



Introduction to the Solaris Development Environment

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817-6295-10
January 2005

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, Java, J2SE, J2EE, JavaHelp, ONC+, Solstice Enterprise Agents, JavaBeans, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was 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.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, Java, J2SE, J2EE, JavaHelp, ONC+, Solstice Enterprise Agents, JavaBeans, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



041115@10082



Contents

Preface	5
1 Introduction to the Solaris Operating System	9
Information Sources for the Solaris Operating System	9
Map to the Solaris OS Man Page Collection	11
Building Applications	13
Packaging Applications for the Solaris OS	13
Using the Runtime Linker and Link Editor	13
Using the Sun Studio Tools	14
Kernel-Level Debugging	14
Solaris Dynamic Tracing	15
Writing Java Programs for the Solaris OS	15
Supplying Online Help for Java Applications	15
Programming to Specific Sun Hardware and Configurations	15
Known Solaris Compatible x86 Hardware	16
Testing x86 Hardware for Solaris Compatibility	16
Developing in the x86 Assembly Language	16
Developing in the SPARC Assembly Language	16
Developing 64-bit Applications for the Solaris OS	17
Writing for a Cluster Environment	17
Network Programming in the Solaris OS	17
Making Remote Procedure Calls	17
Writing a System Management Agent for the Solaris OS	18
Developing Web-Based Enterprise Management Applications	18
Programming with Solaris Interfaces and Frameworks	18
Multithreaded Programming in the Solaris OS	18

Programming Interfaces	19
Developing Custom Dynamic Host Configuration Protocol Applications	19
Developing Security Applications and Services in the Solaris OS	20
Developing Device Drivers	20
Writing System Resource Management Applications	20
Developing Software for International Audiences	21

Preface

Introduction to the Solaris Development Environment is a starting point for those developers who have not previously worked in the Solaris™ Operating System (OS). Seasoned Solaris developers who might not be aware of the various information resources can benefit from this guide as well. The purpose of the book is to help developers take advantage of the numerous interfaces, frameworks, and tools for the Solaris OS. The book provides an overview of the Solaris OS, abstracts of key manuals for Solaris developers, and links to sources of detailed information.

Who Should Use This Book

This book is suggested for any developer who needs to learn more about the Solaris development environment and might not be aware of all the available information resources.

How This Book Is Organized

This single-chapter manual consists of the following major sections:

- “Information Sources for the Solaris Operating System” on page 9
- “Map to the Solaris OS Man Page Collection” on page 11
- “Building Applications” on page 13
- “Programming to Specific Sun Hardware and Configurations” on page 15
- “Network Programming in the Solaris OS” on page 17
- “Programming with Solaris Interfaces and Frameworks” on page 18

Accessing Sun Documentation Online

The docs.sun.comSM Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is <http://docs.sun.com>.

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see “Buy printed documentation” at <http://docs.sun.com>.

Typographic Conventions

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

TABLE P-1 Typographic Conventions

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

TABLE P-1 Typographic Conventions (Continued)

Typeface or Symbol	Meaning	Example
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . Perform a <i>patch analysis</i> . Do <i>not</i> save the file. [Note that some emphasized items appear bold online.]

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-2 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	#

Introduction to the Solaris Operating System

Welcome to the Solaris 10 Operating System (OS). The Solaris OS provides developers with numerous interfaces, frameworks, and tools to take advantage of Solaris technologies. The purpose of this manual is to provide a roadmap for developers to learn about the Solaris OS. The following topics are covered:

- “Information Sources for the Solaris Operating System” on page 9
- “Map to the Solaris OS Man Page Collection” on page 11
- “Building Applications” on page 13
- “Programming to Specific Sun Hardware and Configurations” on page 15
- “Network Programming in the Solaris OS” on page 17
- “Programming with Solaris Interfaces and Frameworks” on page 18

Information Sources for the Solaris Operating System

This manual provides an overview of the Solaris 10 OS with links to sources of further information. Sun Microsystems, Inc. provides numerous vehicles for getting information about developing applications in the Solaris OS:

- **Man pages** – The reference manual (“man”) pages provide the syntax and descriptions for all commands, functions, structures, and file formats in the Solaris OS. If you know the name of the item you are seeking information about, type the **man** command followed by the item name at the command line. Man pages are also available as online documents from Solaris 10 reference manual collection (<http://docs.sun.com/db/coll/40.10>). To learn how the Solaris man pages are organized, see “Map to the Solaris OS Man Page Collection” on page 11.
- **OS Manuals** – Sun Microsystems, Inc. (SMI) provides a family of programming manuals describing APIs and frameworks in the Solaris OS. These manuals are mainly grouped in the Solaris developer collection

(<http://docs.sun.com/db/coll/45.20>).

- **Sun Developer Network articles** – Where the OS manuals provide the fundamentals about technologies, articles on the Sun Developer Network complement documentation by providing information on special techniques and aspects of technologies of a more specialized nature. The Sun Developer Network provides detailed technical articles as well as higher level white papers. The Sun Developer web site also provides information on new products such as linCAT, which points out potential compatibility problems in porting applications from linux to the Solaris OS.
- **Sun Studio documentation** – The Sun Studio software, a separate Sun product, provides a rich development environment for developers who are writing software for the Solaris OS. For complete documentation of the Sun Studio compilers and tools, see the <http://developers.sun.com/prodtech/cc/> portal. For product literature, see <http://www.sun.com/software/products/studio/index.html>. For a summary of the Sun Studio software, see “Using the Sun Studio Tools” on page 14.
- **SunSolve online support documents** – SunSolveSM maintains a knowledge base of short articles that provide solutions for specific issues concerning Sun hardware and software products. See the Support Documents and Knowledge Base web pages at the SunSolve web site. SunSolve also provides patch information, special bulletins, a FAQ for ForteTM for JavaTM, and articles on the StarOfficeTM software.
- **Sun instructor-led and online courses** – Courses on the Solaris OS as well as other technologies offered by Sun Microsystems, Inc., are listed in the course catalog (<http://training.sun.com/index.html>).
- **Third-party books** – You can also get good information on the Solaris OS from books published by Sun Microsystems Press (<http://www.sun.com/books>) and the Blueprints series (<http://www.sun.com/books/blueprints.series.html>). Two notable books for Solaris developers are *Solaris Internals - Core Kernel Architecture* by Jim Mauro and Richard McDougall, 1st edition, October 5, 2000, and *Solaris Systems Programming* by Rich Teer, Prentice Hall PTR, August 19, 2004.
- **Solaris OS product literature** – For product literature on the Solaris OS including data sheets, tours, case studies, and white papers, see the Solaris Operating System (<http://www.sun.com/solaris>) web site.
- **Other Sun documentation** – The following documentation sources can be useful to developers, although these manuals are less directly involved with Solaris OS development:
 - Solaris 10 system administrator collection (<http://docs.sun.com/db/coll/47.16>)
 - Solaris 10 on Sun hardware collection (<http://docs.sun.com/db/coll/1140.1>)
 - Solaris 10 international language support collection (<http://docs.sun.com/db/coll/767.3>)
 - J2SETM platform tutorials (<http://java.sun.com/docs/books/tutorial>)

- J2EE™ platform tutorials
(<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>)
 - *Sun Java System Application Server Developer Guide*
(<http://docs.sun.com/db/doc/817-6087>)
 - Java Web Services tutorial
(<http://java.sun.com/webservices/tutorial.html>)
 - LDAP SDK (<http://docs.sun.com/doc/817-6707>)
 - Identity Server Java APIs (<http://docs.sun.com/doc/817-5713>)
-

Map to the Solaris OS Man Page Collection

As in all UNIX® systems, the Solaris Operating System provides detailed information about system internals through man pages. If you are unfamiliar with the man page concept, see the man page for the `man(1)` command itself or type `man man` on the command line. The Solaris OS provides more than 13,000 man pages. All man page collections are broken up into sections, which vary slightly from vendor to vendor or even release to release. Large sections are sometimes divided into volumes.

Viewing man pages at docs.sun.com offers several advantages over viewing man pages from the command line. Each section or volume is presented as a book with an index and preface, and an alphabetized list of items with their short descriptions. For an example of a section with short descriptions, see the system calls summary (<http://docs.sun.com/db/doc/816-5167/6mbb2jaeu?a=expand>). In some sections, an introductory page provides an overview of the section. The man pages for the Solaris 10 OS are organized into sections and volumes as follows:

- *man pages section 1: User Commands* – Contains all non-administration commands on the system. The `Intro(1)` man page provides information on man page command syntax for users, guidelines for developers who are designing new man pages, and an explanation of the subdivisions within Section 1.
- *man pages section 1M: System Administration Commands* – Contains commands for system maintenance and other administrative purposes. The `Intro(1M)` man page repeats the man page command syntax for users from the User Commands volume.
- *man pages section 2: System Calls* section – Contains prototypes and descriptions for the various system calls that can be made in the Solaris Operating System. This section focuses on calls that directly invoke UNIX system primitives. The `Intro(2)` man page provides a list of all possible error returns from system calls as well as a glossary of system call terminology.
- Section 3 provides information on library calls and is divided into the following volumes:

- *man pages section 3: Library Interfaces and Headers* volume – Contains general libraries that do not fit into the other categories. The `Intro(3)` man page provides the complete organization of the Section 3 libraries, a brief explanation of multithreaded applications, and a few basic definitions.
- *man pages section 3: Basic Library Functions* volume – Contains the core C library functions that are basic to application development.
- *man pages section 3: Networking Library Functions* volume – Contains libraries that are related to networking, such as LDAP and RPC.
- *man pages section 3: Curses Library Functions* volume – Contains libraries for displaying characters and graphics.
- *man pages section 3: Realtime Library Functions* volume – Contains libraries for performing realtime activities.
- *man pages section 3: Extended Library Functions* volume – Contains specialized libraries.
- *man pages section 4: File Formats* section – Describes file formats. The C structure declarations for the file formats are provided where applicable.
- *man pages section 5: Standards, Environments, and Macros* section – Contains miscellaneous information. The `Intro(5)` man page describes the topics in this section.
- *man pages section 6: Demos* section – Contains games and demos.
- *man pages section 7: Device and Network Interfaces* section – Contains various device and network interfaces available on the system. The `Intro(7)` man page explains how the section is organized.
- Section 9 provides information on the device drivers and is organized into the following volumes:
 - *man pages section 9: DDI and DKI Driver Entry Points* volume – Contains information on the Device Driver Interface and the Driver-Kernel Interface (DDI/DKI), which are recommended for device drivers in the Solaris kernel. The `Intro(9E)` man page provides an overview of driver entry points.
 - *man pages section 9: DDI and DKI Kernel Functions* volume – Contains the kernel functions that are available for use by device drivers. The `Intro(9F)` man page presents summaries of the functions, organized into STREAMS and non-STREAMS functions.
 - *man pages section 9: DDI and DKI Properties and Data Structures* volume – Contains the data structures that are used by drivers to share information between the driver and the kernel. The `Intro(9S)` man page presents summaries of the structures, organized into STREAMS and non-STREAMS structures.

Building Applications

This section provides information on Solaris facilities for packaging, compiling, debugging, and tuning applications.

Packaging Applications for the Solaris OS

All software programs must be incorporated into a package to be installed in the Solaris OS. The *Application Packaging Developer's Guide* provides step-by-step instructions and relevant background information for designing, building, and verifying packages. A chapter with case studies provides several package creation examples in a variety of situations. This document also includes descriptions of advanced techniques that you might find helpful during the package creation process.

- Packages that are not part of the Solaris Operating System (unbundled packages) can use the class mechanism to customize server/client installations.
- Relocatable packages can be designed to accommodate the desires of the administrator.
- A complex product can be delivered as a set of composite packages that automatically resolve package dependencies.
- A package designer can customize the upgrading and patching process.
- Patched packages can be delivered in the same way as unpatched packages. Archives for backing out the software can also be included in the product.

Using the Runtime Linker and Link Editor

The Solaris OS provides a link editor and runtime linker. The *Linker and Libraries Guide* covers the link editor `ld(1)`, the runtime linker `ld.so.1(1)`, the ELF object file format, and shared objects, which are sometimes referred to as shared libraries.

The manual is intended for a range of programmers who are interested in the Solaris linkers, from the curious beginner to the advanced user. Beginners learn the principal operations of the link editor and runtime linker. Intermediate programmers learn to create and use efficient custom libraries. Advanced programmers, such as language-tools developers, learn how to interpret and generate object files. A chapter on application binary interfaces describes how to manage the evolution of an interface that is provided by a dynamic object. Other chapters cover thread-local storage and `mapfile` directives.

Using the Sun Studio Tools

The Sun™ Studio software, a separate Sun product, provides modules for creating, editing, building, debugging, and analyzing the performance of a C, C++, or Fortran application. JNI (Java Native Interface) development is also supported. Many Sun Studio tools have both a GUI and command-line equivalent. Those tools with GUIs provide online help. For the command-line versions, use the associated man pages. See the Sun Studio portal (<http://developers.sun.com/prodtech/cc/reference/docs/index.html>) for documentation of the tools and other useful information.

The Sun Studio software includes the following tools:

- **Sun Studio IDE** – An integrated development environment that provides access to the Sun Studio tools. In addition to C, C++, and Fortran tools, the IDE includes a set of basic Java language support modules that can be enabled for JNI development.
- **Sun Studio C compiler** – Includes a C compiler, incremental link editor, and lint program.
- **Sun Studio C++ compiler** – Includes a full-featured C++ compiler and interval arithmetic library.
- **Sun Studio Fortran compiler** – Includes full-featured environment and libraries for both f95 and f77.
- **dbx Debugger** – An interactive, source-level, command-line debugging tool.
- **Sun Studio `dmake` `make` tool** – A command-line tool for building targets in distributed, parallel, or serial mode.
- **Math libraries** – A floating-point environment that is supported by software and hardware on SPARC® and x86 platforms that run the Solaris Operating System.
- **OpenMP API** – Directives and pragmas for converting legacy Sun parallelization directives to OpenMP, which is a portable, parallel programming model for shared memory multiprocessor architectures.
- **Performance Analyzer** – A GUI and command-line tool for collecting and analyzing performance data.
- **Sun Performance Library** – A library of Sun-specific extensions and features for using optimized, high-speed mathematical subroutines for solving linear algebra and other numerically intensive problems.

Kernel-Level Debugging

The Modular Debugger `mdb` is an extensible, general purpose debugging tool for the Solaris OS. The *Solaris Modular Debugger Guide* describes how to use the `mdb(1)` command to debug complex software systems. The guide emphasizes the facilities that are available for debugging the Solaris kernel and associated device drivers and modules. The guide includes a complete reference for the `mdb` language syntax, debugger features, and the `mdb` module programming API.

The *Solaris Modular Debugger Guide* also features a information on kmdb, the kernel-level analogue to mdb.

Solaris Dynamic Tracing

DTrace is a comprehensive dynamic tracing framework for the Solaris Operating System. DTrace provides a powerful infrastructure to enable administrators, developers, and service personnel to concisely answer arbitrary questions about the behavior of the operating system and user programs. The *Solaris Dynamic Tracing Guide* describes how to use DTrace to observe, debug, and tune system behavior. This book also includes a complete reference for bundled DTrace observability tools and the D programming language.

Writing Java Programs for the Solaris OS

The *JDK for Solaris Developer's Guide* is intended for application developers who use the Java 2 Platform Standard Edition 5, on the Solaris Operating System. The Java software is optimized to deliver superior performance to server-side and client-side Java technology applications in an enterprise environment.

Supplying Online Help for Java Applications

The *JavaHelp System User's Guide* (<http://docs.sun.com/db/doc/819-0913>) provides information for developers providing online help for Java applications. The JavaHelp™ system is a URL-based facility that points to topics contained in files. The help files are displayed in the help viewer one file at a time. The manual covers both authoring and packaging.

Programming to Specific Sun Hardware and Configurations

This section provides information needed for development on specific hardware platforms and architectures, including clustered environments.

Known Solaris Compatible x86 Hardware

The Hardware Compatibility List (HCL) (<http://www.sun.com/bigadmin/hcl>) identifies x86 hardware that is compatible with the Solaris OS. Separate lists identify systems and controllers that are certified for Solaris 8, Solaris 9, and Solaris 10 versions of the operating system. The HCL is available in a comma-separated value (CSV) format for advanced users who want to search or sort the list using a spreadsheet or database.

Testing x86 Hardware for Solaris Compatibility

The Hardware Certification Test Suite (HCTS) (<http://www.sun.com/bigadmin/hcl/hcts>) includes system certification tests, controller certification tests, utilities, and a browser-based user interface. HCTS tests certify systems, motherboards, and various network, storage, and serial I/O controllers to work with the Solaris Operating System. HCTS collects log files, determines test pass or fail results, and generates a final test report. Hardware that passes either Level 1 or Level 2 Certification is eligible to be included in the Hardware Compatibility List as Test Suite Certified.

Developing in the x86 Assembly Language

The Solaris OS provides an assembly language for the x86 platform. The *x86 Assembly Language Reference Manual* describes the assembler that generates code for the 32-bit x86 processor architecture and translates source files that are in assembly language format into object files in linking format. This book describes the syntax of the Solaris x86 assembly language and maps the Solaris x86 assembly language instruction mnemonics to the native x86 instruction set.

Developing in the SPARC Assembly Language

The Solaris OS provides an assembly language for the SPARC platform. The *SPARC Assembly Language Reference Manual* (<http://docs.sun.com/db/doc/816-1681>) describes the assembler that runs on the SPARC architecture and translates source files that are in assembly language format into object files in linking format. The description includes the assembler syntax, the executable and linking format, and the relationship between hardware instructions of the SPARC architecture and the assembly language instruction set. The book contains a description of the SPARC-V9 instruction set. Appendixes list the pseudo-operations that the SPARC assembler supports, with examples of their use. Other appendixes describe the available assembler command-line options and an example of the correspondence between the assembly code and the C code.

Developing 64-bit Applications for the Solaris OS

The Solaris OS provides a 64-bit computing environment along with backward compatibility for 32-bit applications. The *Solaris 64-bit Developer's Guide* is written primarily for the application developer, and provides guidance on choosing whether to use the 32-bit or 64-bit Solaris application programming environment. The manual explains the similarities and differences between the 32-bit and 64-bit application environments and explains how to write code that is portable between the two environments. This book also describes some of the tools provided by the operating system for developing 64-bit applications.

Writing for a Cluster Environment

The Solaris OS can be configured into a cluster environment to improve availability. The *Sun Cluster Data Services Developer's Guide for Solaris OS* (<http://docs.sun.com/db/doc/817-6555>) describes the Sun Cluster data service API. Developers can use this API to turn a standard user application such as a web browser or database into a highly available data service that can run in the Sun Cluster environment.

Network Programming in the Solaris OS

This section provides information on making remote procedure calls, programming a system management agent, and developing web-based enterprise management applications.

Making Remote Procedure Calls

The Solaris OS provides ONC+™ distributed services for making remote procedure calls (RPC). The *ONC+ Developer's Guide* describes the ONC+ distributed services that were developed at Sun Microsystems. ONC+ technologies consist of a family of technologies, services, and tools.

- Remote procedure calls (RPC).
- Transport-independent remote procedure call (TI-RPC) was developed to make RPC applications transport independent.
- External data representation (XDR) is an architecture-independent specification for representing data.
- Network Information Services Plus (NIS+) is the enterprise naming service in the Solaris environment. This service provides a scalable and secure information base.

The book contains information on the `rpcgen` tool and the `portmap` utility. The book contains code examples on the use of RPC.

Writing a System Management Agent for the Solaris OS

The System Management Agent (SMA) is based on the open source Net-SNMP agent. The *Solaris System Management Agent Developer's Guide* provides information for developers who want to create Management Information Base (MIB) modules to extend the functionality of the agent. The manual also provides information on the creation of SMA modules and the interfaces that MIB modules use. An appendix provides information for developers who want to migrate an SEA subagent from the Solstice Enterprise Agents™ software to use in the System Management Agent.

Developing Web-Based Enterprise Management Applications

The *Solaris WBEM Developer's Guide* describes the components of the Solaris Web-Based Enterprise Management (WBEM) Software Development Kit (SDK) and explains how to use these components to develop WBEM-based applications. The book describes the following subjects:

- The Common Information Model (CIM) Object Manager
- The WBEM Query Language (WQL)
- Creating JavaBeans™ components using the Managed Object Format (MOF) compiler
- WBEM security mechanisms

Programming with Solaris Interfaces and Frameworks

This section provides information on interfaces and frameworks that are unique to the Solaris OS.

Multithreaded Programming in the Solaris OS

The Solaris OS provides the ability to develop applications with separate, parallel threads to improve application performance. The *Multithreaded Programming Guide* covers the POSIX and Solaris threads APIs, programming with synchronization

objects, and compiling multithreaded programs This guide is for developers who want to use multithreading to separate a process into independent execution threads, improving application performance and structure. Appendixes contain examples of code for both POSIX and Solaris threads.

Programming Interfaces

The *Programming Interfaces Guide* describes programming interfaces that are specific to the Solaris environment. This book has information on the following subjects:

- Memory and CPU management
- Process scheduling
- Input and output interfaces
- Interprocess communication and sockets
- Transport Layer Interface (TLI) and the X/Open Transport Interface (XTI)
- Transport selection
- The Solaris application binary interface

The Solaris OS includes two utilities that enable application developers to verify an application's compliance with the Solaris Application Binary Interface (ABI). Compliance with the Solaris ABI ensures that your software code will be portable across releases of the Solaris OS. The Solaris ABI defines the interfaces that are available for the use of application developers.

The `appcert` utility statically examines the Solaris library interfaces used by ELF binaries for instances of private interface usage. The `apptrace` tool uses the link-auditing capability of the run-time linker to dynamically trace Solaris library routine calls as the application runs.

Developing Custom Dynamic Host Configuration Protocol Applications

The Solaris OS provides dynamic host configuration protocol (DHCP) services as well as a framework for developing custom DHCP applications. The *Solaris DHCP Service Developer's Guide* provides information for developers who want to use a data storage facility that is not currently supported by the Solaris DHCP service. The manual gives an overview of the data access framework used by Solaris DHCP and general guidelines for developers. The book describes the service provider layer API and provides sample code templates.

Developing Security Applications and Services in the Solaris OS

The *Solaris Security for Developers Guide* is for developers of applications that consume security services as well as developers of applications that provide security services for the Solaris OS. Programming interfaces are documented for the following services: PAM, SASL, GSS-API, the Solaris cryptographic framework, and process privileges. The book provides examples of use for the Generic Security Standard API and the Simple Authentication Security Layer.

Developing Device Drivers

The Solaris OS provides a family of interfaces for facilitating the development of robust, portable device drivers. The *Writing Device Drivers* manual provides information on developing drivers for character-oriented devices, block-oriented devices, USB devices, and SCSI target and HBA devices for the Solaris OS. This book discusses how to develop multithreaded reentrant device drivers for all architectures that conform to the Solaris DDI/DKI.

Additional topics include:

- Porting Solaris drivers to a 64-bit environment
- Cluster-aware drivers
- Driver autoconfiguration
- Programmed I/O
- Direct Memory Access (DMA)
- Power management
- Device context management
- Compilation, installation, and testing of drivers
- Debugging drivers

Appendixes provide information on hardware issues, kernel functions available to device drivers, and guidelines on device drivers for 64-bit environments.

A separate book, the *Device Driver Tutorial*, provides an overview to device driver development, several basic driver examples, and a list of tips for developing drivers in the Solaris OS.

Writing System Resource Management Applications

The *Solaris 10 Resource Manager Developer's Guide* describes how to write applications that partition and manage system resources such as processor sets and scheduling class. This book references the programming APIs provided to partition, schedule, and

set bounds on the consumption of system resources and to make the configuration of resources persistent. This book provides programming examples and a discussion of programming issues to consider when writing an application.

Developing Software for International Audiences

The Solaris OS provides an internationalization architecture to assist in the development, the deployment, and the management of applications and language services from around the world. A single multilingual product provides support for 39 different languages and 162 locales. In addition, support is available for the complex text layout that is required for Thai and Hindi scripts. Bidirectional text capability is also supported for languages such as Arabic and Hebrew. The *International Language Environments Guide* describes how to use the current Solaris release to build global software products that support a variety of languages and cultural conventions.

