Introduction to the Oracle® Solaris Developer Documentation

Contents

Preface

Introduction to the Oracle Solaris Developer Documentation is a starting point for those developers who have not previously worked in the Oracle Solaris Operating System (Oracle Solaris OS). Seasoned Oracle 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 Oracle Solaris OS. The book provides an overview of the Oracle Solaris OS development environment, abstracts of key manuals for Oracle Solaris developers, and links to sources of detailed information.

Note – This Solaris release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC, SPARC64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at `http://www.sun.com/bigadmin/hcl`. This document cites any implementation differences between the platform types.

In this document these x86 related terms mean the following:

- "x86" refers to the larger family of 64-bit and 32-bit x86 compatible products.
- "x64" points out specific 64-bit information about AMD64 or EM64T systems.
- "32-bit x86" points out specific 32-bit information about x86 based systems.

For supported systems, see the *Solaris 10 Hardware Compatibility List*.

Who Should Use This Book

This book is suggested for any developer who needs to learn more about the Oracle Solaris development environment. The book provides links for many information resources for learning about and developing on the Oracle Solaris OS.

How This Book Is Organized

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

Documentation, Support, and Training

See the following web sites for additional resources:

- Documentation (http://docs.sun.com)
- Support (http://www.oracle.com/us/support/systems/index.html)
- Training (http://education.oracle.com) – Click the Sun link in the left navigation bar.

Oracle Welcomes Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of its documentation. If you find any errors or have any other suggestions for improvement, go to http://docs.sun.com and click Feedback. Indicate the title and part number of the documentation along with the chapter, section, and page number, if available. Please let us know if you want a reply.

Oracle Technology Network (http://www.oracle.com/technetwork/index.html) offers a range of resources related to Oracle software:

- Discuss technical problems and solutions on the Discussion Forums (http://forums.oracle.com).
- Get hands-on step-by-step tutorials with Oracle By Example (http://www.oracle.com/technology/obe/start/index.html).
- Download Sample Code (http://www.oracle.com/technology/sample_code/index.html).

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file.
		Use `ls -a` to list all files.
		`machine_name% you have mail.`
AaBbCc123	What you type, contrasted with onscreen computer output	`machine_name% `**`su`**
		`Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*.
		A *cache* is a copy that is stored locally.
		Do *not* save the file.
		Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

1

Introduction to the Oracle Solaris Developer Documentation

The Oracle Solaris Operating System (Oracle Solaris OS) provides developers with numerous interfaces, frameworks, and tools to take advantage of Oracle Solaris technologies. This manual provides a roadmap for developers to learn about the Oracle Solaris OS and provides links to sources of further information. The following topics are covered:

Which Oracle Solaris Distribution is Right for You?

The Oracle Solaris OS distribution you choose depends on how you want to use the system.

Oracle Solaris 10 Operating System

Use the Oracle Solaris 10 OS for mission critical activities and production, such as in a large enterprise.

For information about the Oracle Solaris 10 OS, see the Oracle Solaris product site.

To get the Oracle Solaris 10 OS, see Get It: Oracle Solaris 10. You can download the Oracle Solaris 10 OS, you can download the Oracle Solaris 10 appliance image for VirtualBox, or you can order a DVD media kit. You can also get information about Oracle Premier Support for Operating Systems.

OpenSolaris Operating System

Download the OpenSolaris OS in a slim install configuration that fits on a single CD. Install the base OS from the CD, and then install additional software packages of your choice from the OpenSolaris package repository. Alternatively, use Automated Installer to install the OpenSolaris OS and your complete working environment on multiple machines.

For more information about the OpenSolaris OS, or to get the OpenSolaris OS, see the OpenSolaris product site. You can download the LiveCD image, you can download the Automated Installer bootable image, or you can order media. You can also download VirtualBox.

For information about Automated Installer, see *OpenSolaris 2010.05 Automated Installer Guide* and Migrating From Solaris Custom JumpStart to OpenSolaris Automated Installer.

Learning About Development on the Oracle Solaris OS

The various vehicles for getting information about developing applications in the Oracle Solaris OS are:

- **Man pages** – The reference manual ("man") pages provide syntax and descriptions for commands, functions, structures, and file formats in the Oracle 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 the Solaris 10 Reference Manual Collection. To learn how the Oracle Solaris man pages are organized, see "Map to the Oracle Solaris OS Man Page Collection" on page 12.

- **OS Manuals** – A family of programming manuals describes APIs and frameworks in the Oracle Solaris OS. These manuals, which provide the fundamentals about Oracle Solaris technologies, are primarily grouped in the Solaris 10 Software Developer Collection.

- **Developer tools documentation** – Two integrated development environments (IDE) are available for the Oracle Solaris OS: the Oracle Solaris Studio software and the NetBeans IDE.

  The Oracle Solaris Studio software provides a rich development environment for developers who are writing software for the Oracle Solaris OS. Included are C, C++, and Fortran compilers, the dbx debugger, and performance analysis tools. See "Using the Oracle Solaris Studio Tools" on page 15 for a summary of the Oracle Solaris Studio software.

  The NetBeans IDE provides developers with tools for creating professional cross-platform desktop, enterprise, web, and mobile applications. For documentation, see the Documentation, Training & Support area on the NetBeans.org site. For a summary explanation of the NetBeans software, see "Using the NetBeans IDE" on page 17.

  An additional set of tools can be used with the Oracle Solaris Studio tools to develop parallel applications designed to run on distributed-memory systems. See "Using Oracle Message Passing Toolkit" on page 18, formerly Sun HPC ClusterTools, for more information.

- **Sun Developer Network web site** – Highlights of the Sun Developer Network (SDN) Oracle Solaris and OpenSolaris Center include the following:
  - Learning Solaris OS includes tutorials, training, and demos
  - Library includes technical articles, whitepapers, and books
  - Tools includes building, performance, and web stack tools
  - A center for developers who are new to the Solaris platform
  - Opportunity to join the Sun Developer Network community
- **Mediacast** – The Mediacast Sun media sharing web site contains video and slide presentations on many topics. Click a tag or use the Search field to search for Solaris, OpenSolaris, DTrace, ZFS, and other topics.
- **SunSolve online support documents** – SunSolve 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 and special bulletins.
- **Oracle University** – Instructor-led courses, online courses, white papers, and other information on the Oracle Solaris OS and other technologies are among the resources available at Oracle University. For example, see Solaris under Product Courses > Sun.
- **Retail books** – You can also learn about the Oracle Solaris OS from books published by commercial publishers.

  Notable books for Oracle Solaris developers are:
  - *OpenSolaris Bible* by Nicholas A. Solter, Jerry Jelinek, and David Miner; Wiley; February 12, 2009.
  - *Pro OpenSolaris* by Harry Foxwell and Christine Tran; Apress; April 27, 2009.
  - *Solaris 10 System Administration Essentials* by Solaris Engineers; Prentice Hall PTR; November 30, 2009.
  - *Solaris 10 Security Essentials* by Sun Microsystems Security Engineers; Prentice Hall PTR; November 23, 2009.
  - *Solaris 10 ZFS Essentials* by Scott Watanabe; Prentice Hall PTR; January 4, 2010.
  - *Solaris ZFS Administration Guide*; Sun Microsystems; Vervante; 2009.
  - *Solaris Application Programming* by Darryl Gove, Prentice Hall PTR, January 6, 2008.
  - *Solaris Performance and Tools: DTrace and MDB Techniques for Solaris 10 and OpenSolaris* by Richard McDougall, Jim Mauro, and Brendan Gregg, Prentice Hall PTR, July 30, 2006.
  - *Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture* by Jim Mauro and Richard McDougall, Prentice Hall PTR, July 20, 2006.
  - *Solaris Systems Programming* by Rich Teer, Prentice Hall PTR, August 19, 2004.
  - *The Java Tutorial, Fourth Edition* by Isaac Rabinovitch, Sharon Zakhour, Scott Hommel, Jacob Royal and Thomas Risser, September 29, 2006.

- **Oracle Solaris 10 OS product literature** – For product literature on the Oracle Solaris 10 OS including data sheets, tours, case studies, and white papers, see the Solaris Operating System web site.

- **Other Sun documentation** – The following documentation sources can be useful to developers, although these manuals are less directly involved with Oracle Solaris OS development:
  - Solaris 10 System Administrator Collection
  - Solaris 10 On Sun Hardware Documentation
  - Solaris 10 International Language Support Collection
  - Java Tutorials
  - Java EE 5 Tutorial

Map to the Oracle Solaris OS Man Page Collection

The Oracle Solaris OS 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 Oracle Solaris OS provides more than 13,000 man pages. Man page collections are separated into sections, which vary slightly from vendor to vendor or even from one release to another release. Large sections are sometimes divided into volumes. References to commands often include the reference manual section or volume number next to the command. For example the man command might be referred to as man(1).

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. In some sections, an introductory page provides an overview of the section.

The man pages for the Oracle Solaris OS are organized into sections and volumes as follows:

- *man pages section 1: User Commands* – Describes the 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* – Describes commands for system maintenance and other administrative purposes. The Intro(1M) man page repeats the man page command syntax explanation from the User Commands volume.

- *man pages section 2: System Calls* section – Describes prototypes and descriptions for the various system calls that can be made in the Oracle Solaris OS. This section focuses on calls that directly invoke UNIX system primitives. The Intro(2) man page provides a list of 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 – Describes 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 – Describes the core C library functions that are basic to application development.

- *man pages section 3: Networking Library Functions* volume – Describes libraries that are related to networking, such as LDAP and RPC.

- *man pages section 3: Curses Library Functions* volume – Describes libraries for displaying characters and graphics.

- *man pages section 3: Realtime Library Functions* volume – Describes libraries for performing realtime activities.

- *man pages section 3: Extended Library Functions* volume – Describes specialized libraries.

- *man pages section 3: Multimedia Library Functions* volume – Describes the libmlib library that provides functions for multimedia processing.

- *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 – Describes 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 – Describes device interfaces and network interfaces that are 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 – Describes the Device Driver Interface and the Driver-Kernel Interface (DDI/DKI), which are required for device drivers in the Oracle Solaris kernel. The Intro(9E) man page provides an overview of driver entry points.

  - *man pages section 9: DDI and DKI Kernel Functions* volume – Describes the kernel functions that are available for use by device drivers. The Intro(9F) man page presents summaries of the functions, which are organized into STREAMS and non-STREAMS functions.

  - *man pages section 9: DDI and DKI Properties and Data Structures* volume – Describes 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 in the Oracle Solaris OS

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

- "Oracle Solaris Developer Tools" on page 14
- "Using the Oracle Solaris Studio Tools" on page 15
- "Using the NetBeans IDE" on page 17
- "AMP Stack for Web Application Development and Deployment" on page 18
- "Using Oracle Message Passing Toolkit" on page 18
- "Packaging Applications for the Oracle Solaris OS" on page 19
- "Oracle Solaris Dynamic Tracing" on page 20
- "Writing Java Programs for the Oracle Solaris OS" on page 20
- "Supplying Platform-Independent Online Help" on page 21

Oracle Solaris Developer Tools

The Oracle Solaris OS includes a number of developer software packages, which are available when you install the appropriate Oracle Solaris software group for developers. For installation purposes, the Oracle Solaris OS is logically divided into software groups, which are collections of Oracle Solaris packages.

When you install the Oracle Solaris OS, you must select one of the following software groups, which contain the developer packages:

- Developer
- Entire
- Entire Plus OEM

With these software groups, you get compilers such as the GNU C compiler (`gcc`) and the Java compiler (`javac`). Also installed are the GNU source-level debugger (`gdb`) and the modular debugger (`mdb` and `kmdb`), a linker (`ld`), source control utilities (`sccs`), and build utilities such as `make`. See the article Building Software on the Solaris OS for a quick lesson in using some of the standard tools.

Note – Most developers should use the more powerful Oracle Solaris Studio and NetBeans development and build tools. See "Using the Oracle Solaris Studio Tools" on page 15 and "Using the NetBeans IDE" on page 17.

Using the Runtime Linker and Link Editor

The Oracle 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 Oracle Solaris linkers, from the 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.

Kernel-Level Debugging

The Modular Debugger mdb is an extensible, general purpose debugging tool for the Oracle Solaris OS. The *Oracle 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 Oracle 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 *Oracle Solaris Modular Debugger Guide* also features information on kmdb, the kernel-level analogue to mdb.

Using the Oracle Solaris Studio Tools

The Oracle Solaris Studio software provides modules for creating, editing, building, debugging, and analyzing the performance of a C, C++, or Fortran application. Many Oracle Solaris 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. If you start dbx from the command line, type commands at the (dbx) prompt to get a brief description of each dbx command.

Download Oracle Solaris Studio software from Sun Studio Downloads.

The Oracle Solaris Studio Express program offers early access releases of the next Oracle Solaris Studio release in development. Download Oracle Solaris Studio Express from Oracle Solaris Studio Express Downloads.

Note that the Oracle Solaris Studio IDE installs its own version of the NetBeans IDE. This NetBeans installation is not intended to be used independently of the Oracle Solaris Studio software, and you might experience errors if you use it separately. If you want to use the NetBeans IDE, you should install the NetBeans IDE separately from installing Oracle Solaris Studio or Oracle Solaris Studio Express software.

The Oracle Solaris Studio software includes the following tools:

- **Oracle Solaris Studio IDE** – An integrated development environment that provides access to the Oracle Solaris Studio C, C++, and Fortran tools.

    A NetBeans plugin enables you to use the Oracle Solaris Dynamic Tracing facility (DTrace) from the Oracle Solaris Studio IDE. DTrace enables you to explore the inner workings of the software programs running on your system. The DTrace GUI plugin enables you to use DTrace from the IDE by running D scripts in a window. The plugin includes several D scripts that can be easily extended and customized to suit your needs. See "Oracle Solaris Dynamic Tracing" on page 20 in this manual for more information about DTrace. The Oracle Solaris Studio 12 Update 1 IDE includes the NetBeans DTrace GUI plugin. For Oracle Solaris Studio 12.2, the DTrace GUI plugin is separate. See NetBeans DTrace GUI FAQ and documentation.

    The Oracle Solaris Studio 12 Update 1 IDE also includes the DLight tool, which offers a variety of instrumentation that takes advantage of the Oracle Solaris Dynamic Tracing (DTrace) debugging and performance analysis functionality.

- **Oracle Solaris Studio C compiler** – Includes a C compiler, incremental link editor, and lint program.

- **Oracle Solaris Studio C++ compiler** – Includes a full-featured C++ compiler and interval arithmetic library.

- **Oracle Solaris Studio Fortran compiler** – Includes a full-featured environment and libraries for both `f95` and `f77`.

- **dbx Debugger** – An interactive, source-level, command-line debugging tool.

- **Sun Memory Error Discovery Tool** (Discover) – An advanced development tool for detecting memory access errors.

- **Uncover** – A simple and easy to use command-line tool for measuring code coverage of applications.

- **DLight** – A separate tool for Oracle Solaris Studio 12.2, different from the DLight tool that is integrated into the Oracle Solaris Studio 12 Update 1 IDE. The Oracle Solaris Studio 12.2 DLight tool includes three profiling tools for processes in the AMP stack, in addition to profiling tools for C, C++, and Fortran programs.

- **Oracle Solaris 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 Oracle Solaris OS.

- **OpenMP** – A portable, pragma-based parallel programming model for shared memory multiprocessor architectures, is natively accepted and compiled by all three Oracle Solaris Studio compilers.

- **Performance Analyzer** – A GUI and command-line tool for collecting and analyzing performance data.

- **Thread Analyzer** – A GUI and command-line tool for analyzing the execution of multithreaded programs and checking for a variety of multithreaded programming errors.
- **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.

See the following web sites for Oracle Solaris Studio documentation:

- Sun Developer Network
- Sun Studio Compilers and Tools on wikis.sun.com

Using the NetBeans IDE

NetBeans IDE (Integrated Development Environment) provides tools to help you build cross-platform applications for the Oracle Solaris OS and other operating platforms. The NetBeans IDE is available from netbeans.org. To see the installation guide, go to the Releases & Planning page, click the "General Info" link for the release you want, and then find the *Installation Instructions* on that page.

The NetBeans IDE contains the following features:

- Java Desktop Applications – Create professional desktop applications using the NetBeans Java GUI Builder with Swing Application Framework and Beans Binding support. Build rich internet applications with JavaFX.
- PHP Development – A fast and lightweight PHP IDE with code completion and quick fixes, integrated FTP and Xdebug, and support for popular Web Services.
- Java EE and Web Applications – Build web applications using Ajax, JavaScript, and CSS. Support for frameworks including JSF, Struts, Spring and Hibernate. Full set of tools for EJB development.
- Ruby and Ruby on Rails Development – Powerful Ruby editor with code completion and debugger, and full support for Ruby on Rails. Includes the JRuby runtime.
- Visual Mobile Development – Create, test and debug GUI applications that run on mobile phones, set-top boxes, and PDAs.
- C and C++ Development – Full-featured C/C++ editor, debugger, project templates, support for multiple project configurations, remote development, performance profiling, and packaging of completed projects.

Visit the netbeans.org web site for more information. New users might find the following pages particularly useful:

- Guided Video Tour of the NetBeans IDE
- NetBeans IDE Java Quick Start Tutorial
- NetBeans documentation

- NetBeans Plugin Portal for plugin modules
- NetBeans Wiki with *New and Noteworthy*, community documentation, and answers to Frequently Asked Questions

AMP Stack for Web Application Development and Deployment

Web Stack software is a collection of popular open source web technologies that enable developers to easily build and deploy web applications. Web Stack includes an AMP (Apache/MySQL/Perl or PHP) stack that has been optimized for the Oracle Solaris OS. See Sun GlassFish Web Stack Documentation for more information.

The Web Stack components include the following technologies:

- Apache – An HTTP web server, which serves web pages to web clients such as browsers.
- MySQL – A relational database engine (DBMS) that can be used to store most dynamic data for web applications.
- PHP – A scripting language, which is used to embed instructions within HTML pages to provide dynamic content. A NetBeans plugin for PHP can also be installed in the NetBeans IDE from the Tools > Plugins > Available Plugins menu.
- PostgreSQL – An object-relational database management system (ORDBMS). PostgreSQL uses a client/server model.
- Ruby on Rails – A framework for web application development. Ruby is the object-oriented programming language. Rails is an open source Ruby framework for developing web-based, database-driven applications. In Web Stack, Ruby includes extensions for OpenSSL, curses, Tcl/Tk, and readline. The NetBeans IDE includes support for Ruby and JRuby.
- Squid – A fully-featured HTTP/1.0 proxy server. Squid offers a rich access control, authorization and logging environment to develop web proxy and content serving applications.
- memcached – memcached is a high-performance, distributed memory object caching system, which speeds up dynamic web applications by alleviating database load.
- Tomcat Servlet container server software, which processes Java Servlet and JavaServer Pages (JSP) applications to provide dynamic web pages.

Using Oracle Message Passing Toolkit

The Oracle Message Passing Toolkit, formerly Sun HPC ClusterTools, is a set of development tools that you can use to develop parallel applications designed to run on distributed-memory systems.

For Oracle Solaris 10, the latest Oracle Message Passing Toolkit can be downloaded from the Oracle Message Passing Toolkit page.

Oracle Message Passing Toolkit includes the following technologies:

- Open Message Passing Interface (Open MPI) – an open source implementation of the Message Passing Interface (MPI) standard. Open MPI is fully compliant with the MPI 2 standard including complete MPI I/O implementation and one-sided communications between MPI processes.

- Open Run-Time Environment (ORTE) – provides a basic set of parallel job management facilities, and includes plugin modules to support the following technologies:

  - The Oracle Grid Engine, which allows parallel jobs to be launched and maximizes the utilization of shared resources.

  - The Portable Batch System (PBS), a job scheduler that allocates network resources to batch jobs on networked, multi-platform environments.

The Oracle Message Passing Toolkit can be used with the Oracle Solaris Studio compilers for C, C++, and Fortran. The Oracle Message Passing Toolkit is integrated with Oracle Solaris Studio Performance Analyzer to deliver full profiling capabilities for MPI applications, including MPI States, a feature unique to Performance Analyzer. The Oracle Message Passing Toolkit also supports the popular Totalview and Allinea DDT parallel debuggers.

See the Sun HPC ClusterTools documentation set for complete information about using the Oracle Message Passing Toolkit.

Packaging Applications for the Oracle Solaris OS

Software programs must be incorporated into a package to be installed in the Oracle Solaris OS. The *Application Packaging Developer's Guide* provides step-by-step instructions and relevant background information for designing, building, and verifying System V, or SVR4 packages on the Oracle Solaris OS. 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 to be helpful during the package creation process.

- Unbundled packages, which are not part of the Oracle Solaris OS, 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.

For OpenSolaris releases, applications can be packaged for the Image Packaging System (IPS). See the *OpenSolaris 2010.05 Image Packaging System Guide* and Multi-platform Packaging for Layered Distros for more information about application packaging for IPS.

Oracle Solaris Dynamic Tracing

Oracle Solaris Dynamic Tracing (DTrace) is a comprehensive dynamic tracing framework for the Oracle Solaris OS. The DTrace facility 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. DTrace can help developers identify performance issues and bugs in applications. The *Oracle Solaris Dynamic Tracing Guide* describes in depth how to use DTrace to observe, debug, and tune system behavior. This guide also includes a complete reference for bundled DTrace observability tools and the D programming language.

In addition to the manual, you can find links to training, articles and other resources for DTrace at the following locations:

- *DTrace Quick Start Guide: Observing Native and Web Applications in Production*
- Using DTrace with Sun Studio Tools to Understand, Analyze, Debug, and Enhance Complex Applications
- Using DTrace to Demystify Watchpoints in the Sun Studio dbx Debugger
- Tutorial: DTrace by Example where you can search for DTrace articles, such as:
- How to Add DTrace Probes to an Application
- DTrace Quick Reference Guide
- How to use Oracle Solaris DTrace from Oracle Solaris and OpenSolaris Systems
- DTrace page on BigAdmin
- OpenSolaris DTrace Community
- DTrace Topics wiki at SolarisInternals.com

Writing Java Programs for the Oracle Solaris OS

The Java software is optimized to deliver superior performance to server-side and client-side Java technology applications in an enterprise environment. The java.sun.com web site provides complete documentation for Java Platform, Standard Edition (Java SE). The SDN New to Java Programming Center is a good place to start learning about Java programming.

Supplying Platform-Independent Online Help

The JavaHelp system is a full-featured, platform-independent, extensible help system that enables you to incorporate online help in applets, components, applications, operating systems, and devices. You can also use the JavaHelp software to deliver online documentation for web applications. The JavaHelp System product page includes links for downloading JavaHelp software and the *JavaHelp System User's Guide* and other documentation. Note that the JavaHelp system is not an authoring system, but a help infrastructure. The product page also includes information about third-party help authoring tools that support the JavaHelp system. In addition, you can subscribe to the mailing list to get help from other JavaHelp users.

Programming to Specific Hardware and Configurations

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

Known Oracle Solaris Compatible Hardware

The Hardware Compatibility Lists (HCL) identifies hardware that is compatible with the Oracle Solaris 10 OS and the OpenSolaris OS. Separate lists identify systems and components that are compatible with the latest commercial versions of the Oracle Solaris OS and with the latest development versions. The HCL is available in a comma-separated value (CSV) format, for searching and sorting the list in a spreadsheet or database.

The Solaris on x86 Platforms Device Support list shows drivers and the devices they support for x86 Oracle Solaris 10 and OpenSolaris systems. Select an Oracle Solaris 10 release or an OpenSolaris release and then select a device type such as Network, Storage, or Video. Drivers of that type that are bundled in that selected release are displayed.

The Oracle Device Detection Tool reports both built-in and third-party Oracle Solaris driver availability for x86 systems.

Testing x86 Hardware for Oracle Solaris Compatibility

See Solaris on x86 Platforms Device Support for the latest list of device support that comes built in to the Oracle Solaris OS.

On any x86 system, you can use the Installation Check Tool to determine quickly whether the system is likely to be able to install the Oracle Solaris OS.

The Hardware Certification Test Suite (HCTS) includes system certification tests, controller certification tests, a command-line interface , and a browser-based user interface. HCTS tests certify systems, motherboards, and various network, storage, and serial I/O controllers to work

with the Oracle Solaris OS. HCTS collects log files, determines test pass or fail results, and generates a final test report. Hardware that passes HCTS certification testing is eligible to be included in the Hardware Compatibility List as Certified.

Developing in the x86 Assembly Language

The Oracle 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 Oracle Solaris x86 assembly language. The book also maps the instruction mnemonics of the Oracle Solaris x86 assembly language to the native x86 instruction set.

Developing in the SPARC Assembly Language

The Oracle Solaris OS provides an assembly language for the SPARC platform. The *SPARC Assembly Language Reference Manual* describes the assembler that runs on the SPARC architecture. The assembler translates source files that are in assembly language format into object files in linking format. The book describes the assembler syntax, and the executable and linking format. The relationship between hardware instructions of the SPARC architecture and the assembly language instruction set is discussed. The book also 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 Oracle Solaris OS

The Oracle 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. The book provides guidance for choosing whether to use the 32-bit Oracle Solaris application development environment or the 64-bit environment. The manual explains the similarities and differences between the two 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.

Developing for a Cluster Environment

The Oracle Solaris OS can be configured into a cluster environment to improve availability. The *Sun Cluster Data Services Developer's Guide for Solaris OS* 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 Oracle Solaris Cluster environment. For high availability in virtualized environments, see the Oracle Solaris Containers feature and Oracle VM Server for SPARC.

In addition, you can parallelize applications to enable them to reap the benefits of running in a clustered environment. Oracle Message Passing Toolkit, formerly Sun HPC ClusterTools, is a set of parallel development tools for high-end distributed memory applications. See "Using Oracle Message Passing Toolkit" on page 18 for more information.

Network Programming in the Oracle 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 Oracle 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.

ONC+ technologies consist of a family of technologies, services, and tools:

- Remote procedure calls (RPC).
- Transport-independent remote procedure calls (TI-RPC) to make RPC applications transport independent.
- External data representation (XDR), an architecture-independent specification for representing data.
- Network Information Services Plus (NIS+), the enterprise naming service in the Oracle 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 also contains code examples on the use of RPC.

Adding Device Management Information to the System Management Agent

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 extend the functionality of the agent. The manual provides information on the creation of Management Information Base (MIB) modules for SMA, and the interfaces that MIB modules use. Modules enable a specific application, device, system, or network to be managed through an SNMP management application. An appendix provides information about migrating an SEA subagent from the Solstice Enterprise Agents software for use in the System Management Agent.

Developing Web-Based Enterprise Management Applications

The *Solaris WBEM Developer's Guide* describes the components of the Oracle Solaris Web-Based Enterprise Management (WBEM) Software Development Kit (SDK). The manual 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 with the Managed Object Format (MOF) compiler
- WBEM security mechanisms

Programming with Oracle Solaris Interfaces and Frameworks

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

Multithreaded Programming in the Oracle Solaris OS

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

If you are new to multithreaded or parallel programming, see the article Making Sense of Parallel Programming Terms. The article explains terminology and contains links to additional sources of information about parallel programming.

For a more comprehensive understanding of programming with POSIX threads, consider reading the following retail books:

- *Programming with Threads* by Steve Klieman, Devang Shah, and Bart Smaalders, 1st edition, January 23, 1996.
- *Programming with POSIX Threads* by David R. Butenhof, 1st edition, May 16, 1997.

Programming Interfaces

The *Programming Interfaces Guide* describes programming interfaces that are specific to the Oracle Solaris environment.

The *Programming Interfaces Guide* has information on the following subjects:

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

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

The appcert(1) utility statically examines the Oracle Solaris library interfaces used by ELF binaries for instances of private interface usage. The apptrace(1) tool uses the link-auditing capability of the run time linker to dynamically trace library routine calls as the application runs.

Developing Custom Storage Modules for the DHCP Service

The Oracle Solaris OS provides Dynamic Host Configuration Protocol (DHCP) services. The Oracle Solaris DHCP service provides a framework that enables the development and use of custom databases for storing DHCP data. The *Solaris DHCP Service Developer's Guide* describes

how to enable the DHCP service to use additional data storage facilities. The manual enables developers to write a module to store DHCP data in a database that is not currently supported by the Oracle Solaris DHCP service. The manual gives an overview of the data access framework used by Oracle Solaris DHCP and general guidelines for developers. The book also provides sample code templates.

Developing Security Applications and Services in the Oracle Solaris OS

The *Oracle Solaris Security for Developers Guide* is for developers of applications that consume security services as well as developers of applications that provide security services. Programming interfaces are documented for the following services: PAM, SASL, GSS-API, the Oracle 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 Oracle Solaris OS includes a set of standard interfaces for developing device drivers. The interfaces are known as the DDI/DKI, or Device Driver Interface/Driver-Kernel Interface. The DDI/DKI interfaces enable you to upgrade to a new Oracle Solaris release or migrate to a new platform without recompiling your driver. These interfaces are documented in man page section 9, described in "Map to the Oracle Solaris OS Man Page Collection" on page 12.

The *Device Driver Tutorial* provides hands-on information about how to develop device drivers for the Oracle Solaris OS. This book includes step-by-step descriptions for writing, building, installing, loading, and testing simple device drivers. This book also gives an overview of the driver development environment and the tools available to develop drivers. Links to driver development resources and techniques for avoiding some driver development problems are also provided.

The *Writing Device Drivers* manual provides much more complete information about developing drivers for character-oriented devices and block-oriented devices. Specific devices such as network devices, USB devices, and SCSI target and HBA devices are covered as well.

The *Writing Device Drivers* manual includes the following additional topics:

- Multithreaded re-entrant drivers
- Driver autoconfiguration
- Programmed I/O
- Direct Memory Access (DMA)
- Power management
- Hardening drivers
- Device context management

- Compilation, installation, testing, and debugging drivers
- Guidelines for drivers for 64-bit environments

For more driver development resources, see the Device Drivers section of Software Developer Information Resources and the OpenSolaris Device Drivers Community.

Writing System Resource Management Applications

The *Solaris Containers: Resource Management and Solaris Zones Developer's Guide* describes how to write applications that partition and manage system resources such as processor sets and thread scheduling classes. This book references the programming APIs provided to partition, schedule, and set bounds on the consumption of system resources. This book provides programming examples and a discussion of programming issues to consider when writing an application. This book also includes a brief overview of Oracle Solaris Zones technology and discusses design considerations for applications that run in zones.

Developing Software for International Audiences

The Oracle Solaris OS provides an internationalization architecture to assist in the development, deployment, and 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 Oracle Solaris release to build global software products that support a variety of languages and cultural conventions.