bottom of the screen, a written overview of a component (for example, an overview of XGL).

Note - The SDK on-line technology demonstrations and corresponding source code are provided as a technology demonstration only. SunSoft, Inc. does not support the demonstrations nor the source code. SunSoft, Inc. makes no commitment to enhance the demo or fix any bugs that may exist in the code.

C and C++ Compiler Try-and-Buy CD-ROM Disc

The Solaris 2.4 SDK box includes a copy of the SunPro *Try-and-Buy* Product. *Try-and-Buy* lets you evaluate any or all of the following SunPro Professional products for 30 days:

- SPARC® or ProCompiler™ Language Systems
- SPARC or ProWorks™ Development Environment
- SPARC or ProWorks TeamWare code management tools

You can begin your free 30 day evaluation with a phone call. Read the enclosed *SPARC/ProWorks Product Evaluation and Installation Guide* for information on receiving an evaluation license password. When you receive your password you can follow the installation instructions and begin your work immediately.

Internationalization

The Solaris development environment is fully internationalized, enabling you to build applications that conform to level-4 internationalization. This means that this Solaris release:

- Is 8-bit clean.
- Adheres to standard formats and collation in displaying:
 - Money
 - Numbers
 - Date
 - Time of day.
- Supports messaging in native locale.
- Supports multibyte characters allowing for support of Asian character sets.

Using the internationalized features of the Solaris development environment, you can create applications that can be localized for any language. See the *Developer's Guide to Internationalization* for more information on Solaris internationalization. This guide describes both internationalization and localization of applications, formats, and conventions in different countries, internationalization facilities and their use, and various special considerations.

The SDK is made up of individual software technologies and supporting documentation. The software technologies and the supporting documentation are called the *components* of the SDK. This chapter describes each SDK component and tells you about the documentation associated with that component. If a component has features that are new in the Solaris 2.4 release; they are listed. At the end of this chapter, documentation that supports software development, but is not closely associated with a software component of the SDK, is listed and described.

Multithreaded Programming

The operating system kernel is multithreaded, allowing for asymmetric, multiple processors to execute the kernel at the same time. This allows for application concurrence, or the execution of applications written as independent computations. The multithreaded kernel is on the Solaris CD-ROM disc. For further information, see the *Multithreaded Programming Guide*, which introduces multithread programming concepts and describes basic use of threads, synchronization functions, and the Solaris two-level model.

STREAMS Framework

STREAMS, a framework for character I/O, is implemented throughout the operating system. The STREAMS framework provides you with:

- Integral functions.
- A set of utility routines.
- Facilities that expedite software design and implementation.

For further information on the STREAMS framework, see the *STREAMS Programmer's Guide* which provides information on the use of the STREAMS mechanism at user and kernel levels.

Network Applications

In a Solaris development environment, you can write network applications without coding the details of specific protocols and address formats. This is network transport independence. The Solaris programming environment provides these capabilities through the ONC+ facilities. ONC+ is a family of network protocols for connecting large heterogeneous networks.

Remote procedure call (RPC), which is included in ONC+, enables an application to execute procedures on remote systems over a network. The transport layer interface (TLI) makes a protocol-enhanced RPC enabling RPC programs to run across multiple network transports. External data representation (XDR) in ONC+ allows applications to exchange data across heterogeneous hardware systems. The `rpcgen` tool for RPC is included as a standard part of Solaris programming environment. You can use the `rpcgen` tool to generate templates for client programs and server routines in a high-level, C-like language called RPC language. The code generator creates the necessary RPC code to be included in the client and server pieces of the application.

For further information, see the *Network Interfaces Programmer's Guide*, which describes the use of basic, intermediate, and high-level facilities to:

- Develop distributed applications
- Convert existing applications from single-computer to distributed-computer
- Maintain distributed applications in the Solaris 2.x environment.

New Features

There is a new networking interface in the Solaris 2.4 release. It is the *multithreaded-safe server* remote procedure call (RPC) interface. The *multithreaded-safe client* RPC interface was new in the Solaris 2.3 release. You may invoke these interfaces to put a server into one of two multithreaded

modes: *auto mode* or *user mode*. The auto mode is easy to use and powerful. The user mode requires that you know more about multithreaded interfaces and issues, but offers more flexibility. These RPC interfaces are described in the *Network Interfaces Programmer's Guide*.

ToolTalk Messaging Services

The ToolTalk service included in the Solaris development environment is a network-spanning, interprocess messaging system. Applications can create, send, and receive messages by calling ToolTalk functions from the ToolTalk application-programming interface (API) library. The ToolTalk service enables independent applications to communicate with other applications without having direct knowledge of each other. The ToolTalk service allows one application to effectively be replaced with another application as long as both applications support the same ToolTalk interface. For further information on the ToolTalk service, see:

- *ToolTalk User's Guide*—This guide describes ToolTalk. It discusses ToolTalk functionality, including modifying applications to use ToolTalk. It describes ToolTalk messaging and message sets. Use this guide if you are creating or maintaining ToolTalk applications, or if you are a system administrator supporting ToolTalk developers.
- *ToolTalk Reference Manual*—This manual describes the API components, commands, and error messages.
- *ToolTalk White Paper*—This paper provides a brief and high-level discussion of ToolTalk features and benefits. Read this paper if you want a quick overview of ToolTalk.
- *Introduction to the ToolTalk Service White Paper*—This paper gives an introduction to ToolTalk. It discusses how ToolTalk can fit into a system and a product strategy. This paper is for non-developers who want to understand what ToolTalk is and how it can relate to their products.
- *The ToolTalk Service White Paper*—This papers gives a technical introduction to ToolTalk, models, services, and terminology. It contains an overview of all basic ToolTalk capabilities, what they do, and how you would use them. Read this paper if you want to evaluate or use ToolTalk.
- *Designing and Writing a ToolTalk Procedural Protocol*—This paper describes the general principles for designing, writing, and implementing ToolTalk protocols.

- *The ToolTalk Desktop Services Message Set White Paper*—This paper describes a basic set of ToolTalk messages that are designed to facilitate control and cooperation among clients on a desktop (for example, launching, halting, control, or passing output from one client to another).
- *The ToolTalk Document and Media Exchange Message Set White Paper*—This paper describes a set of ToolTalk messages designed to facilitate the exchange of multimedia data (for example, graphics, sound, video) among cooperating applications.
- *A Simple Demonstration of How the ToolTalk Service Works*—This paper explains how to design and build applications that interoperate with other applications that are using the ToolTalk service.
- *The ToolTalk Service and Project DOE Relationship*—This paper describes the relationship between the ToolTalk Service and the direct object environment (DOE) project at SunSoft.
- *The Messaging Object Service*—This paper describes the messaging-object-service for project-DOE objects and how you can use this service in your object oriented applications. It also describes how applications can send, specify, and receive other applications.

VISUAL for Solaris

The *VISUAL for Solaris* environment includes several graphics and multimedia software *foundation libraries*. Foundation libraries are the lowest-level device-independent layer of the Solaris software. This level of interface is designed to support a wide variety of common functions. You can build higher-level libraries on top of the foundation libraries, or the foundation libraries can be used directly by a software application. These foundation libraries enable you to create applications that incorporate 2-D and 3-D graphics, imaging, and digital video. The libraries are the XGL graphics library (which serves as a foundation library for Solaris PEX) and the XIL imaging library.

The *VISUAL for Solaris* environment also includes the OpenWindows X Server and libraries (X11/Xlib) and the Display PostScript (DPS) libraries.

For further information on the *VISUAL for Solaris* environment, see the *Solaris Visual White Paper*, which gives an overview and philosophy of the *VISUAL for Solaris* environment.

Solaris XGL 3.1 Graphics Library

The *Solaris XGL Graphics Library* is a graphics interface, providing foundation-level geometry graphics support for Solaris-based applications. The XGL graphics library is both an API and a geometry foundation library that provides:

- A set of low-level graphics primitive operations for 2-D and 3-D geometric applications.
- High performance rendering capabilities for a wide variety of graphics operations across numerous graphics hardware platforms.
- An API that presents an immediate-mode graphics programming model.
- Support for graphics-based applications, such as CAD programs.
- A foundation library for higher-level graphics APIs, such as PEXlib, PHIGS, and GKS.

The XGL runtime library is delivered on the Solaris 2.4 CD-ROM disc. The XGL developer libraries and documentation are delivered on the SDK CD-ROM disc. For further information on the XGL graphics library, see the following documents:

- *XGL Programmer's Guide*—This guide provides information and example programs that you can use if you are writing applications using the XGL graphics library.
- *XGL Reference Manual*—This manual includes the manual pages that describe the syntax for using the operators (functions) and attributes in the XGL graphics library.
- *The XGL White Paper*—This paper describes the purpose, structure, and features of functions in the XGL graphics library.

Solaris PEX 2.2

The PEX runtime environment is an implementation of the X Consortium PEX specification. The PEX client-side runtime library (PEXlib) is an extension to the X Window System for support of fully accelerated distributed 3-D graphics.

PEX software consists of PEXlib and a PEX server-extension shared-object file. PEXlib is an application programmer interface to the PEX protocol in the same way that Xlib is an application programmer interface to the core X protocol.

PEXlib provides application portability across hardware platforms and enables 3-D graphics rendering on local and remote displays. The PEX runtime packages are on the Solaris 2.4 CD-ROM disc.

Note – See the *Software Developer Kit Installation Guide* for special instructions on how to install the PEX software packages.

The PEXlib implementation for the Solaris 2.4 release is based on the *PEXlib—Specification and C Language Binding: Version 5.1*, which is available from the X Consortium. The PEX server is based on the *PEX Protocol Specification, Version 5.1P*. Portable additions to PEXlib and the PEX server are from the Common Open Software Environment's *Common Graphics Environment (CGE) PEX 5.1 Specification*. For further information on PEX, see the following documentation:

- *PEXlib Programming Manual*, O'Reilly and Associates, Inc.—This manual describes PEXlib, the programming library for 3-D graphics, and how to use it. It also recommends how to write PEXlib programs for better interoperability across any vendor's PEX implementation.
- *PEXlib Reference Manual*, O'Reilly and Associates, Inc.—This manual contains manual pages for each PEXlib routine.
- *Solaris PEX 2.2 Implementation Specification*—This specification describes PEX software implementation and limitations.
- *CGE PEX 5.1 Portability Guide*—This guide assists you in creating portable and interoperable PEXlib applications across platforms supporting the CGE.

Solaris XIL 1.2 Imaging Library

The XIL 1.2 Imaging Library is a foundation library that provides image processing and digital-video applications. The ISV interface enables you to access the functions in the library as C subroutines. The XIL imaging library supports:

- Image processing
- Image compression
- A wide variety of general digital video applications
- Higher level APIs

All users who are running the Solaris operating system can run XIL applications because the XIL runtime environment is included with the Solaris software. For further information on the XIL imaging libraries, see:

- *XIL Programmer's Guide*—This guide describes, in detail, how to use the XIL imaging library to create imaging or video applications. It includes discussion of input, output, and error handling. It also describes image-processing functions and covers using the XIL imaging library to compress and decompress sequences of digital images.
- *XIL Reference Manual*—This manual contains all the manual pages available with the XIL imaging library.
- *The XIL White Paper*—This paper describes the purpose, structure, and features of functions in the XIL imaging library.

Open Windows Libraries and OpenWindows X Server 3.4

OpenWindows, the X-based window system environment, is a standard part of the Solaris development environment. The OpenWindows environment is network-based, offering the X protocol interface. An application that you write in this environment can take advantage of multiple machines distributed over the network. The OpenWindows libraries (X11/Xlib, and Display PostScript) and the OpenWindows X Server are on the Solaris CD-ROM disc. For more information on OpenWindows libraries and the OpenWindows X Server, see the following manuals:

- *OpenWindows Server Programmer's Guide*—This guide instructs you in writing applications that are based on X11 and Display PostScript and that run in the OpenWindows environment.
- *OpenWindows Reference Manual*—This manual contains the manual pages for the OpenWindows environment.

New Features

With the Solaris 2.4 release, the following OpenWindows X Server features are new:

- Support for x86 and SPARC architectures
- An addition to the graphics porting interface (GPI) to support transparent overlays
- An enhanced direct graphics access (DGA) drawable interface
- OWconfig file format enhancements

Windows Graphical User Interface Toolkits

There are three window graphical user interface (GUI) toolkits available for application development:

- Motif
- XView
- OLIT

The Motif toolkit supports the Motif graphical user interface and is delivered through the SDK. XView and OLIT support the OPEN LOOK graphical user interface and are delivered as a standard part of the Solaris release. All three toolkits provide the routines and libraries for designing user interfaces and handling events such as mouse pointing and user input.

The Motif toolkit is the SunSoft strategic-toolkit for the future. If you are planning to develop new applications, SunSoft strongly encourages you to use the Motif toolkit. If you have existing applications based on XView or OLIT you can continue to do application enhancement or maintenance using the XView or OLIT toolkits.

Motif Toolkit

The Motif toolkit gives you a head start in creating applications for the new *Common Desktop Environment* (CDE). It includes a comprehensive set of tools that include the:

- Motif libraries
- Motif window manager
- User interface language (UIL) compiler

The Motif toolkit is based on the industry standard *X Window System* and *X Toolkit Intrinsics* from the *X Consortium*. It works with X11R5 libraries, making it portable across a range of hardware platforms. The Motif toolkit also offers a set of drag-and-drop operations, enabling you to create graphical user models for data selection and transfer. The UIL compiler enables rapid user-interface design and prototyping without requiring massive amounts of code, significantly reducing application development time.

The Motif runtime libraries and header files are on the Solaris 2.4 CD-ROM disc in the `SUNWmfrun` package. The `SUNWmfrun` package is in the *All* or *Developer* cluster of the Solaris 2.4 release. For further information on the Motif toolkit, see the following documents:

- *OSF Application Environment Specification (AES) User Environment Volume*, Revision C, PTR Prentice Hall—This specification describes how to use the Open Software Foundation (OSF) and Motif GUI. It describes how to manipulate windows on-screen, use applications written for the Motif environment, and customize the Motif desktop environment.
- *OSF/Motif Programmer's Guide*, Release 1.2, PTR Prentice Hall—This guide describes how to use the OSF/Motif API to create Motif applications. It gives an overview of the Motif widget-set architecture, explains Motif toolkit features, and presents a model and examples for constructing Motif applications.
- *OSF/Motif Programmer's Reference*, Release 1.2, PTR Prentice Hall—This manual describes the API for the OSF/Motif GUI. It is a reference manual for OSF/Motif commands and functions (it describes all toolkit, window manager, and user interface language commands and functions).
- *OSF/Motif 1.2 Style Guide*, PTR Prentice Hall—This guide describes the behavior specifications of the OSF/Motif GUI. It guides application developers, widget developers, user interface system developers, and window manager developers in the design and implementation of new products consistent with the OSF/Motif user interface.
- *OPEN LOOK to Motif GUI Transition Guide*—This guide describes how to migrate from OPEN LOOK to Motif with the Motif 1.2.3 toolkit.

New Features

With the Solaris 2.4 release, the following Motif Toolkit features are new:

- The Motif release is upgraded from 1.2.2 to 1.2.3.
- The API and form widget implementation are now completely in line with the OSF standard. See the *OPEN LOOK to Motif GUI Transition Guide* for more information.

XView Toolkit

The XView toolkit is an X11 Windows System-based user-interface toolkit. It supplies you with a set of prebuilt OPEN LOOK objects such as buttons and menus. You can use these objects to build OPEN LOOK-compliant applications. See the *XView Developer's Notes*, which include XView developer information not present in the O'Reilly XView documentation set. These notes contain two parts:

- Motif interoperability and XView panel issues
- Internationalizing XView applications.

OPEN LOOK Intrinsic Toolkit

The OPEN LOOK Intrinsic Toolkit (OLIT) is an X Window System-based widget set and library that you can use to create applications supporting the OPEN LOOK graphical interface. Widgets are user interface objects such as:

- Buttons
- Scrollbars
- Control areas
- Text edit areas
- drawing areas

When you create and manipulate widgets with the OPEN LOOK Intrinsic Toolkit, you can create an OPEN LOOK user interface for your application. See the following documents for more information on OLIT:

- *OLIT QuickStart Programmer's Guide*—This guide presents fundamental OLIT programming concepts and provides a foundation for more advanced OLIT topics.

- *OLIT Reference Manual*—This manual provides descriptions of the OLIT widget set and related convenience functions that you can use with the widget set. It guides you in developing internationalized OLIT applications using these widgets and functions.

OpenWindows 3.0.1 Developer's Guide

The OpenWindows Developer's Guide (Devguide) is a GUI builder for developing OPEN LOOK applications. It is designed to make your job easier. Devguide is a programming tool that lets you create and try user interfaces without writing any code. Because you can create and modify an interface easily, you can concentrate on designing and testing interfaces and enhancing functionality, rather than on coding.

Once you finish with interface design, you can save significant time and reduce cost by using one of the included code generators, `gxv` or `golit`, to generate XView or OLIT application code.

Most of the principles common to the OpenWindows environment are present in Devguide. You can select, drag, and drop objects using a mouse to quickly create and test an interface design before any code is created. Devguide enables you to create applications that integrate with the Solaris DeskSet™ environment, not only in look and feel, but also in interoperability. This capability provides you with a consistent and easy-to-use desktop.

Devguide also enables you to create applications that can be easily localized for the international market. For more information on Devguide see the following documents:

- *OpenWindows Developer's Guide: User's Guide*—This guide provides a comprehensive description of Devguide.
- *OpenWindows Developer's Guide: Motif Conversion Utilities Guide*—This guide describes `GMF` and `GUIL` utilities used with Devguide user interface GIL files to produce Motif C code or Motif UIL files.
- *OpenWindows Developer's Guide: XView Code Generator Programmer's Guide*—This guide describes `gxv` and `gxv++`, the XView toolkit code generators used with Devguide user interface GIL files to produce XView C code.

- *OpenWindows Developer's Guide: OLIT Code Generator Programmer's Guide*—This guide describes `golit`, the OLIT code generator used with Devguide user interface GIL files to produce OLIT C code.

Devguide Motif Utilities

Devguide Motif utilities enable you to migrate Devguide-generated OPEN LOOK applications to Motif. The code generators, `gmf` and `guil`, take the Devguide user interface GIL files and output Motif C and Motif UIL source code, respectively. You can continue to add C code to the Motif or UIL code to complete an application. The generated UIL Motif code can also be fed back into a third party Motif GUI builder for further development and testing. For further information on Devguide Motif utilities, see the *OPEN LOOK to Motif GUI Transition Guide*, which describes how to migrate from OPEN LOOK to Motif applications with the Motif 1.2.3 toolkit.

Supplemental On-line Documentation

In addition to the documents described in the component sections of this chapter, the following on-line documentation is also provided with the SDK. This documentation is available in three on-line documentation sets called:

- *Solaris 2.4 Software Developer AnswerBook*
- *Solaris 2.4 Reference Manual AnswerBook*
- *Solaris 2.4 Supplemental Developer AnswerBook*

For a complete list of SDK documentation, see “A Complete List of SDK Documentation” on page 25.

These documents are in the *Solaris 2.4 Software Developer AnswerBook* on-line documentation set:

- *System Services Guide*—This guide describes the operating-system services. It also provides information for experienced programmers and system administrators writing or administering real-time applications.
- *Standards Conformance Reference Manual*—This manual describes how the Solaris programming environment and the operating system comply to such industry standards as the:
 - *Application Binary Interface, X/Open™*
 - *IEEE Standard 1003.1 (Posix.1)*

- *SPARC Compliance Definition*

If you adhere to the guidelines in this manual, you can write applications that are consistent across conforming systems.

- *Programming Utilities Guide*—This guide describes the following built-in operating system programming tools: `m4`, `make`, `sccs`, `lex`, `prof`, `yacc`, and System V `make`.
- *SPARC Assembly Language Reference Manual*—This manual describes the SPARC Assembler. It includes the Assembler syntax, executable and linking format, conversion of files, and instruction-set mapping.
- *Linker and Libraries Guide*—This manual describes the operations of the operating system linker and libraries, summarizes their mechanisms, and describes how to use them to build executable programs and libraries.
- *Solaris Binary Compatibility Guide*—This guide describes how to use the binary compatibility package, which supports SunOS 4.x applications in the SunOS 5.4 environment.
- *Solaris Source Compatibility Guide*—This guide describes how the SunOS/BSD Source Compatibility Package supports the SunOS 4.x commands, library routines, and files in the SunOS 5.4 environment.
- *Desktop Integration Guide*—This guide instructs ISVs who wish to integrate their applications with other applications and tools that run in the OpenWindows environment.

The following documents are provided in the *Solaris 2.4 Reference Manual AnswerBook* on-line documentation set:

- *SunOS Reference Manual*—This manual contains the SunOS 5.4 manual pages. These manual pages are organized by section as follows:
 - *man Pages(1): User Commands*
 - *man Pages(1M): System Administration Commands*
 - *man Pages(2): System Calls*
 - *man Pages(3): Library Routines*
 - *man Pages(4): File Formats*
 - *man Pages(5): Headers, Tables and Macros*
 - *man Pages(6): Demos*
 - *man Pages(7): Special Files*
 - *man Pages(9): DDI and DKI Overview*

- *man Pages(9E): DDI and DKI Driver Entry Points*
- *man Pages(9F): DDI and DKI Kernel Functions*
- *man Pages(9S): DDI and DKI Data Structures*

The following document is provided in the *Solaris 2.4 Supplemental Developer AnswerBook* on-line documentation set:

- *Application Packaging Developer's Guide*—This guide describes how to create an installable package for application software and gives examples of package creation. It also describes how to create a CD-ROM disc that contains a package.

Technical Support, Training, and Documentation

3 

This chapter tells you how to obtain technical assistance when you are installing or using the SDK. It also lists technical classes offered through Sun Educational Services and tells you how to obtain a class-location list. In addition, a complete list of the SDK documentation is provided for easy reference.

Technical Support

If you need help with the installation or use of the SDK and you're calling from the United States or Canada, call 1-800-SOFTSPT (1-800-763-8778) for an Authorized Service Provider near you. Customers outside the United States or Canada, please call 1-510-460-3267. Also refer to your Support Addendum card.

The SDK is also supported through the *SunSoft Catalyst Developer's Program* (CatalystSM). The Catalyst program offers a variety of technical support services to assist you in bringing your Solaris-ready software applications to market. For information about the Catalyst program, contact the Catalyst Information Center at 510-460-3267.

Sun Educational Services

In partnership with Sun Educational Services, SunSoft Authorized Education Centers provide the following training on Solaris Developer products. These course are offered at many locations in the United States and throughout the world. For a more current listing of classes and class-location list, call Sun Educational Services at 1-800-422-8020.

Lecture and Lab Courses

- Fundamentals of Solaris 2.x
- Solaris 2.X for Advanced Users
- Solaris for Programmers
- Solaris 2.X Concepts and Tuning
- Solaris 2.X Internals
- Programming Transition to Solaris
- Shell Programming
- C Shell Programming
- Korn Shell Programming
- Motif Programming
- ToolTalk Interapplication Programming
- Multithreaded Interapplication Programming
- Network Programming
- C++ Programming
- Object-Oriented Analysis and Design Using C++
- Introduction to Writing Device Drivers Solaris 2.X
- Grep, Sed, and Awk Programming
- System Interface Programming
- ANSI C for Nonprogrammers
- ANSI C Programming
- Advanced ANSI C Programming
- SPARCworks Programming Environment

Self-Paced Courses

- Fundamentals of Solaris
- UNIX Software Tools for Programmers
- UNIX System V Release 4 Internals
- Shell Command Language for Programmers
- ANSI C Language for Programmers
- Using C++

- Advanced C++
- Object-Oriented Design
- OSF/Motif Video Workshop Volume One: Introduction to OSF/Motif
- OSF/Motif Video Workshop Volume Two: Introduction to OSF/Motif
- Programming in C++
- X Window Video Workshop Volume One: An Introduction to the X Window System
- X Window Video Workshop Volume Two: Programming with Xlib

A Complete List of SDK Documentation

This section lists all of the SDK documentation and tells you where to find it. The SDK documentation is provided in the following ways:

- On-line through the AnswerBook product and in PostScript files.
- In hard copy in the SDK box.
- In hard copy through SunExpress.
- In hard copy from third-party publishers.

Documents Available Through the AnswerBook Product

The AnswerBook product provides a way for you to view the SDK documentation on-line. The SDK includes several sets of on-line documentation. These sets of documentation are called:

- *Solaris 2.4 Software Developer AnswerBook*
- *OpenWindows Developer Guide AnswerBook*
- *OPEN LOOK to Motif GUI Transition Guide*
- *Solaris XGL 3.1 AnswerBook*
- *Solaris XIL 1.2 AnswerBook*
- *Solaris 2.4 Supplemental Developer AnswerBook*
- *Solaris 2.4 Reference Manual AnswerBook*

For information on installing the SDK on-line documentation, see the *Software Developer Kit Installation Guide*.

Solaris 2.4 Software Developer AnswerBook

- *Software Developer Kit Introduction* (this manual)
- *Desktop Integration Guide*
- *Standards Conformance Reference Manual*

- *System Services Guide*
- *SPARC Assembly Language Reference Manual*
- *Programming Utilities Guide*
- *Linker and Libraries Guide*
- *Solaris Source Compatibility Guide*
- *Solaris Binary Compatibility Guide*
- *Developer's Guide to Internationalization*
- *Multithreaded Programming Guide*
- *Network Interfaces Programmer's Guide*
- *x86 Assembly Language Reference Manual*
- *ToolTalk User's Guide*
- *ToolTalk Reference Manual*
- *OpenWindows Server Programmer's Guide*
- *OpenWindows Reference Manual*
- *XView Developer's Notes*
- *OLIT QuickStart Programmer's Guide*
- *OLIT Reference Manual*

OpenWindows Developer Guide AnswerBook

- *OpenWindows Developer's Guide: OLIT Code Generator Programmer's Guide*
- *OpenWindows Developer's Guide: XView Code Generator Programmer's Guide*
- *OpenWindows Developer's Guide: Motif Conversion Utilities Guide*
- *OpenWindows Developer's Guide: User's Guide*

Solaris XGL 3.1 AnswerBook

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

Solaris XIL 1.2 AnswerBook

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

Solaris 2.4 Supplemental Developer AnswerBook

- *Application Packaging Developer's Guide*
- *STREAMS Programmer's Guide*

Solaris 2.4 Reference Manual AnswerBook

- *SunOS Reference Manual*
 - *man Pages(1): User Commands*
 - *man Pages(1M): System Administration Commands*
 - *man Pages(2): System Calls*
 - *man Pages(3): Library Routines*
 - *man Pages(4): File Formats*
 - *man Pages(5): Headers, Tables and Macros*
 - *man Pages(6): Demos*
 - *man Pages(7): Special Files*
 - *man Pages(9): DDI and DKI Overview*
 - *man Pages(9E): DDI and DKI Driver Entry Points*
 - *man Pages(9F): DDI and DKI Kernel Functions*
 - *man Pages(9S): DDI and DKI Data Structures*
- *OpenWindows Reference Manual*

Documents Available Through PostScript Files

These files are installed in `/opt/SUNWsdkwp/doc`.

Title	File Name
<i>ToolTalk White Paper</i>	<code>ToolTalk-Spec.ps</code>
<i>Introduction to the ToolTalk Service White Paper</i>	<code>ToolTalk-Intro.ps</code>
<i>ToolTalk Service White Paper</i>	<code>ToolTalk-TechReport.ps</code>
<i>A Simple Demonstration of How the ToolTalk Service Works</i>	<code>ToolTalk-demo_WP.ps</code>
<i>Designing and writing a ToolTalk Procedural Protocol</i>	<code>protocol_WP.doc</code>
<i>The ToolTalk Service and Project DOE Relationship</i>	<code>TT2DOE_wp.ps</code>
<i>The Messaging Object Service</i>	<code>mos_wp.ps</code>
<i>Object Services: Working in Concert</i>	<code>OSinConcert-wp.ps</code>
<i>Project DOE White Paper</i>	<code>ProjectDOE.ps</code>
<i>Object Storage in DOE White Paper</i>	<code>ObjStorage_wp.ps</code>

Title	File Name
<i>The ToolTalk Desktop Services Message Set White Paper</i>	MessageSet_Desktop_1.1.1.ps
<i>The ToolTalk Document and Media Exchange Message Set White Paper</i>	MessageSet_MediaEx_1.1.1.ps
<i>COSE White Paper</i>	cose.WP.ps, coseWPfig.ps
<i>Developing OpenStep Applications</i>	OpenStep_wp.ps
<i>Solaris Visual White Paper</i>	SVISUAL_<1...7>.ps
<i>The XGL White Paper</i>	XGL.ps
<i>The XIL White Paper</i>	XIL.ps
<i>PEX Implementation Specification</i>	PEX-ImplSpec.ps
<i>CGE PEX 5.1 Portability Guide</i>	PEX_portgd.ps
<i>OPEN LOOK to Motif Transition White Paper</i>	OLtoMotifTransition.ps
<i>The Display PostScript White Paper</i>	dpswhitepaper.ps
<i>Solaris Distributed Computed Foundation Federated Services</i>	fed_WP.ps

Documents Available in Hard Copy Only

- *Software Developer Kit Installation Guide*
- *Software Developer Kit Open Issues and Late-Breaking News*

On-Line Documents Also Available in Hard Copy

To get hardcopy versions of the SDK on-line documentation contact SunExpress™ (1-800-873-7869) or an authorized Sun reseller.

Third-Party Documents

- *PEXlib Programming Manual*, O’Rielly & Associates, Inc.
- *PEXlib Reference Manual*, O’Rielly & Associates, Inc.
- *OSF Application Environment Specification (AES) User Environment Volume*, Revision C, PTR Prentice Hall
- *OSF/Motif Programmer’s Guide*, Release 1.2, PTR Prentice Hall
- *OSF/Motif Programmer’s Reference*, Release 1.2, PTR Prentice Hall
- *OSF/Motif 1.2 Style Guide*, PTR Prentice Hall

