

Pro OpenSolaris: A New Open Source OS for Linux Developers and Administrators

Copyright © 2009 by Harry J. Foxwell and Christine Tran

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-1891-3

ISBN-13 (electronic): 978-1-4302-1892-0

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editors: Michelle Lowman, Frank Pohlmann

Technical Reviewer: Sam Nicholson

Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell,

Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper,

Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Kylie Johnston

Copy Editor: Kim Wimpsett

Associate Production Director: Kari Brooks-Copony

Production Editor: Richard Ables

Compositor: Diana Van Winkle, Van Winkle Design Group

Proofreader: Nancy Sixsmith

Indexer: Becky Hornyak

Interior Designer: Diana Van Winkle, Van Winkle Design Group

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com>.

PART 1



Getting Started

OpenSolaris is a community-developed and community-supported operating system platform you can use to host your open source applications and web solutions. Although its look and feel is purposely designed to appeal to Linux developers, OpenSolaris has unique features that make it an attractive alternative. In this part of the book, you'll learn about these features and the growing community that supports this new operating system.



Introducing OpenSolaris

To change this rock into a jewel, you must change its true name.

—Kurremkarmerruk, the Master Namer, in *A Wizard of Earthsea*, by Ursula Le Guin

Welcome, open source solution developers! If you’ve purchased and are reading this book (thanks!) or if you’re scanning it in the bookstore (please keep reading!), you probably think of yourself as a “Solaris developer” or a “Linux developer.” You Solaris users, administrators, and developers may already know that as Sun Microsystems’ commercial version of Solaris continues to evolve, it will look more and more like today’s OpenSolaris, so it’s great that you’ve chosen this book to learn more about it.

But if you call yourself a “Linux developer,” is that your *true name*? Are you really among the less than 1 percent of all Linux users who download the kernel source code, change it or add to it, recompile it, and use the result for highly specialized purposes? Or are you truly an *open source solutions developer* who just happens to use one of the popular Linux operating system kernel distributions as the foundation for your work? If you’re among the latter, *why* have you chosen Linux (if, in fact, you were the one who made the choice)?

The typical answers that we hear when we ask this question is that Linux distributions:

- “are free,”
- “are open source,”
- “support commodity platforms,”
- “run high quality application software,” and
- “have vendor support available.”

Well, so do other open source operating systems and their associated user and development environments. This book is about another such operating system with these characteristics, OpenSolaris.

If you're an open source solutions developer, you should realize that the true name of your development environment is not limited to Linux and that there is generally no such thing as a "Linux application." Rather, you're likely using a packaged distribution of open source user interface tools and developer tools that have been compiled for the Linux kernel. Red Hat's Enterprise Linux, Novell's SUSE Linux Enterprise, and Ubuntu Linux are among the most popular of such distributions. So-called Linux applications are really "open source applications" that happen to have been compiled and distributed with the Linux kernel; they could just as easily have been compiled for other kernels, as indeed many of them have. Some of these applications and tools have even been compiled for so-called proprietary environments such as Microsoft Windows. Indeed, calling an open source application like the Apache web server a "Linux application" actually limits its consideration for use on other platforms. There's even a term for this view of the typical collection of web software technologies that Linux developers use—the *LAMP stack*, comprising Linux, Apache, MySQL, and PHP—and there are many books on this specific combination of software. In this book, we'll be showing you that there are some advantages to substituting something else for that *L*.

To make it clear what we're emphasizing, look at the typical open source stack of software you're probably using in Table 1-1. The Linux kernel *can* provide the operating system foundation for the stack, but it's *not required*; other commercial or community operating systems such as AIX, HP-UX, Solaris 10, OS X, FreeBSD, or OpenSolaris are widely used as well, since virtually all the "standard" open source tools in the stack are available for these systems. So, it's useful to examine the relative advantages of the available operating system kernels.

Table 1-1. *A Typical Open Source User and Developer Stack*

Stack Component	Tools and Utilities
User and developer tools	Firefox browser, Thunderbird email, OpenOffice, gcc and Java compilers; Perl, Ruby, other languages; IDEs, vi/vim and emacs, bash and csh shells
Desktop environment	GNOME and/or KDE Desktop tools
Window system	X Window System and tools
Services and utilities	UNIX and GNU programs, services, and utilities, web servers, application servers, databases, Apache, JBoss, GlassFish, MySQL
APIs and system libraries	POSIX libraries and extensions, language libraries
Operating system kernel and services	Process scheduling, security, network, resource management, virtualization support; Linux, FreeBSD, OS X, or OpenSolaris
System hardware	Multicore, multithreaded processors; Intel, AMD, SPARC, PowerPC

Clearly, what you are most concerned with when designing and creating your open source solutions is the availability of a familiar, productive, efficient, and comprehensive user and development environment. This is provided by the open source components that have been compiled for and integrated with the underlying operating system kernel. And of course you'll want the operating system to provide high-quality services for performance, scalability, and security; special features such as virtualization support, debugging aids, and resource management would be a bonus. The various open source operating system kernels, along with their "distribution" tools and accessories, each have specific advantages for open source development and deployment. This book focuses on the advantages of using the OpenSolaris distribution for such work.

The *True Name* of Open Source Software

While we're discussing open source software, let's call it by its true name. The term *open source software* is often misunderstood to mean Linux software, free software, shareware, or other terms that emphasize the method of acquiring the programs or the fact that you can obtain and inspect the programs' source code.

The names we give to concepts can either enhance understanding or confuse practitioners and users. The term *virtualization*, for example, includes some technologies that have no true virtualization components; we prefer *workload containment*, which includes true virtualization but also includes other concepts.

Open source is another somewhat misused and misunderstood term that focuses on the wrong component of modern software development. A better phrase to use is *community-developed software*, which focuses on the true methodology (yes, you obviously can't have community development without access to the source code, but the source code itself should not be the main focus...Microsoft could publish the entire Vista source and still severely restrict its sharing and use!). We'll have more on the community aspect of OpenSolaris later.

We also avoid the term *free* and prefer using *freely sharable* or some other words that focus on the community rather than on the object of the community's activity. Then we can emphasize the distinction between community-supported software and commercially supported software, *regardless of the software's origin* (community developers and/or commercial developers).

When you think about it, much of what so-called open source businesses such as Red Hat, Novell, Sun, HP, IBM, and others are selling is commercially supported community software. It's not as pretty sounding as *open source software*, but it's more accurate nevertheless. However, we do not expect businesses or developers to change their vocabulary simply for the sake of accuracy.

Speaking of the commercial software developers, it's very interesting to observe the recent trend of such companies acquiring the trademarks, distribution rights, and support rights to popular and mature community-developed tools. For example:

- Novell acquires SuSE Linux (2004).
- Red Hat acquires JBoss (2006).
- Oracle acquires the Sleepycat database (2006).
- Citrix acquires XenSource and Xen (2007).
- Sun Microsystems acquires MySQL (2008), Lustre (2007), and Innotek/VirtualBox (2008).

These vendors and others are also taking advantage of the community development model for their own internally originated software in order to expand their user bases and to obtain the hoped-for benefits of community support. Sun, notably, has released much of its own as well as its acquired software under various OSI-approved open source licenses, including the following, to name just a few:

- Java
- GlassFish Application Server
- NetBeans IDE
- StarOffice (now OpenOffice)
- Grid Engine
- Solaris 10 (now OpenSolaris)

You should therefore anticipate a future open source development environment with a variety of support and distribution models, from the traditional “totally free” approach to the multitier approach that provides both “community” and “enterprise” editions and support levels for open source software on a wide selection of hardware and operating system platforms. Sun Microsystems does provide subscription support services for OpenSolaris (see <http://www.sun.com/service/opensolaris/faq.xml>), and these services are still evolving as Sun enhances its operating system offerings from Solaris 10 to OpenSolaris. In the meantime, we'll concentrate on describing the features and benefits of this new, community-developed operating system.

What You'll Find and Learn in This Book

Pro OpenSolaris assumes you're already generally familiar with either Linux or Solaris 10 as an end user, software developer, or system administrator. It explains the origin and intent of Sun Microsystems' effort to move from the internally developed Solaris 10 operating system to the community-developed OpenSolaris. Later in this chapter, we'll detail the projects and activities of the OpenSolaris community and will encourage you to become part of it. Chapter 2 will highlight the advantages of using OpenSolaris as your solutions development platform, including advanced technologies such as the DTrace observability tool, the ZFS file system, virtualization with zones, and Service Management, all typically not found in any of today's Linux distributions. In Chapters 3 and 4, we'll discuss the various options for installing OpenSolaris, and we'll review the common features that both Linux and Solaris users will find familiar, such as the GNOME user environment shown in Figure 1-1. Then we'll get to work, showing you how to exploit OpenSolaris's unique technologies such as containers, including how to set up and use the AMP software stack for OpenSolaris-hosted web solutions.

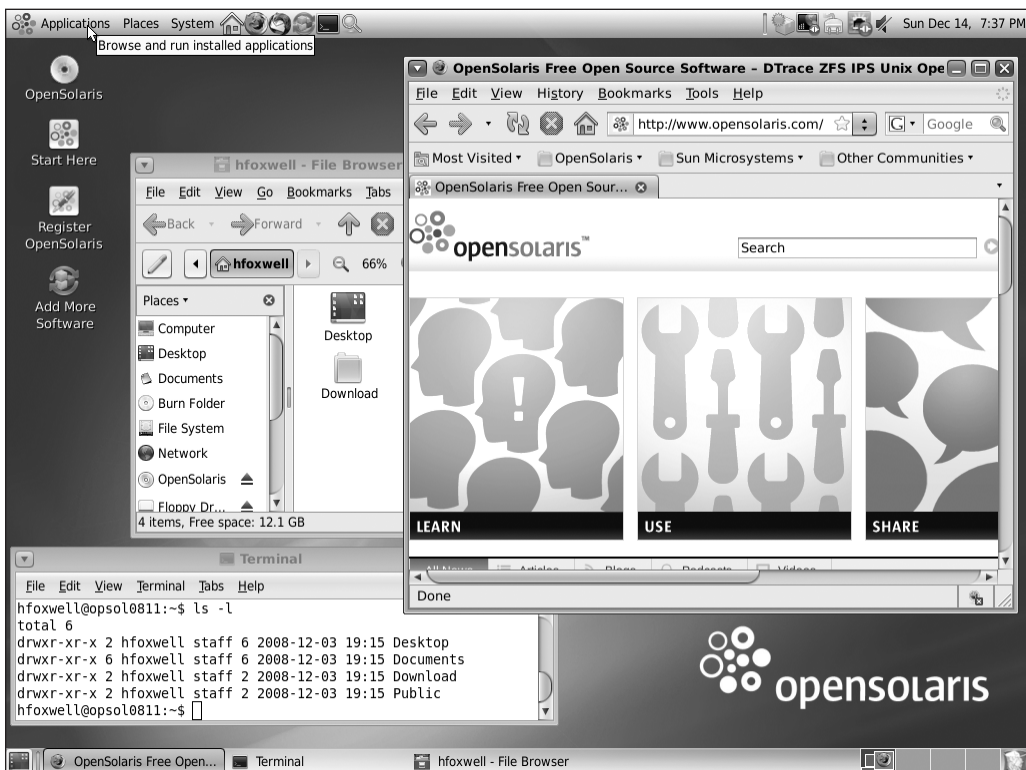


Figure 1-1. A typical OpenSolaris user session

We won't be digging into the internals of the OpenSolaris source code or covering how to write OpenSolaris device drivers, because this book focuses on *using* OpenSolaris rather than its design or development. Readers interested in those topics can go directly to the source at <http://www.opensolaris.org/os/get/> and <http://www.solarisinternals.com> or to the OpenSolaris device drivers community at http://www.opensolaris.org/os/community/device_drivers/. If you're willing and able to contribute new features or bug fixes to the OpenSolaris community, there is an active contributor community FAQ on how to get involved (see Figure 1-2).



Figure 1-2. *The OpenSolaris contributing developers FAQ*

Don't forget to check out the other references in Appendix A for topics not covered here.

The Origin of OpenSolaris

OpenSolaris has a long and rather unique history as an open source operating system, and its origins go all the way back to the early days of UNIX, which, ironically, was one of the first major open source projects. We don't need to review all of its detailed history except to highlight some key events relevant to today's OpenSolaris:

- **1965–1972:** Ken Thompson and Dennis Ritchie develop UNIX at AT&T's Bell Laboratories; AT&T distributes UNIX source code to universities and industry users.
- **1976–1979:** Bill Joy at UC Berkeley enhances UNIX memory management; AT&T continues UNIX technical development.
- **1982–1983:** AT&T releases commercial UNIX System V; Sun Microsystems founded and releases UNIX-based SunOS.
- **1984–1987:** AT&T releases additional versions of UNIX System V.
- **1988–1992:** AT&T works with Sun on UNIX development; Open Software Foundation and UNIX International formed; Novell purchases AT&T's UNIX Software Lab; *Sun acquires rights to UNIX SVR4 code for distribution of source, binaries, and derivatives.*
- **1991:** Linus Torvalds releases the first implementation of Linux.
- **1994–1995:** First Red Hat Linux distribution released; Red Hat Software founded, first SuSE Linux distribution released.
- **1999:** Sun Microsystems announces intent to release Solaris under open source license.
- **2001:** National Security Agency (NSA) releases Linux-based SELinux.
- **2002:** Sun Microsystems reenters x86 server market, considers own Linux distribution, and later decides to OEM and support Red Hat and SuSE Linux.
- **2004:** Novell acquires SuSE Linux.
- **2005:** Solaris 10 is released, which is the first release of Solaris source code under the OSI-approved CDDL license; OpenSolaris.org founded.
- **2007:** Sun Microsystems hires Debian developer Ian Murdock to guide the OpenSolaris project.
- **2008:** First OpenSolaris binary is release, with source code; IDC reports Sun rises to no. 4 provider of x86 servers; NSA announces collaboration with OpenSolaris community to integrate mandatory access control (MAC), based on the Flux Advanced Security Kernel (Flask) architecture into OpenSolaris.

Being derived from the first UNIX, OpenSolaris has evolved from one of the original collaborations between the industry and the general software development community, passing through a period of commercial ownership with restricted access to the source code and arriving today as an enterprise-quality open source operating system that is developed, distributed, and supported by Sun, by other technology vendors, and by a growing community of users and developers.

Why did Sun start down this open source path with its highly regarded core operating system technology? In part, the answer has to do with you, the developer. In recent years, the two largest groups of developers have been for Windows and for Linux. And although today's Solaris 10 arguably has compelling advantages as an open source deployment platform, one of its target developer groups—Linux users and programmers—found its user interface and tools to be unfamiliar and sometimes lacking when compared to what they were accustomed to in modern Linux distributions.

Sun has now started what might be called the “Linuxification” of Solaris. Future versions of Solaris will look more like today's OpenSolaris, and that's good news for Linux users and developers because they now have a powerful yet familiar alternative environment on which to host their open source solutions. Obviously, Sun's strategy is to attract more developers to OpenSolaris, and that appears to be working considering the rapid growth of the OpenSolaris.org community.

Goals and Future Directions

Having finally released a stable, well-designed binary distribution of OpenSolaris, what's next? Sun would certainly like to preserve the high quality and popularity of Solaris 10 while transforming it into the preferred community-developed operating system. It wants to keep Solaris's scalability, stability, and binary compatibility features; develop a profitable product support model; and continue to expand OpenSolaris's user base. It also wants to continually add to the operating system's list of useful features, both those unique to OpenSolaris and those popular in Linux distributions.

According to the public road maps for Solaris 10 and OpenSolaris on <http://www.sun.com> and on <http://www.opensolaris.org>, you'll eventually see the expansion of Sun's OpenSolaris support subscription offerings, support for SPARC platforms (especially for the new multicore chips), a growing base of ISVs supporting their applications on OpenSolaris, and encouragement of contributing community software developers. Sun is also increasing its focus on the academic research and student communities. There is already a comprehensive OpenSolaris-based curriculum for university computer science courses that some schools have started to use, and hundreds of thousands of students and computer science faculty from around the world have downloaded OpenSolaris for their research and coursework.

New builds of OpenSolaris will be posted on the web site every two weeks. As with Solaris, these builds include bug fixes, new features, and support for newly released

hardware. Officially supported, stable binary releases, such as the 2008.11 release, are targeted for a 6-month update cycle, along with an 18-month support subscription cycle for each release. Also on the road map are enhancements of the OpenSolaris application software repositories' content and features, including specialized repositories for supported, experimental, and ISV packages.

Note There have been several Solaris-related binary release downloads available: Solaris 10 for x86 and SPARC, the Solaris Express Community and Developer Edition releases, and the OpenSolaris releases. The Solaris Express releases, which gave periodic snapshots of future Solaris 10 technologies, will be replaced by the OpenSolaris distributions for x86 and SPARC.

That Troublesome CDDL License

Nothing seems to arouse the passions of open source developers more than discussions of how source code is licensed. Whole books, thousands of web pages, and innumerable blogs argue the merits and deficiencies of the various “open source” or “free software” licenses. The Open Source Initiative (OSI) organization, which works with community software developers, is the caretaker of the official definition of *open source* and has approved more than 70 different licenses for such software. Two of these approved licenses, the GNU General Public License (GPL) and the Common Development and Distribution License (CDDL), are the focus of much of the controversy over how Sun has released the code for OpenSolaris.

It is important to understand how community-developed source code is licensed if you use such code to build new programs. To put it as briefly as possible, the GPL's goal is to encourage the broadest sharing of community-developed software; all projects using GPL-licensed code, including derived works, require the public sharing of any changes or enhancements to the code.

After more than 5 years of legal review and documentation effort, Sun released the nearly 10 million lines of source code for OpenSolaris under the file-based CDDL, which essentially requires that source files derived from common files must be shared but that executables of derived works may use other licenses, including those that permit protection of software patents, which many GPL proponents consider “evil.”

Under the CDDL, you can combine your program files with those covered by other licenses, you can release your code under more than one license, you're *not* required to release the source of your “proprietary” value-added code, and you can distribute and sell binaries derived from OpenSolaris source files.

The controversy will no doubt continue, and the CDDL might be reexamined for new OpenSolaris components in light of new business requirements and community needs and the development of new licenses such as the GPLv3, which attempts to address some of the concerns about the commercialization of open source software.

The OpenSolaris Community: OpenSolaris.org

No open source project survives for long without a large, actively contributing community of developers and users. Founded by Sun in 2005, OpenSolaris.org now includes more than 150,000 registered community members, more than 300 discussion groups with more than 250,000 postings, and nearly 100 registered OpenSolaris User Groups worldwide. More than half of the visitors to the OpenSolaris web site, shown in Figure 1-3, are from outside the United States, including many from Germany, the United Kingdom, Japan, India, China, and South America.

OpenSolaris community members support the development and deployment of the operating system and tools by participating in online help forums and mailing lists, promoting and explaining the project and its activities to the press and to developer communities, creating and contributing marketing materials, and of course writing and debugging kernel components and user applications and tools. In addition to the Sun kernel development engineers who now work on Solaris and OpenSolaris almost exclusively through the public web sites, hundreds of contributors who do not work for Sun are refining and adding to this project.

You might wonder, who decides what goes into OpenSolaris? Like with Linux, Java, JBoss, Xen, MySQL, and other open source software, it's the owner of the trademark who has the final say of what that trademark means. Linus Torvalds is the owner of the Linux trademark and is therefore the "benevolent dictator" for what goes into the Linux kernel. Sun Microsystems, advised by members of the Java Community Process (JCP, at <http://www.jcp.org>), does the same for open source Java. Similarly, Sun currently is the final arbiter of OpenSolaris content and features, with significant input from the OpenSolaris Community Advisory Board (CAB) and community members, whose charter is "...the collaborative production of open source software related to the OpenSolaris family of operating systems and committed to fostering the evolution and adoption of the OpenSolaris code base." The CAB has a community-developed governance charter and open elections, along with well-defined roles and responsibilities of board members, regular members, and technical contributors to projects.

OpenSolaris users, developers, and contributors have gathered for tutorials, technical presentations, and BoF sessions at many of the traditional Linux and annual open source conferences and workshops such as JavaOne, OSCON, ApacheCon, CommunityOne, and Usenix, as well as at Tech Days sponsored around the world by Sun in Europe, India, China, the United States, and Japan. As the OpenSolaris community grows, active users are posting wikis and blogs to share what they've learned. One of the most popular blogs is the OpenSolaris Observatory at <http://blogs.sun.com/observatory>, which includes video demonstrations and tutorials (see Figure 1-4).

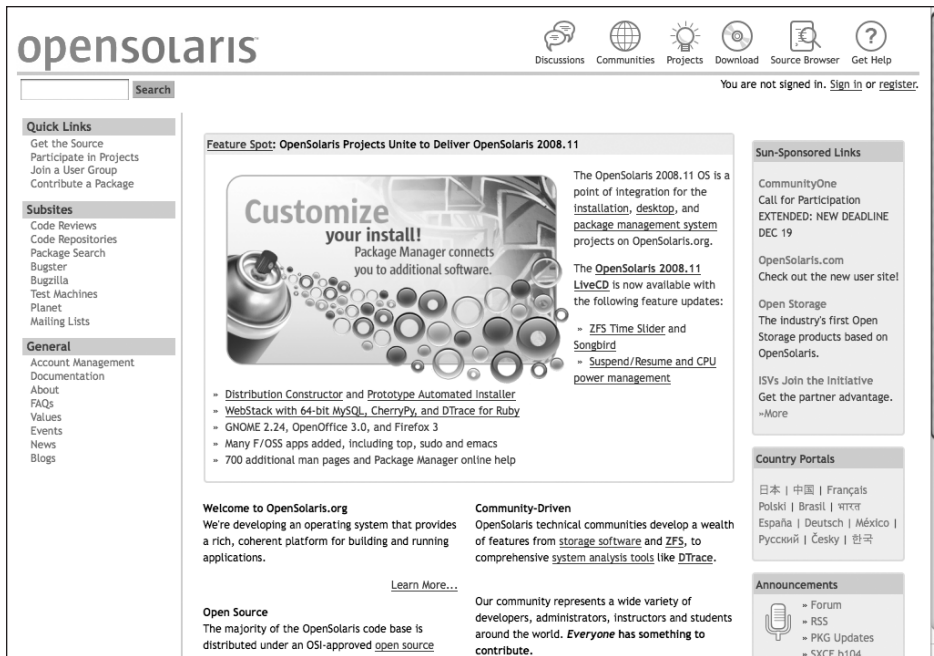


Figure 1-3. The OpenSolaris community web page

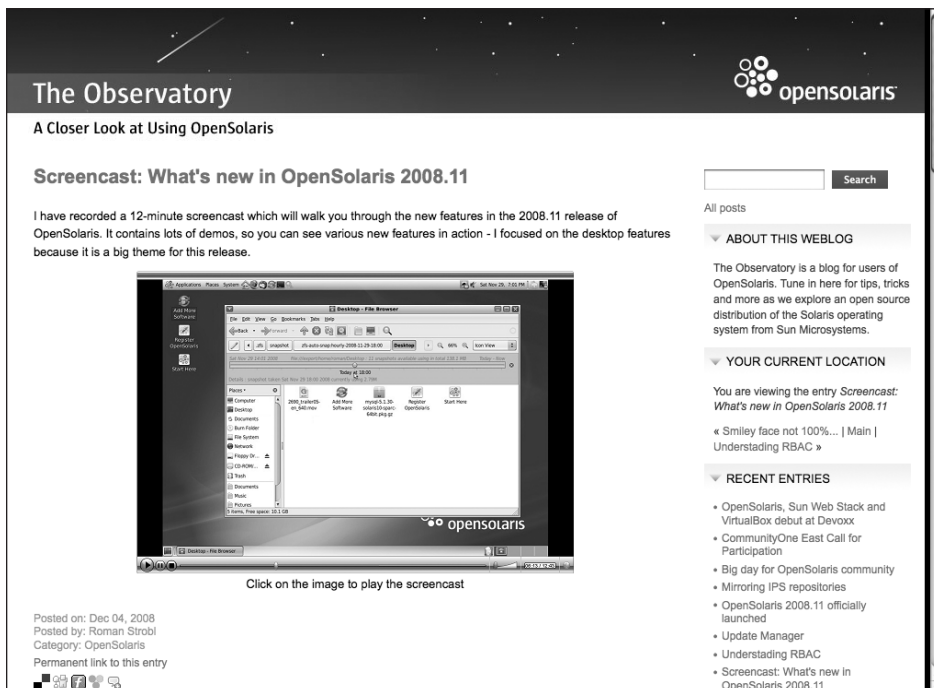


Figure 1-4. The OpenSolaris Observatory blog

If you're interested in becoming a contributor to the OpenSolaris project, there is ample opportunity and encouragement to participate in any of the nearly 50 special-interest communities focused on issues such as accessibility, documentation, games, high-performance computing, performance, security, and virtualization, to name but a few. The process and web sites for submitting bug fixes and code changes is well defined (Figure 1-5) and includes many suggested requests for enhancement (RFEs) that developers can choose from if they're interested in becoming involved in improving OpenSolaris.

opensolaris™

Discussions Communities Projects Download Source Browser

About OpenSolaris
Project Overview
FAQ Center
Roadmap
Governing Board
Site Map

Communities
Portal
Nevada
Community
All Communities

Projects
Portal
All Projects

Code
Source Browser
Download
Tools
Bug Database

Connect
Register
Mail Lists
Jive Forums
Documentation
User Groups
Blogs
Planet
Announcements
Events
News
Related Links

Bug Database Search

Type: Sort By:

State: Show search results per page.

BugID:

Keyword:

Text Search:

Category:

Subcategory:

New or changed in the last days.

☐ Only show synopsis in results page.

Report a bug or request a feature
If we don't know about your problem, we can't fix it. If you've isolated a problem that you think we're causing, and you can't find it here, [submit a bug!](#) Make sure you include all the relevant information including all the details needed to reproduce the problem. Submissions will be verified and prioritized. (Please note that bug fixes are not guaranteed.)

Want to help? Want to get involved? Try an [oss-bite-size](#) bug. Bugs marked with the oss-bite-size keyword tend to be small and self-contained and are a good place to start to get familiar with OpenSolaris source code. If you choose to work on one, please send email to the request-sponsor alias so the bug can be marked and people will know you're working on it.

[Terms of Use](#) | [Privacy](#) | [Trademarks](#) | [Copyright Policy](#) | [Site Guidelines](#) | [Help](#)
Your use of this web site or any of its content or software indicates your agreement to be bound by these Terms of Use.
Copyright © 1995-2008 Sun Microsystems, Inc.
Our lawyer is making us say that OpenSolaris is a trademark of Sun Microsystems, Inc.

Figure 1-5. *The OpenSolaris bug-tracking and RFE site*

Essential URLs

Be sure to visit these web sites frequently to learn what's going on in the OpenSolaris communities and to learn about new software features, bug fixes, and opportunities to participate:

- The OpenSolaris community: <http://www.opensolaris.org>
- The OpenSolaris FAQs: <http://www.opensolaris.org/os/about/faq/>
- Getting started as an OpenSolaris contributor: http://www.opensolaris.org/os/about/faq/getting_started_developers/
- OpenSolaris downloads and learning resources: <http://www.opensolaris.com/learn/>

Summary

OpenSolaris is already a successful open source project because of its historical roots in Sun's Solaris operating system and its large and active development community. You can participate as an end user and application developer or as a contributor to OpenSolaris to improve its features, documentation, or usage. Next, you'll learn about some of the technologies that make OpenSolaris a good choice for open source development.

