Introduction

Bill Calkins has been training Solaris system administrators for more than 15 years. This book contains the training material that he uses in his basic and advanced Solaris administration courses that, over the years, have helped thousands of Solaris administrators become certified. This is our second edition of the Solaris 10 System Administration Exam Prep. It covers updates that Sun has made to the Solaris 10 operating environment as of the October 2008 release. It began with the Training Guide for Solaris 2.6, 7, 8, and 9 and is now the Exam Prep for Solaris 10. Instructors from universities and training organizations around the world have used the book as courseware in their Solaris administration courses. In addition, administrators from around the world have used this book for self-study when instruction from a Sun training center is either unavailable or not within their budget. Many of you have written with your success stories, suggestions, and comments. Your suggestions are what keep making this guide more valuable.

The Solaris 10 System Administration Exam Prep books, Parts I and II, provide training materials for anyone interested in becoming a Sun Certified System Administrator (SCSA) for Solaris 10. When used as a study guide, these two books will save you a great deal of time and effort searching for information you will need to know when taking the exam. Each book covers the exam objectives in enough detail for inexperienced administrators to learn the objectives and apply the knowledge to real-life scenarios. Experienced readers will find the material in these books complete and concise, making it a valuable study guide for the Sun Certified System Administrator exams.

This book is not a cheat sheet or cram session for the exam; it is a training manual. In other words, it does not merely give answers to the questions you will be asked on the exam. We have made certain that this book addresses the exam objectives in detail, from start to finish. If you are unsure about the objectives on the exams, this book teaches you what you need to know. After reading each chapter, assess your knowledge of the material covered using the review questions at the end of the chapter. When you have completed reading a section, use the practice exam at the end of the book and the ExamGear test engine on the CD-ROM to assess your knowledge of the objectives covered on each exam. This CD-ROM contains sample questions similar to what you are likely to see on the real exams. More sample questions are available at http://www.UnixEd.com, so make sure you visit this site to find additional training and study materials.
How This Book Helps You

This book teaches you advanced topics in administering the Solaris 10 operating system. It offers you a self-guided training course of all the areas covered on the CX-310-202 certification exam by installing, configuring, and administering the Solaris 10 operating environment. You will learn the specific skills that are required to administer a system and, specifically, to pass the second part of the Sun Certified System Administrator exam for Solaris 10 (CX-310-202). If you are an experienced administrator who is upgrading an existing Solaris certification, you’ll find in-depth coverage of the new topics you need to learn for the CX-310-203 upgrade exam in both the SCSA Solaris 10 OS CX-310-200 and CX-310-202 Exam Prep books.

Throughout the book, we provide helpful tips and real-world examples that we have encountered as system administrators. In addition, we provide useful, real-world exercises to help you practice the material you have learned. This book is set up as follows:

- **Organization**: This book is organized according to individual exam objectives. Every objective you need to know to install, configure, and administer a Solaris 10 system is in this book. We have attempted to present the objectives in an order that is as close as possible to that listed by Sun. However, we have not hesitated to reorganize them as needed to make the material as easy as possible for you to learn. We have also attempted to make the information accessible in the following ways:
  - This book includes the full list of exam topics and objectives.
  - Read the “Study and Exam Prep Tips” element early on to help develop study strategies. This element provides you with valuable exam-day tips and information on exam/question formats such as adaptive tests and case study-based questions.
  - Each chapter begins with a list of the objectives to be covered, exactly as they are defined by Sun. Throughout each section, material that is directly related to the exam objectives is identified.
  - Each chapter also begins with an outline that provides you with an overview of the material and the page numbers where particular topics can be found.

- **Instructional features**: This book is designed to provide you with multiple ways to learn and reinforce the exam material. The following are some of the helpful methods:
  - **Objective explanations**: As mentioned, each chapter begins with a list of the objectives covered in the chapter.
  - **Study strategies**: The beginning of each chapter also includes strategies for studying and retaining the material in the chapter, particularly as it is addressed on the exam.
  - **Exam Alerts**: Throughout each chapter you’ll find exam tips that will help you prepare for exam day. These tips were written by those who have already taken the Solaris 10 certification exams.
- **Key Terms**: A list of key terms appears near the end of each chapter.

- **Notes**: These contain various types of useful information, such as tips on technology or administrative practices, historical background on terms and technologies, or side commentary on industry issues.

- **Cautions**: When you use sophisticated information technology, mistakes or even catastrophes are always possible because of improper application of the technology. Cautions alert you to such potential problems.

- **Step By Steps**: These are hands-on lab exercises that walk you through a particular task or function relevant to the exam objectives.

- **Exercises**: Found near the end of the chapters, exercises are performance-based opportunities for you to learn and assess your knowledge.

- **Suggested Reading and Resources**: At the end of each chapter is a list of additional resources that you can use if you are interested in going beyond the objectives and learning more about the topics presented in the chapter.

- **Extensive practice test options**: The book provides numerous opportunities for you to assess your knowledge and practice for the exam. The practice options include the following:

  - **Exam questions**: Each chapter ends with questions. They allow you to quickly assess your comprehension of what you just read in the chapter. Answers to the questions are provided in a separate element titled “Answers to Exam Questions.”

  - **Practice exam**: A practice exam is included in Part II, “Final Review,” for each exam (as discussed in a moment).

  - **ExamGear**: The ExamGear software included on the CD-ROM provides further practice questions.

**NOTE**

ExamGear software For a complete description of the ExamGear test engine, see Appendix A, “What’s on the CD-ROM.”

- **Final Review**: This part of the book provides you with three valuable tools for preparing for the exam:

  - **Fast Facts**: This condensed version of the information contained in the book will prove extremely useful for last-minute review.

  - **Practice Exam**: A full practice exam is included, with questions written in styles similar to those used on the actual exam. Use the practice exam to assess your readiness for the real exam.
Answers to Practice Exam: This element provides the answers to the full practice exam, with detailed explanations. These should help you assess your strengths and weaknesses.

Appendixes: The book contains valuable appendixes as well, including a glossary and a description of what is on the CD-ROM (Appendix A).

These and all the other book features mentioned previously will enable you to thoroughly prepare for the exam.

Conventions Used in This Book

Commands: In the steps and examples, the commands you type are displayed in a special monospace font.

Arguments, options, and <cr>: In command syntax, command options and arguments are enclosed in <>. The words within the <> stand for what you will actually type. You don’t type the <>. The <cr> that follows the command means to press Enter. You don’t type the <cr>.

```
lp -d<printer name> <filename> <cr>
```

Using the mouse: When using menus and windows, you select items with the mouse. Here is the default mapping for a three-button mouse:

- Left button: Select
- Middle button: Transfer/adjust
- Right button: Menu

You use the Select button to select objects and activate controls. The middle mouse button is configured for either Transfer or Adjust. By default, it is set up for Transfer, which means that you use this button to drag or drop list or text items. You use the left mouse button to highlight text, and then you use the middle button to move the text to another window or to reissue a command. The middle button can also be used to move windows around on the screen. You use the right mouse button, the Menu button, to display and choose options from pop-up menus.

Menu options: The names of menus and the options that appear on them are separated by a comma. For example, “Select File, Open” means to pull down the File menu and choose the Open option.

Code continuation character: When a line of code is too long to fit on one line of the book, it is broken and continued to the next line. The continuation is preceded by a backslash.
Introduction

**Audience**

This book is the second book in a series designed for anyone who has a basic understanding of UNIX and wants to learn more about Solaris system administration. Whether or not you plan to become certified, the *Solaris 10 System Administration Exam Prep* books, Part I and Part II, are the starting point to becoming a Solaris System Administrator. It’s the same training material that the author uses in his Solaris 10 Intermediate and Advanced System Administration courses. This book covers advanced system administration topics you need to know before you begin administering the Solaris operating system. Our goal is to present the material in an easy-to-follow format, with text that is easy to read and understand. The only prerequisite is that you have read my *Solaris 10 System Administration Exam Prep Part I* book.

This book is intended for experienced system administrators who want to become certified, update their current Solaris certification, or simply learn about the features of the Solaris 10 operating environment. To pass the CX-310-202 and CX-310-203 certification exams, you need a solid understanding of the fundamentals of administering Solaris 10. This book helps you review the fundamentals required to pass the certification exam.

**The Sun Certified System Administrator Exams**

To become a Sun Certified System Administrator, you need to pass two exams: CX-310-200 (Part I) and CX-310-202 (Part II). This book covers the material on the Part II exam. You must pass the CX-310-200 exam before taking the CX-310-202 exam. You will not receive a certificate until you have passed both examinations. Also, if you are already certified in Solaris 2.6, 7, 8, or 9, you need to know the material covered in this book as well as in *Solaris 10 System Administration Exam Prep: CX-310-200 Part I* to take the upgrade exam, CX-310-203, to become certified on Solaris 10.

Beware of fakes. We have seen some websites promoting their own certification programs, so be sure to evaluate them carefully. Certification programs promoted by these sites are not the same as the Sun certification program. You will not receive a certificate from Sun until you pass Sun’s exams from a certified Sun testing center. Go to my website (www.UnixEd.com) for links to the real exams and information on Sun’s certification program if you are in doubt. In addition, feel free to visit our online Solaris certification discussion forum at www.UnixEd.com, where you can ask me questions directly.
Summary

It's not uncommon for Sun to change the exam objectives or to shift them around after the exams have been published. We highly recommend that before you begin reading this book, you visit my website at www.UnixEd.com to get the most up-to-date list of exam objectives, the errata for this book, up-to-date sample exam questions, and any other last-minute notes about these exams. We will provide all the information you need to pass the exam—all you need to do is devote the time. Learning the objectives is the first step; the next step is to practice. You need access to both SPARC and x86/x64-based systems running Solaris 10 so that you can practice what you have learned. Unless you have a supernatural memory, it’s difficult to pass the exams without practice.

In the back of this book is the ExamGear software test CD that will prepare you for the questions you might see on the exam. The CD-ROM-based test engine was designed by educational experts to help you learn as you test. It is a preview of the types of questions to expect on the exams and tests your knowledge of all the exam objectives. If you are weak in any area, the sample questions will help you identify that area so that you can go back to the appropriate chapter and study the topic. Each question on the CD-ROM has a flash card to help you in case you get stuck. This flash card contains brief, concise textbook excerpts that explain why each answer is correct so that you can learn as you test.

Also, for an additional cost, you can purchase more questions for the ExamGear test engine from our website. You'll receive hundreds of questions that will take you deep into each exam objective. This will give you a comprehensive skills assessment and help you evaluate your readiness and retention of the materials.

Advice on Taking the Exam

More extensive tips are found in the “Study and Exam Prep Tips” element and throughout the book, but keep in mind the following advice as you study for the exam:

▶ **Read all the material.** This book includes information not reflected in the exam objectives to better prepare you for the exam and for real-world experiences. Read all the material to benefit from this.

▶ **Do the step-by-step lab exercises and complete the exercises in each chapter.** This will help you gain experience and prepare you for the scenario-type questions that you will encounter.

▶ **Use the questions to assess your knowledge.** Each chapter contains review questions and exam questions. Use these to assess your knowledge and determine where you need to review material.
Review the exam objectives. Develop your own questions and examples for each topic listed. If you can develop and answer several questions for each topic, you should not find it difficult to pass the exam.

Relax and sleep before taking the exam. The time for taking the examination is limited. However, if you have prepared and you know Solaris network administration, you will have plenty of time to answer all the questions. Be sure to sleep well the night before the exam because of the stress that the time limitations put on you.

Review all the material in the “Fast Facts” element the night before or the morning you take the exam.

If you don’t know the answer to a question, just skip it and don’t waste time. You need to complete the exam in the time allotted. Don’t be lazy during the examination; answer all the questions as quickly as possible. Any unfinished questions will be marked incorrect.

Visit my website, www.UnixEd.com. It contains the following:

• Late-breaking changes that Sun might make to the exam or the objectives. You can expect Sun to change the exams frequently. Make sure you check my website before taking the exam.

• A FAQs page with frequently asked questions and errata regarding this book or the exams.

• Links to other informative websites.

• Additional practice questions and sample exams for the ExamGear test engine. The ExamGear test engine has hundreds of questions that you can use to further assess your retention of the material presented in the book. The exams feature electronic flash cards that take the place of those sticky notes that you’ve used as bookmarks throughout the book. Don’t attempt the real exam until you can pass every section of the practice exams with a 95% or better score.

• An online forum where you can discuss certification-related issues with me and other system administrators, including some who have already taken the exam.

• Additional study materials, training programs, and online seminars related to Solaris certification.

• You can also email me directly from this website with questions or comments about this book. I always try to answer each one.

When you feel confident, take the real exams and become certified. Don’t forget to drop me an email and let me know how you did on the exam (guru@UnixEd.com).
Objectives

The following test objectives for Exam CX-310-202 are covered in this chapter:

Configure Role-Based Access Control (RBAC), including assigning rights profiles, roles, and authorizations to users.

- This chapter describes Role-Based Access Control (RBAC), and identifies the four main databases involved with RBAC. The system administrator needs to understand the function and structure of each of these databases and how to apply the RBAC functionality in real-world situations.

Analyze RBAC configuration file summaries and manage RBAC using the command line.

- You will see how to assign a role to a user and use rights profiles by using commands that are described in this chapter. These can greatly assist the system administrator when managing a large number of rights that are to be assigned to a number of users.

Explain syslog function fundamentals, and configure and manage the /etc/syslog.conf file and syslog messaging.

- This chapter describes the basics of system messaging in the Solaris operating environment, introduces the daemon responsible for managing the messaging, and describes the configuration file that determines what information is logged and where it is stored. It also describes the new method of restarting/refreshing the syslog process when changes are made to its configuration file.
Study Strategies

The following strategies will help you prepare for the test:

- As you study this chapter, it’s important that you practice each exercise and each command that is presented on a Solaris system. Hands-on experience is important when learning these topics, so practice until you can repeat the procedures from memory.

- Be sure you understand each command and be prepared to match the command to the correct description.

- Be sure you know all the terms listed in the “Key Terms” section near the end of this chapter. Pay special attention to the databases used in Role-Based Access Control (RBAC) and the uses and format of each. Be prepared to match the terms presented in this chapter with the correct description.

- Finally, you must understand the concept of system messaging—its purpose, how it works, and how to configure and manage it.
Introduction

This chapter covers two main topics—Role-Based Access Control (RBAC) and system messaging (syslog). These are both related in that they participate in the securing and monitoring of systems in a Solaris environment. The use of Role-Based Access Control makes the delegation of authorizations much easier for the system administrator to manage, as groups of privileges can easily be given to a role through the use of profiles. Also, the use of roles means that a user has to first log in using his or her normal ID and then use the su command to gain access to the role (and therefore assigned privileges). This has the advantage of being logged and therefore helps establish accountability. The system messaging service (syslog) stores important system and security messages and is fully configurable. The system administrator can tune the service so that certain messages are delivered to several places (such as a log file, a message, and the system console), greatly increasing the chances of it being noticed quickly.

Role-Based Access Control (RBAC)

Objectives

- Configure Role-Based Access Control (RBAC) including assigning rights profiles, roles, and authorizations to users.
- Analyze RBAC configuration file summaries and manage RBAC using the command line.

Granting superuser access to nonroot users has always been an issue in UNIX systems. In the past, you had to rely on a third-party package, such as sudo, to provide this functionality. The problem was that sudo was an unsupported piece of freeware that had to be downloaded from the Internet and installed onto your system. In extreme cases, the system administrator had to set the setuid permission bit on the file so that a user could execute the command as root.

With Role-Based Access Control (RBAC) in the Solaris 10 operating environment, administrators can not only assign limited administrative capabilities to nonroot users, they can also provide the mechanism where a user can carry out a specific function as another user (if required). This is achieved through three features:

- **Authorizations**: User rights that grant access to a restricted function.
- **Execution profiles**: Bundling mechanisms for grouping authorizations and commands with special attributes; for example, user and group IDs or superuser ID.
- **Roles**: Special type of user accounts intended for performing a set of administrative tasks.
Using RBAC

To better describe RBAC, it’s easier to first describe how a system administrator would utilize RBAC to delegate an administrative task to a nonroot user in a fictional setting at Acme Corp.

At Acme Corp., the system administrator is overwhelmed with tasks. He decides to delegate some of his responsibility to Neil, a user from the engineering department who helps out sometimes with system administration tasks.

The system administrator first needs to define which tasks he wants Neil to help with. He has identified three tasks:

- Change user passwords, but do not add or remove accounts.
- Mount and share file systems.
- Shut down the system.

In RBAC, when we speak of delegating administrative tasks, it is referred to as a role account. A role account is a special type of user account that is intended for performing a set of administrative tasks. It is like a normal user account in most respects except that users can gain access to it only through the su command after they have logged in to the system with their normal login account. A role account is not accessible for normal logins, for example, through the CDE login window. From a role account, a user can access commands with special attributes, typically the superuser privilege, which are unavailable to users with normal accounts.

At Acme Corp., the system administrator needs to define a role username for the tasks he wants to delegate. Let’s use the role username “adminusr.” After Neil logs in with his normal login name of ncalkins, he then needs to issue the su command and switch to adminusr whenever he wants to perform administrative tasks. In this chapter, you learn how to create a role account using the command line interface, although you should note that the Solaris Management Console can also be used.

So far we have determined that we want to name the role account adminusr. The system administrator creates the role account using the roleadd command. The roleadd command adds a role account to the /etc/passwd, etc/shadow, and /etc/user_attr files. The syntax for the roleadd command is as follows:
You’ll notice that `roleadd` looks a great deal like the `useradd` command. Table 4.1 describes the options for the `roleadd` command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-c &lt;comment&gt;</code></td>
<td>Any text string to provide a brief description of the role.</td>
</tr>
<tr>
<td><code>-d &lt;dir&gt;</code></td>
<td>The home directory of the new role account.</td>
</tr>
<tr>
<td><code>-m</code></td>
<td>Creates the new role’s home directory if it does not already exist.</td>
</tr>
<tr>
<td><code>-e &lt;expire&gt;</code></td>
<td>Specifies the expiration date for a role. After this date, no user can access this role. The <code>&lt;expire&gt;</code> option argument is a date entered using one of the date formats included in the template file <code>/etc/datemsk</code>. For example, you can enter 10/30/02 or October 30, 2002. A value of &quot; &quot; defeats the status of the expired date.</td>
</tr>
<tr>
<td><code>-f &lt;inactive&gt;</code></td>
<td>Specifies the maximum number of days allowed between uses of a login ID before that login ID is declared invalid. Normal values are positive integers.</td>
</tr>
<tr>
<td><code>-g &lt;group&gt;</code></td>
<td>Specifies an existing group’s integer ID or character-string name. It redefines the role’s primary group membership.</td>
</tr>
<tr>
<td><code>-G &lt;group&gt;</code></td>
<td>Specifies an existing group’s integer ID, or character string name. It redefines the role’s supplementary group membership. Duplicates between groups with the <code>-g</code> and <code>-G</code> options are ignored.</td>
</tr>
<tr>
<td><code>-k &lt;skeldir&gt;</code></td>
<td>A directory that contains skeleton information (such as <code>.profile</code>) that can be copied into a new role’s home directory. This directory must already exist. The system provides the <code>/etc/skel</code> directory that can be used for this purpose.</td>
</tr>
<tr>
<td><code>-s &lt;shell&gt;</code></td>
<td>Specifies the user’s shell on login. The default is <code>/bin/pfsh</code>.</td>
</tr>
<tr>
<td><code>-A &lt;authorization&gt;</code></td>
<td>Both of these options respectively assign authorizations and profiles to the role.</td>
</tr>
<tr>
<td><code>-P &lt;profile&gt;</code></td>
<td>Authorizations and profiles are described later in this section.</td>
</tr>
<tr>
<td><code>-u &lt;uid&gt;</code></td>
<td>Specifies a UID for the new role. It must be a nonnegative decimal integer. The UID associated with the role’s home directory is not modified with this option; a role does not have access to its home directory until the UID is manually reassigned using the <code>chown</code> command.</td>
</tr>
</tbody>
</table>

The other options are the same options that were described for the `useradd` command, outlined in *Solaris 10 System Administration Exam Prep: CX-310-200, Part I*.

When creating a role account with the `roleadd` command, you need to specify an authorization or profile to the role. An *authorization* is a user right that grants access to a restricted function. It is a unique string that identifies what is being authorized as well as who created the authorization.
Certain privileged programs check the authorizations to determine whether users can execute restricted functionality. Following are the predefined authorizations from the `/etc/security/auth_attributes` file that apply to the tasks to be delegated:

```
solaris.admin.usermgr.pswd:::Change Password::help=AuthUserMgrPswd.html
solaris.system.shutdown:::Shutdown the System::help=SysShutdown.html
solaris.admin.fsmgr.write:::Mount and Share File Systems::
help=AuthFsMgrWrite.html
```

All authorizations are stored in the `auth_attr` database, so the system administrator needs to use one or more of the authorizations that are stored in that file. For the Acme Corp. example, the system administrator needs to specify the authorizations shown here:

```
solaris.admin.usermgr.pswd
solaris.system.shutdown
solaris.admin.fsmgr.write
```

The system administrator would therefore issue the `roleadd` command as follows:

```
# roleadd -m -d /export/home/adminusr -c "Admin Assistant" \
-A solaris.admin.usermgr.pswd,solaris.system.shutdown,\
    solaris.admin.fsmgr.write adminusr<cr>
```

A role account named `adminusr` with the required directory structures has been created. The next step is to set the password for the `adminusr` role account by typing the following:

```
passwd adminusr
```

You are prompted to type the new password twice.

Now we need to set up Neil’s account so he can access the new role account named `adminusr`. With the `usermod` command, we assign the role to the user account using the `-R` option:

```
usermod -R adminusr neil
```

### NOTE

**No need to be logged out**  
Previously, you needed to ensure that the user was not logged in at the time of assigning a role; otherwise, you received an error message and the role was not assigned. This is no longer the case. A role can be assigned to a user while the user is still logged in.

To access the administrative functions, Neil needs to first log in using his regular user account named `neil`. Neil can check which roles he has been granted by typing the following at the command line:

```
$ roles<cr>
```

The system responds with the roles that have been granted to the user account `neil`:

`adminusr`
Neil then needs to su to the adminusr account by typing the following:

$ su adminusr<cr>

Neil is prompted to type the password for the role account.

Now Neil can modify user passwords, shut down the system, and mount and share file systems. Any other user trying to su to the adminusr account gets this message:

$ su adminusr<cr>
Password: 
Roles can only be assumed by authorized users
su: Sorry
$

If the system administrator later wants to assign additional authorizations to the role account named adminusr, he would do so using the rolemod command. The rolemod command modifies a role’s login information on the system. The syntax for the rolemod command is as follows:

rolemod [-u uid] [-o] [-g group] [-G group] [-d dir] [-m] [-s shell]
[-P profile] <role account>

Table 4.2 describes options for the rolemod command where they differ from the roleadd command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-A &lt;authorization&gt;</td>
<td>One or more comma-separated authorizations as defined in the auth_attr database. This replaces any existing authorization setting.</td>
</tr>
<tr>
<td>-d &lt;dir&gt;</td>
<td>Specifies the new home directory of the role. It defaults to &lt;base_dir&gt;/&lt;login&gt;, in which &lt;base_dir&gt; is the base directory for new login home directories, and &lt;login&gt; is the new login.</td>
</tr>
<tr>
<td>-l &lt;new_logname&gt;</td>
<td>Specifies the new login name for the role. The &lt;new_logname&gt; argument is a string no more than eight bytes consisting of characters from the set of alphabetic characters, numeric characters, period (.), underline (_), and hyphen (-). The first character should be alphabetic and the field should contain at least one lowercase alphabetic character. A warning message is written if these restrictions are not met. A future Solaris release might refuse to accept login fields that do not meet these requirements. The &lt;new_logname&gt; argument must contain at least one character and must not contain a colon (:) or newline (\n).</td>
</tr>
<tr>
<td>-m</td>
<td>Moves the role’s home directory to the new directory specified with the -d option. If the directory already exists, it must have permissions read/write/execute by group, in which group is the role’s primary group.</td>
</tr>
</tbody>
</table>
Table 4.2 rolemod Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-o</td>
<td>Allows the specified UID to be duplicated (nonunique).</td>
</tr>
<tr>
<td>-P &lt;profile&gt;</td>
<td>Replaces any existing profile setting. One or more comma-separated execution profiles are defined in the auth_attr database.</td>
</tr>
<tr>
<td>-u &lt;uid&gt;</td>
<td>Specifies a new UID for the role. It must be a nonnegative decimal integer. The UID associated with the role’s home directory is not modified with this option; a role does not have access to its home directory until the UID is manually reassigned using the chown command.</td>
</tr>
</tbody>
</table>

To add the ability to purge log files, you need to add `solaris.admin.logsvc.purge` to the list of authorizations for `adminusr`. To do this, issue the `rolemod` command:

```
# rolemod -A solaris.admin.usermgr.pswd,solaris.system.shutdown,\
solaris.admin.fsmgr.write,solaris.admin.logsvc.purge  adminusr<cr>
```

You can verify that the new authorizations have been added to the role by typing the `auths` command at the command line:

```
# auths adminusr<cr>
solaris.admin.usermgr.pswd,solaris.system.shutdown,solaris.admin.fsmgr.\
write,solaris.admin.logsvc.purge,...
```

You can verify that the new authorizations have been added to the role by typing the `auths` command at the command line:

```
# auths adminusr<cr>
solaris.admin.usermgr.pswd,solaris.system.shutdown,solaris.admin.fsmgr.\
write,solaris.admin.logsvc.purge,...
```

[output has been truncated]

**CAUTION**

**rolemod warning** The rolemod command does not add to the existing authorizations; it replaces any existing authorization setting.

If you want to remove a role account, use the `roledel` command:

`roledel [-r] <role account name>`

The `-r` option removes the role’s home directory from the system. For example, to remove the `adminusr` role account, issue the following command:

```
# roledel -r adminusr<cr>
```

The next section discusses each of the RBAC databases in detail, describing the entries made when we executed the `roleadd` and `usermod` commands.
RBAC Components

RBAC relies on the following four databases to provide users access to privileged operations:

- `/etc/user_attr (extended user attributes database)`: Associates users and roles with authorizations and profiles.
- `/etc/security/auth_attr (authorization attributes database)`: Defines authorizations and their attributes and identifies the associated help file.
- `/etc/security/prof_attr (rights profile attributes database)`: Defines profiles, lists the profile’s assigned authorizations, and identifies the associated help file.
- `/etc/security/exec_attr (profile attributes database)`: Defines the privileged operations assigned to a profile.

These four databases are logically interconnected.

**EXAM ALERT**

RBAC database functions You need to be able to correctly identify the function and location of each RBAC database. A common exam question is to match the description with the relevant RBAC database. Remember that the user_attr database resides in the /etc directory and not in the /etc/security directory.

**Extended User Attributes (user_attr) Database**

The `/etc/user_attr` database supplements the passwd and shadow databases. It contains extended user attributes, such as authorizations and profiles. It also allows roles to be assigned to a user. Following is an example of the `/etc/user_attr` database:

```bash
# more /etc/user_attr
# Copyright 2003 by Sun Microsystems, Inc. All rights reserved.
#
#/etc/user_attr
#
# user attributes. see user_attr(4)
#
#pragma ident "@(#)user_attr 1.1 03/07/09 SMI"
#
adm::::profiles=Log Management
lp::::profiles=Printer Management
root::::auths=solaris.*,solaris.grant;profiles=All
adminusr::::type=role;auths=solaris.admin.usermgr.pswd,/
solaris.system.shutdown,solaris.admin.fsmgr.write;profiles=All
neil::::type=normal;roles=adminusr
```
The following fields in the user_attr database are separated by colons:

user:qualifier:res1:res2:attr

Each field is described in Table 4.3.

Table 4.3  user_attr Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Describes the name of the user or role, as specified in the passwd database.</td>
</tr>
<tr>
<td>qualifier</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>res1</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>res2</td>
<td>Reserved for future use.</td>
</tr>
</tbody>
</table>
| attr       | Contains an optional list of semicolon-separated (;) key-value pairs that describe the security attributes to be applied when the user runs commands. Eight valid keys exist: auths, profiles, roles, type, project, defaultpriv, limitpriv, and lock_after_retries:

auths specifies a comma-separated list of authorization names chosen from names defined in the auth_attr database. Authorization names can include the asterisk (*) character as a wildcard. For example, solaris.device.* means all the Solaris device authorizations.

profiles contains an ordered, comma-separated list of profile names chosen from prof_attr. A profile determines which commands a user can execute and with which command attributes. At a minimum, each user in user_attr should have the All profile, which makes all commands available but without attributes. The order of profiles is important; it works similarly to UNIX search paths. The first profile in the list that contains the command to be executed defines which (if any) attributes are to be applied to the command. Profiles are described in the section titled "Authorizations (auth_attr) Database."

roles can be assigned to the user using a comma-separated list of role names. Note that roles are defined in the same user_attr database. They are indicated by setting the type value to role. Roles cannot be assigned to other roles.

type can be set to normal, if this account is for a normal user, or to role, if this account is for a role. A normal user assumes a role after he has logged in.

project can be set to a project from the projects database, so that the user is placed in a default project at login time.

defaultpriv is the list of default privileges the user is assigned.

limitpriv: The system administrator can limit the set of privileges allowed, and this attribute contains the maximum set of privileges the user can be allowed. Care must be taken when limiting privileges so as to not affect other applications the user might execute.
lock_after_retries specifies whether an account is locked out following a number of failed logins. The number of failed logins is taken from the RETRIES option in /etc/default/login. The default is no.

In the previous section, we issued the following roleadd command to add a role named adminusr:

```
# roleadd -m -d /export/home/adminusr -c "Admin Assistant" -A solaris.admin.usermgr.pswd,solaris.system.shutdown,"
```

The roleadd command made the following entry in the user_attr database:

```
adminusr::::type=role;auths=solaris.admin.usermgr.pswd,\solaris.system.shutdown,solaris.admin.fsmgr.write;profiles=All
```

We can then issue the following usermod command to assign the new role to the user neil:

```
# usermod -R useradmin neil<cr>
```

and then make the following entry to the user_attr database:

```
neil::::type=normal;roles=adminusr
```

**Authorizations (auth_attr) Database**

An authorization is a user right that grants access to a restricted function. In the previous section, the system administrator wanted to delegate some of the system administrative tasks to Neil. Assigning authorizations to the role named adminusr did this. An authorization is a unique string that identifies what is being authorized as well as who created the authorization. Remember that we used the following authorizations to give Neil the ability to modify user passwords, shut down the system, and mount and share file systems:

- solaris.admin.usermgr.pswd
- solaris.system.shutdown
- solaris.admin.fsmgr.write

Certain privileged programs check the authorizations to determine whether users can execute restricted functionality. For example, the solaris.jobs.admin authorization is required for one user to edit another user’s crontab file.

All authorizations are stored in the auth_attr database. If no name service is used, the database is located in a file named /etc/security/auth_attr. Authorizations can be assigned directly to users (or roles), in which case they are entered in the user_attr database. Authorizations can also be assigned to profiles, which in turn are assigned to users. They are described in the “Rights Profiles (prof_attr) Database” section, later in this chapter.

The fields in the auth_attr database are separated by colons, as shown here:

```
authname:res1:res2:short_desc:long_desc:attr
```
Each field is described in Table 4.4.

Table 4.4 **auth_attr Fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authname[suffix]</td>
<td>A unique character string used to identify the authorization in the format prefix. Authorizations for the Solaris operating environment use <code>solaris</code> as a prefix. All other authorizations should use a prefix that begins with the reverse-order Internet domain name of the organization that creates the authorization (for example, <code>com.xyzcompany</code>). The suffix indicates what is being authorized—typically, the functional area and operation. When no suffix exists (that is, the authname consists of a prefix and functional area and ends with a period), the authname serves as a heading for use by applications in their GUIs rather than as an authorization. The authname <code>solaris.printmgr</code> is an example of a heading. When the authname ends with the word <code>grant</code>, the authname serves as a grant authorization and allows the user to delegate related authorizations (that is, authorizations with the same prefix and functional area) to other users. The authname <code>solaris.printmgr.grant</code> is an example of a grant authorization; it gives the user the right to delegate such authorizations as <code>solaris.printmgr.admin</code> and <code>solaris.printmgr.nobanner</code> to other users.</td>
</tr>
<tr>
<td>res1</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>res2</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>short_desc</td>
<td>A shortened name for the authorization suitable for displaying in user interfaces, such as in a scrolling list in a GUI.</td>
</tr>
<tr>
<td>long_desc</td>
<td>A long description. This field identifies the purpose of the authorization, the applications in which it is used, and the type of user interested in using it. The long description can be displayed in the help text of an application.</td>
</tr>
<tr>
<td>attr</td>
<td>An optional list of semicolon-separated (;) key-value pairs that describe the attributes of an authorization. Zero or more keys can be specified. The keyword <code>help</code> identifies a help file in HTML. Help files can be accessed from the <code>index.html</code> file in the <code>/usr/lib/help/auths/locale/C</code> directory.</td>
</tr>
</tbody>
</table>

The following are some typical values found in the default `auth_attr` database:

```
solaris.admin.usermgr.pswd:::Change Password::help=AuthUserMgrPswd.html
solaris.system.shutdown:::Shutdown the System::help=SysShutdown.html
solaris.admin.fsmgr.write:::Mount and Share File Systems::
help=AuthFsMgrWrite.html
```

Look at the relationship between the `auth_attr` and the `user_attr` databases for the `adminusr` role we added earlier:

```
adminusr::::type=role;auths=solaris.admin.usermgr.pswd,\
solaris.system.shutdown,solaris.admin.fsmgr.write;profiles=All
```
Notice the authorization entries that are **bold**. These authorization entries came out of the auth_attr database, shown previously. The `solaris.system.shutdown` authorization, which is defined in the auth_attr database, gives the role the right to shut down the system.

**Rights Profiles (prof_attr) Database**

We referred to rights profiles, or simply profiles, earlier in this chapter. Up until now, we assigned authorization rights to the role account. Defining a role account that has several authorizations can be tedious. In this case, it’s better to define a profile, which is several authorizations bundled together under one name called a **profile name**. The definition of the profile is stored in the prof_attr database. Following is an example of a profile named Operator, which is in the default prof_attr database. Again, if you are not using a name service, the prof_attr file is located in the /etc/security directory.

```
Operator:::Can perform simple administrative tasks:profiles=Printer Management,Media Backup,All;help=RtOperator.html
```

Several other profiles are defined in the prof_attr database. Colons separate the fields in the prof_attr database:

```
profname:res1:res2:desc:attr
```

The fields are defined in Table 4.5.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>profname</td>
<td>The name of the profile. Profile names are case-sensitive.</td>
</tr>
<tr>
<td>res1</td>
<td>A field reserved for future use.</td>
</tr>
<tr>
<td>res2</td>
<td>A field reserved for future use.</td>
</tr>
<tr>
<td>desc</td>
<td>A long description. This field should explain the purpose of the profile, including what type of user would be interested in using it. The long description should be suitable for displaying in the help text of an application.</td>
</tr>
<tr>
<td>attr</td>
<td>An optional list of key-value pairs separated by semicolons (;) that describe the security attributes to apply to the object upon execution. Zero or more keys can be specified. The four valid keys are <code>help</code>, <code>auths</code>, <code>privs</code>, and <code>profiles</code>. The keyword <code>help</code> identifies a help file in HTML. Help files can be accessed from the <code>index.html</code> file in the <code>/usr/lib/help/auths/locale/C</code> directory. <code>auths</code> specifies a comma-separated list of authorization names chosen from those names defined in the auth_attr database. Authorization names can be specified using the asterisk (*) character as a wildcard.</td>
</tr>
</tbody>
</table>
Perhaps the system administrator wants to create a new role account and delegate the task of printer management and backups. He could look through the auth_attr file for each authorization and assign each one to the new role account using the roleadd command, as described earlier. Or, he could use the Operator profile currently defined in the prof_attr database, which looks like this:

The Operator profile consists of three other profiles:
- Printer Management
- Media Backup
- All

Let’s look at each of these profiles as defined in the prof_attr database:

Printer Management:::Manage printers, daemons, spooling:help=RtPrntAdmin.html;auths=solaris.admin.printer.read,solaris.admin.printer.modify,
solaris.admin.printer.delete
Media Backup:::Backup files and file systems:help=RtMediaBkup.html
All:::Execute any command as the user or role:help=RtAll.html

Printer Management has the following authorizations assigned to it:
- solaris.admin.printer.read
- solaris.admin.printer.modify
- solaris.admin.printer.delete

When you look at these three authorizations in the auth_attr database, you see the following entries:
- solaris.admin.printer.read:::View Printer Information::help=AuthPrinterRead.html
- solaris.admin.printer.modify:::Update Printer Information::help=AuthPrinterModify.html
- solaris.admin.printer.delete:::Delete Printer Information::help=AuthPrinterDelete.html

Assigning the Printer Management profile is the same as assigning the three authorizations for viewing, updating, and deleting printer information.

The Media Backup profile provides authorization for backing up data, but not restoring data. The Media Backup profile does not have authorizations associated with it like the Printer Management profile has. I’ll describe how this profile is defined in the next section when I describe execution attributes.
The All profile grants the right for a role account to use any command when working in an administrator’s shell. These shells can only execute commands that have been explicitly assigned to a role account through granted rights. We’ll explore this concept further when I describe execution attributes in the next section.

To create a new role account named admin2 specifying the Operator profile, use the roleadd command with the -P option:

```
# roleadd -m -d /export/home/admin2 -c "Admin Assistant" -P Operator admin2
```

The following entry is added to the user_attr database:

```
admin2:::type=role;profiles=Operator
```

At any time, users can check which profiles have been granted to them with the profiles command:

```
$ profiles
```

The system lists the profiles that have been granted to that particular user account.

**Execution Attributes (exec_attr) Database**

An execution attribute associated with a profile is a command (with any special security attributes) that can be run by those users or roles to which the profile is assigned. For example, in the previous section, we looked at the profile named Media Backup in the prof_attr database. Although no authorizations were assigned to this profile, the MediaBackup profile was defined in the exec_attr database as follows:

```
Media Backup:solaris:act:::Tar;*;*;*;*:privs=all
Media Backup:solaris:act:::Tar;*;TAR,MAGTAPE;*;>0:privs=all
Media Backup:solaris:act:::TarList;*;*;*;*:
Media Backup:suser:cmd:::/usr/bin/mt:euid=0
Media Backup:suser:cmd:::/usr/lib/fs/ufs/ufsdump:euid=0;gid=sys
Media Backup:suser:cmd:::/usr/sbin/tar:euid=0
```

The fields in the exec_attr database are as follows and are separated by colons:

```
name:policy:type:res1:res2:id:attr
```

The fields are defined in Table 4.6.
Table 4.6 exec_attr Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the profile. Profile names are case-sensitive.</td>
</tr>
<tr>
<td>policy</td>
<td>The security policy associated with this entry. Currently, suer (the superuser policy model) and solaris are the only valid policy entries. The solaris policy recognizes privileges, whereas the suer policy does not.</td>
</tr>
<tr>
<td>type</td>
<td>The type of entity whose attributes are specified. The two valid types are cmd (command) and act. The cmd type specifies that the ID field is a command that would be executed by a shell. The act type is available only if the system is configured with Trusted Extensions.</td>
</tr>
<tr>
<td>res1</td>
<td>This field is reserved for future use.</td>
</tr>
<tr>
<td>res2</td>
<td>This field is reserved for future use.</td>
</tr>
<tr>
<td>id</td>
<td>A string identifying the entity; the asterisk (*) wildcard can be used. Commands should have the full path or a path with a wildcard. To specify arguments, write a script with the arguments and point the id to the script.</td>
</tr>
<tr>
<td>attr</td>
<td>An optional list of semicolon (;) separated key-value pairs that describe the security attributes to apply to the entity upon execution. Zero or more keys can be specified. The list of valid keywords depends on the policy being enforced. Six valid keys exist: euid, uid, egid, gid, privs, and limitprivs.</td>
</tr>
</tbody>
</table>

- **euid** and **uid** contain a single username or numeric user ID. Commands designated with euid run with the effective UID indicated, which is similar to setting the setuid bit on an executable file. Commands designated with uid run with both the real and effective UIDs.

- **egid** and **gid** contain a single group name or numeric group ID. Commands designated with egid run with the effective GID indicated, which is similar to setting the setgid bit on an executable file. Commands designated with gid run with both the real and effective GIDs.

**NOTE**

**Trusted Solaris** You will see an additional security policy if you are running Trusted Solaris, a special security-enhanced version of the operating environment. The policy tsol is the trusted solaris policy model.

Looking back to the Media Backup profile as defined in the exec_attr database, we see that the following commands have an effective UID of 0 (superuser):

```
/usr/bin/mt
/usr/sbin/tar
/usr/lib/fs/ufs/ufsdump
```
Therefore, any user that has been granted the Media Backup profile can execute the previous backup commands with an effective user ID of 0 (superuser).

In the prof_attr database, we also saw that the Operator profile consisted of a profile named All. Again, All did not have authorizations associated with it. When we look at the exec_attr database for a definition of the All profile, we get the following entry:

```
All:suser:cmd:::*:
```

Examining each field, we see that All is the profile name, the security policy is suser, and the type of entity is cmd. The attribute field has an *.

It’s common to grant all users the All profile. The * is a wildcard entry that matches every command. In other words, the user has access to any command while working in the shell. Without the All profile, a user would have access to the privileged commands, but no access to normal commands such as ls and cd. Notice that no special process attributes are associated with the wildcard, so the effect is that all commands matching the wildcard run with the UID and GID of the current user (or role).

---

**NOTE**

The All profile. Always assign the All profile last in the list of profiles. If it is listed first, no other rights are consulted when you look up command attributes.

---

**syslog**

**Objective**

- Explain syslog function fundamentals and configure and manage the /etc/syslog.conf file and syslog messaging.

A critical part of the system administrator's job is monitoring the system. Solaris uses the syslog message facility to do this. syslogd is the daemon responsible for capturing system messages. The messages can be warnings, alerts, or simply informational messages. As the system administrator, you customize syslog to specify where and how system messages are to be saved.

The syslogd daemon receives messages from applications on the local host or from remote hosts and then directs messages to a specified log file. To each message that syslog captures, it adds a timestamp, the message type keyword at the beginning of the message, and a newline at the end of the message. For example, the following messages were logged in the /var/adm/messages file:

```
July 15 23:06:39 sunfire ufs: [ID 845546 kern.notice] NOTICE: alloc: /var: \file system full
Sep  1 04:57:06 docbert nfs: [ID 563706 kern.notice] NFS server saturn.east ok
```
syslog enables you to capture messages by facility (the part of the system that generated the message) and by level of importance. Facility is considered to be the service area generating the message or error (such as printing, email, or network), whereas the level can be considered the level of severity (such as notice, warning, error, or emergency). syslog also enables you to forward messages to another machine so that all your messages can be logged in one location. The syslogd daemon reads and logs messages into a set of files described by the configuration file /etc/syslog.conf. When the syslogd daemon starts up, it preprocesses the /etc/syslog.conf file through the m4 macro processor to get the correct information for specific log files. syslogd does not read the /etc/syslog.conf file directly. syslogd starts m4, which parses the /etc/syslog.conf file for ifdef statements that can be interpreted by m4. The function ifdef is an integral part of m4 and identifies the system designated as LOGHOST. The macro then can evaluate whether log files are to be held locally or on a remote system, or a combination of both.

If m4 doesn't recognize any m4 commands in the syslog.conf file, output is passed back to syslogd. syslogd then uses this output to route messages to appropriate destinations. When m4 encounters ifdef statements that it can process, the statement is evaluated for a true or false condition and the message is routed relative to the output of the test.

EXAM ALERT
/etc/syslog.conf and ifdef statements Make sure you become familiar with the facilities and values listed in the tables in this section. An exam question might provide a sample file and ask where a specific type of message, such as a failed login, will be logged. Also watch out for the ifdef statements to see if the logging is being carried out on a remote system.

An entry in the /etc/syslog.conf file is composed of two fields:

selector      action

The selector field contains a semicolon-separated list of priority specifications of this form:
facility.level [ ; facility.level ]

The action field indicates where to forward the message. Many defined facilities exist.

EXAM ALERT
Separate with tabs The separator between the two fields must be a tab character. Spaces do not work and give unexpected results. This is a very common mistake.
The facilities are described in Table 4.7.

**Table 4.7 Recognized Values for Facilities**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Messages generated by user processes. This is the default priority for messages from programs or facilities not listed in this file.</td>
</tr>
<tr>
<td>kern</td>
<td>Messages generated by the kernel.</td>
</tr>
<tr>
<td>mail</td>
<td>The mail system.</td>
</tr>
<tr>
<td>daemon</td>
<td>System daemons, such as <code>in.ftpd</code>.</td>
</tr>
<tr>
<td>auth</td>
<td>The authorization system, such as <code>login</code>, <code>su</code>, <code>getty</code>, and others.</td>
</tr>
<tr>
<td>1pr</td>
<td><code>1pr</code> is the <code>syslogd</code> facility responsible for generating messages from the line printer spooling system—<code>lpr</code> and <code>lpc</code>.</td>
</tr>
<tr>
<td>news</td>
<td>Reserved for the Usenet network news system.</td>
</tr>
<tr>
<td>uucp</td>
<td>Reserved for the UUCP system. It does not currently use the <code>syslog</code> mechanism.</td>
</tr>
<tr>
<td>cron</td>
<td>The <code>cron</code>/<code>at</code> facility, such as <code>crontab</code>, <code>at</code>, <code>cron</code>, and others.</td>
</tr>
<tr>
<td>audit</td>
<td>The audit facility, such as <code>auditd</code>.</td>
</tr>
<tr>
<td>local0-7</td>
<td>Reserved for local use.</td>
</tr>
<tr>
<td>mark</td>
<td>For timestamp messages produced internally by <code>syslogd</code>.</td>
</tr>
<tr>
<td>*</td>
<td>Indicates all facilities except the <code>mark</code> facility.</td>
</tr>
</tbody>
</table>

Table 4.8 lists recognized values for the `syslog level` field. They are listed in descending order of severity.

**Table 4.8 Recognized Values for level**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emerg</td>
<td>Panic conditions that would normally be broadcast to all users.</td>
</tr>
<tr>
<td>alert</td>
<td>Conditions that should be corrected immediately, such as a corrupted system database.</td>
</tr>
<tr>
<td>crit</td>
<td>Warnings about critical conditions, such as hard device errors.</td>
</tr>
<tr>
<td>err</td>
<td>Other errors.</td>
</tr>
<tr>
<td>warning</td>
<td>Warning messages.</td>
</tr>
<tr>
<td>Notice</td>
<td>Conditions that are not error conditions but that might require special handling, such as a failed login attempt. A failed login attempt is considered a notice and not an error.</td>
</tr>
<tr>
<td>info</td>
<td>Informational messages.</td>
</tr>
<tr>
<td>debug</td>
<td>Messages that are normally used only when debugging a program.</td>
</tr>
<tr>
<td>none</td>
<td>Does not send messages from the indicated facility to the selected file. For example, the entry <code>*.debug;mail.none</code> in <code>/etc/syslog.conf</code> sends all messages except <code>mail</code> messages to the selected file.</td>
</tr>
</tbody>
</table>
Values for the action field can have one of four forms:

- A filename, beginning with a leading slash. This indicates that messages specified by the selector are to be written to the specified file. The file is opened in append mode and must already exist. `syslog` does not create the file if it doesn’t already exist.

- The name of a remote host, prefixed with a @. An example is `@server`, which indicates that messages specified by the selector are to be forwarded to `syslogd` on the named host. The hostname `loghost` is the hostname given to the machine that will log `syslogd` messages. Every machine is its own `loghost` by default. This is specified in the local `/etc/hosts` file. It is also possible to specify one machine on a network to be `loghost` by making the appropriate host table entries. If the local machine is designated as `loghost`, `syslogd` messages are written to the appropriate files. Otherwise, they are sent to the machine `loghost` on the network.

- A comma-separated list of usernames, which indicates that messages specified by the selector are to be written to the named users if they are logged in.

- An asterisk, which indicates that messages specified by the selector are to be written to all logged-in users.

Blank lines are ignored. Lines in which the first nonwhitespace character is a # are treated as comments.

All of this becomes much clearer when you look at sample entries from an `/etc/syslog.conf` file:

```
*.err    /dev/console
*.err;daemon,auth.notice;mail.crit    /var/adm/messages
mail.debug    /var/log/syslog
*.alert    root
*.emerg    *
kern.err    @server
*.alert;auth.warning    /var/log/auth
```

In this example, the first line prints all errors on the console.

The second line sends all errors, daemon and authentication system notices, and critical errors from the mail system to the file `/var/adm/messages`.  

"Levels include all higher levels too" When you specify a syslog level, it means that the specified level and all higher levels. For example, if you specify the `err` level, this includes `crit`, `alert`, and `emerg` levels as well.
The third line sends mail system debug messages to `/var/log/syslog`.

The fourth line sends all alert messages to user root.

The fifth line sends all emergency messages to all users.

The sixth line forwards kernel messages of `err` (error) severity or higher to the machine named `server`.

The last line logs all alert messages and messages of `warning` level or higher from the authorization system to the file `/var/log/auth`.

The level `none` may be used to disable a facility. This is usually done in the context of eliminating messages. For example:

```
*.debug;mail.none /var/adm/messages
```

This selects debug messages and above from all facilities except those from mail. In other words, mail messages are disabled. The mail system, sendmail, logs a number of messages. The mail system can produce a large amount of information, so some system administrators disable mail messages or send them to another file that they clean out frequently. Before disabling mail messages, however, remember that sendmail messages come in very handy when you’re diagnosing mail problems or tracking mail forgeries.

As of Solaris 10, the mechanism for stopping, starting, and refreshing `syslogd` has changed. The `syslog` function is now under the control of the Service Management Facility (SMF), which is described in detail in the book *Solaris 10 System Administration Exam Prep: CX-310-200, Part I*.

To stop or start `syslogd`, use the `svcadm` command with the appropriate parameter, `enable` or `disable`:

```
# svcadm enable -t system-log<cr>
# svcadm disable -t system-log<cr>
```

The `syslog` facility reads its configuration information from `/etc/syslog.conf` whenever it receives a refresh command from the service administration command, `svcadm`, and when the system is booted. You can make your changes to `/etc/syslog.conf` and then run the following command to cause the file to be reread by the `syslogd` daemon:

```
# svcadm refresh system-log<cr>
```

**EXAM ALERT**

No more `kill -HUP`. Make sure you remember that the `kill -HUP` facility should no longer be used to try to cause a daemon process to re-read its configuration file, even though it still works. The `svcadm refresh` command is now the recommended way of achieving this.
The first message in the log file is logged by the syslog daemon itself to show when the process was started.

syslog logs are automatically rotated on a regular basis. In previous Solaris releases, this was achieved by the program newsyslog. A new method of log rotation was introduced with Solaris 9—logadm, a program normally run as a root-owned cron job. A configuration file /etc/logadm.conf is now used to manage log rotation and allows a number of criteria to be specified. See the logadm and logadm.conf manual pages for further details.

**Using the logger Command**

The logger command provides the means of manually adding one-line entries to the system logs from the command line. This is especially useful in shell scripts.

The syntax for the logger command is as follows:

```
logger [-i] [-f file] [-p priority] [-t tag] [message] ...
```

Options to the logger command are described in Table 4.9.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-i</td>
<td>Logs the Process ID (PID) of the logger process with each line written to a log file.</td>
</tr>
<tr>
<td>-f &lt;file&gt;</td>
<td>Use the contents of file as the message to be logged.</td>
</tr>
<tr>
<td>-p &lt;priority&gt;</td>
<td>The message priority. This can be defined as a numeric value or as a facility.level pair, as described in Tables 4.7 and 4.8. The default priority is user.notice.</td>
</tr>
<tr>
<td>-t &lt;tag&gt;</td>
<td>Marks each line with the specified tag.</td>
</tr>
<tr>
<td>message</td>
<td>One or more string arguments, separated by a single space character comprising the text of the message to be logged.</td>
</tr>
</tbody>
</table>

For example, perhaps you have a simple shell script that backs up files:

```
#!/bin/ksh
tar cvf /tmp/backup .
logger -p user.alert "Backups Completed"
```

The last line of the script uses the logger command to send a "Backups Completed" message to the default system log (/var/adm/messages). After running the script, I see the following message appended to the log file:

```
Jan 23 14:02:52 sunfire root: [ID 702911 user.alert] Backups Completed
```
Summary

In this chapter you learned about Role-Based Access Control (RBAC), which allows the system administrator to delegate administrative responsibilities to users without having to divulge the root password. A number of profiles allow privileges to be grouped together so that a user can easily be granted a restricted set of additional privileges. Four main RBAC databases interact with each other to provide users with access to privileged operations:

- /etc/security/auth_attr: Defines authorizations and their attributes and identifies the associated help file.
- /etc/security/exec_attr: Defines the privileged operations assigned to a profile.
- /etc/security/prof_attr: Defines the profiles, lists the profile’s assigned authorizations, and identifies the associated help file.
- /etc/user_attr: Associates users and roles with authorizations and execution profiles.

Also in this chapter, you learned about the system logging facility (syslog) and the configuration that facilitates routing of system messages according to specific criteria, as well as determining where the messages are logged. The logger command was covered, which allows the system administrator to enter ad-hoc messages into the system log files.

Key Terms

- Authorization
- Execution profile
- logger
- RBAC
- RBAC databases (know about all four)
- Rights profile
- Role
- syslog
- svcadm command
Apply Your Knowledge

Exercise

4.1 Creating a User and a Role

In this exercise, you'll create a new role named admin1 and a profile called Shutdown. The Shutdown profile will be added to the role. A user account trng1 will be created and have the admin1 role assigned to it. The user will then assume the role and execute a privileged command to shut down the system.

**Estimated time:** 20 minutes

To create a user and a role, follow these steps:

1. Create the role named admin1:
   
   ```
   # roleadd -u 2000 -g 10 -d /export/home/admin1 -m admin1<cr>
   # passwd admin1<cr>
   ```

   You are prompted to enter the password twice.

2. Create a profile to allow the user to shut down a system.

   Edit the `/etc/security/prof_attr` file and enter the following line:
   
   ```
   Shutdown:::Permit system shutdown:
   ```

   Save and exit the file.

3. Add the Shutdown and All profiles to the role:
   
   ```
   # rolemod -P Shutdown,All admin1<cr>
   ```

4. Verify that the changes have been made to the user_attr database:
   
   ```
   # more /etc/user_attr<cr>
   ```

5. Create the user account and assign it access to the admin1 role:
   
   ```
   # useradd -u 3000 -g 10 -d /export/home/trng1 -m -s /bin/ksh -R admin1 trng1<cr>
   ```

6. Assign a password to the new user account:
   
   ```
   # passwd trng1<cr>
   ```

   You are prompted to enter the password twice.

7. Verify that the entry has been made to the passwd, shadow, and user_attr files:
   
   ```
   # more /etc/passwd<cr>
   # more /etc/shadow<cr>
   # more /etc/user_attr<cr>
   ```
8. Assign commands to the Shutdown profile:

   Edit the /etc/security/exec_attr file and add the following line:
   
   `Shutdown:suser:cmd:::/usr/sbin/shutdown:uid=0`

   Save and exit the file.

9. Test the new role and user account as follows:

   a. Log in as trng1.

   b. List the roles that are granted to you by typing the following:
      
      `$ roles<cr>`

   c. Use the su command to assume the role admin1:
      
      `$ su admin1<cr>`

      You are prompted to enter the password for the role.

   d. List the profiles that are granted to you by typing the following:
      
      `$ profiles<cr>`

   e. Shut down the system:
      
      `$/usr/sbin/shutdown -i 0 -g 0<cr>`

### Exam Questions

1. Which of the following commands is used to create a role?

   - A. useradd
   - B. makerole
   - C. roleadd
   - D. addrole

2. In Role-Based Access Control, which file contains details of the user attributes?

   - A. /etc/security/prof_attr
   - B. /etc/user_attr
   - C. /etc/security/user_attr
   - D. /etc/shadow
3. Which two statements about the \texttt{roleadd} command are true? (Choose two.)

- A. \texttt{roleadd} looks similar to the \texttt{useradd} command.
- B. \texttt{roleadd} uses the profile shell (\texttt{pfsh}) as the default shell.
- C. The -A option associates an account with a profile.
- D. An account created with \texttt{roleadd} is the same as a normal login account.

4. Which component of RBAC associates users and roles with authorizations and profiles?

- A. \texttt{user_attr}
- B. \texttt{prof_attr}
- C. \texttt{auth_attr}
- D. \texttt{exec_attr}

5. Which component of RBAC defines the privileged operations assigned to a profile?

- A. \texttt{user_attr}
- B. \texttt{prof_attr}
- C. \texttt{auth_attr}
- D. \texttt{exec_attr}

6. In the execution attributes database, which of the following is not a valid value for the \texttt{attr} field?

- A. \texttt{euid}
- B. \texttt{uid}
- C. \texttt{egid}
- D. \texttt{suid}

7. After creating an RBAC role, you find that the only commands that can be executed within the role are the privileged commands that you have set up. Ordinary nonprivileged commands are not available. The RBAC setup has a problem. What is the cause of this problem?

- A. The role is not associated with a correct profile.
- B. The access mechanism to the role is not initializing properly.
- C. The role's profile is not associated with the correct commands.
- D. The file identifying the privileged commands has missing entries.
- E. The role's profile is not associated with the correct authorizations.
8. Which of the following are valid RBAC databases? (Choose three.)
   ○ A. /etc/usr_attr
   ○ B. /etc/user_attr
   ○ C. /etc/security/exec_attr
   ○ D. /etc/security/prof_attr

9. You want to enable a user to administer all user crontab tables. This includes amending entries in any user's crontab. Given due care to system security, what should you do to enable the user to carry out this duty?
   ○ A. Give the user the root password.
   ○ B. Set the suid on the crontab command.
   ○ C. Use RBAC to authorize the user to administer crontab tables.
   ○ D. Use RBAC to give the user an ID of root when executing the crontab command.
   ○ E. Use the ACL mechanism to give the user RW access to each crontab table.

10. Which command(s) grant a user access to a role account? (Choose two.)
    ○ A. roleadd
    ○ B. rolemod
    ○ C. useradd
    ○ D. usermod

11. Which option to the rolemod command appends an authorization to an exiting list of authorizations?
    ○ A. -A
    ○ B. -P
    ○ C. -a
    ○ D. -o
    ○ E. None

12. In which files are profiles defined? Choose all that apply. (Choose two.)
    ○ A. /etc/security/prof_attr
    ○ B. /etc/user_attr
    ○ C. /etc/security/exec_attr
    ○ D. /etc/security/auth_attr
13. Which statements are true regarding the following line? (Choose all that apply.)

```
Media Restore: suser: cmd:: /usr/lib/fs/ufs/ufsrestore: euid=0
```

- A. It represents a profile in the exec_attr database.
- B. Any role that has Media Restore as a profile can execute the ufsrestore command with an effective UID of root.
- C. It represents a profile in the prof_attr database.
- D. It represents a role definition in the user_attr database.

14. In RBAC, which of the following is a bundling mechanism for grouping authorizations and commands with special attributes?

- A. Profile
- B. Role
- C. Authorization
- D. Group

**Answers to Exam Questions**

1. C. Use the `roleadd` command to create a role account. For more information, see the “Using RBAC” section.

2. B. `/etc/user_attr` contains details of the extended user attributes. For more information, see the “RBAC Components” section.

3. A, B. The `roleadd` command looks very similar to the `useradd` command, but it uses the profile shell as the default shell. For more information, see the “Using RBAC” section.

4. A. `user_attr` (extended user attributes database) associates users and roles with authorizations and profiles. For more information, see the “RBAC Components” section.

5. D. `exec_attr` (profile attributes database) defines the privileged operations assigned to a profile. For more information, see the “RBAC Components” section.

6. D. Six valid keys exist: `euid`, `uid`, `egid`, `gid`, `privs`, and `limitprivs`. For more information, see the “RBAC Components” section.

7. A. If a role is not associated with a correct profile, the only commands that can be executed within the role are the privileged commands that you have set up. Ordinary nonprivileged commands are unavailable. For more information, see the “RBAC Components” section.
8. B, C, D. The three valid RBAC databases are /etc/user_attr, /etc/security/exec_attr, and /etc/security/prof_attr. For more information, see the “RBAC Components” section.

9. C. To enable a user to administer all user cron tables, configure RBAC to authorize the user to administer cron tables. For more information, see the “Using RBAC” section.

10. C, D. Use the roleadd command to create a role account. Then, with the usermod command, assign the role to an existing user account using the -R option. If you are creating a new user account, use the useradd command with the -R option to assign the role to the new user account. For more information, see the “Using RBAC” section.

11. E. The rolemod command does not add to the existing authorizations; it replaces any existing authorization setting. For more information, see the “Using RBAC” section.

12. A, C. /etc/security/prof_attr (rights profile attributes database) defines profiles, lists the profile’s assigned authorizations, and identifies the associated help file. /etc/security/exec_attr (profile attributes database) defines the privileged operations assigned to a profile. For more information, see the “RBAC Components” section.

13. A, B. The following entry in the exec_attr database represents a profile named Media Restore:

```
Media Restore:suser:cmd:::/usr/lib/fs/ufs/ufsrestore:euid=0
```

Any role that has Media Restore as a profile can execute the ufsrestore command with an effective UID of root. For more information, see the “RBAC Components” section.

14. A. Execution profiles are bundling mechanisms for grouping authorizations and commands with special attributes. For more information, see the “RBAC Components” section.

**Suggested Reading and Resources**


Index

A

action field (syslog), 206
activating, new boot environments, 450-452
active study strategies, 10
add install client, options, 372
adding
    devices to ZFS storage pools, 488-489
    patches on OS installed boot environments, 459
    software packages to boot environments, 457-458
    ZFS datasets to non-global zones, 519-521
addresses, displaying MAC addresses, 23
administration, ZFS, 474
all profiles, 203
answers to practice exam, 583-590
archive_location keyword (class files), 344-346
attributes, /etc/nscd.conf, 258-259
auth_attr database, 198
authentication (NIS+), 249
authorization
    NIS+, 250-251
    roleadd, 191-192
authorization (auth_attr) database, RBAC, 197
auths command, 194
autofs, 547
AutoFS, 97. See also automount command
    exam question answers, 116-119
    exam questions, 109-114
    exercises, 108-109
    maps
        direct maps, 89-93, 97
        indirect maps, 93-97
        master maps, 85-89
        naming, 89
        overview, 85
        overview, 83-85
automount command. See also AutoFS
    overview, 82-85
    when to use, 97
automountd command, 83
B
backing up, zones, 304
backup_media keyword (class files), 346-347
begin scripts, JumpStart, 342-343
binding problems (NIS), 247
block devices
Solaris Volume Manager, 139
Zpools, 473
boot environments, maintaining with Solaris Live Upgrade, 456
adding
patches on OS installed boot environments, 459
software packages, 457-458
changing
descriptions of, 460-461
names, 460
deleting inactive boot environments, 459-460
removing
patches on OS installed boot environments, 458
software packages, 456-457
viewing configuration of, 461
boot servers, JumpStart, 324-329
/etc/bootparams, 327
/etc/dfs/dfstab, 327
/etc/ethers, 326
/etc/hosts, 326
/tftpboot, 327
booting
WAN boot client, 431
from local CD/DVD, 431-434
from the OBP interactively, 434-436
from the OBP non-interactively, 436
with DHCP servers, 436-437
zones, 289-291
booting x86 clients, 402
bootlog-cgi, 420
boot_createbe keyword (class files), 348
boot_device keyword (class files), 347
child directories, 249
CIDR (classless inter domain routing), description of, 540
class A networks, 539
class B networks, 539
class C networks, 539
class D networks, 539
class files (JumpStart)
archive_location, 344-346
backup_media, 346-347
boot_createbe, 348
boot_device, 347
client_arch, 348-349
client_root, 349
client_swap, 349-350
cluster, 350
dontuse, 351, 363
filesys, 351-354
forced_deployment, 354
geo, 354-355
install_type, 354
layout_constraint, 355-356
locale, 355-357
local_customization, 357
metadb, 357-358
no_content_check, 358
no_master_check, 358
num_clients, 358
overview, 343-344
package, 358-359
partitioning, 360
patch, 361-362
pool, 360-361
root_device, 362
system_type, 362
testing class files, 363-365
usedisk, 351, 363
client boot problems, troubleshooting JumpStart, 376
client-side failovers, 78
client/server model, 20
hosts, 20-21
IPv4, 21-22
clients, 20
DNS, 252-254
JumpStart
sample installation, 379-381
setting up, 372-376
LDAP (Lightweight Directory Access Protocol), 256-257
NFS, 69-70
devices

NIS, 243-244
WAN boot, 563
WAN boot clients, booting, 431
with DHCP servers, 436-437
from local CD/DVD, 431-434
interactively from OBP, 434-436
noninteractively from OBP, 436
client_arch keyword (class files), 348-349
client_root keyword (class files), 349
client_swap keyword (class files), 349-350
clones, ZFS, 512-513, 552
destroying, 513
replacing ZFS file systems, 513-514
cloning zones, 302-304
Cluster keyword (class files), 350
Clustered environments, 133
commands
metastat, 137
running in zones, 296
common sense study strategies, 11
components
of WAN boot, 420-421, 562
of ZFS, 481-482
disks in storage pools, 482
files in storage pools, 482-483
concatenated stripes, 134, 548
concatenated volumes, creating, 146-147
concatenations, 126-127, 548
SVM, 133-134
configuration diskettes, JumpStart, 332-333
configuration servers, JumpStart, 331-332
configuration diskettes, 332-333
sample installation, 378-379
configuring
JumpStart files, 423-428
SVM, 136
WAN boot files, 423-428
WAN boot servers, 422-423
consolidation (containers), overview, 276
containers
consolidation, 276
resource management, 275-276
versus zones, 275. See also zones
copy-on-write semantics, ZFS, 473-474
core dumps, 542. See also virtual memory
core files
definition of, 542
overview, 63-66
coreadm command, 63, 66, 542-543
-d and –e flag options, 65
options, 64
patterns, 543
coreadm patterns, 64
crash dumps, 543-544
configuring, 66, 68
exam question answers, 116-119
exam questions, 109-114
swap spaces for, 58
critical file systems, Solaris Live Upgrade, 439
custom installations. See JumpStart
CX-310-202 exam, i
objectives reference, i-ii
CX-310-203 exam, ii

D
daemons. See individual daemon names
data, destroying in ZFS, 479
data mirroring, 128
data striping, 134-136
concatenated stripes, 134
data striping (RAID 0), 126-127
databases
state databases (SVM)
creating, 141-143
monitoring, 143-144
recovering from problems, 144-146
state databases, SVM, 133
datasets, ZFS, 552
adding to non-global zones, 519-521
deleagting to non-global zones, 521-522
deleagting ZFS datasets to non-global zones, 521-522
deleting
inactive boot environments, 459-460
zones, 292
dependent keywords, 367
descriptions, changing in boot environments, 460-461
destroying
ZFS clones, 513
ZFS data, 479
ZFS snapshots, 510
detaching devices from mirrored pools, 491-492
devices
adding to ZFS storage pools, 488-489

How can we make this index more useful? Email us at indexes@quepublishing.com
dfmounts command

detaching from mirrored pools, 491-492
replacing in storage pools (ZFS), 515-517
storage pools, 489-490
taking offline and online, 492-493

dfmounts command, 78
dfshares command, 74

DHCP
  PXE (Preboot Execution Environment), 392
  configuring the server, 393-401
  preparing for, 393
  PXE clients, 392
  X86 clients, 401

DHCP servers
  booting WAN boot clients, 436-437
  configuring for SPARC-based clients, 323

differential Flash Archives, creating, 390

direct map fields, 90

direct maps, 89-93, 97
  naming, 89

disk reads, 134-135

disk scrubbing, ZFS, 514

disk sets, 133

disk storage, SVM volumes, 124, 133. See also SVM

disk writes, 134-135

disks in ZFS storage pools, 482

displaying
  MAC addresses, 23
  network information with snoop, 42
  ZFS storage pool information, 484-488

DNS, 254. See also NIS
  clients, configuring, 252-254
  name service exception, 220
  overview, 251, 558

domain name keywords (JumpStart), 367-369

domains, NIS (planning), 233-234

don't use keyword (class files), 351, 363

drivers, Metadisk driver, 132, 139

dry run installations, 363

dumpadm command, 66-68, 544
  options, 544

duration values, 83

errors
  NFS: service not responding error, 80
  No such file or directory error, 81
  RPC: Program not registered error, 80
  RPC: Unknown host error, 81
  Server not responding error, 81

etc files, 556
  overview, 226

etc/auth_attr databases, 197-199

etc/auto master, 85-86

etc/boottabrams, JumpStart, 327

etc/defaultdomain, 30

etc/defaultrouter, 30

etc/dfs/dfstab, 71-74
  JumpStart, 327

etc/ethers, JumpStart, 326

etc/exec_attr databases, 201-203

etc/hostname.*interface* files, 26

etc/hostname.interface, 540

etc/hosttab, JumpStart, 326

etc/inet/hosttab, 27, 540-541

etc/inet/ipnodes, 31, 541

etc/inet/netmasks, 29-31

etc/inetd.conf file, 31

etc/mnttab, 87-89

etc/nscd.conf, attributes, 258-259

etc/nsswitch.conf, 223-225, 557

etc/prof_attr databases, 199-201

etc/services files, 34

etc/syslog.conf, 204

etc/user_attr databases, 195-197

etc/vfstab, swap spaces, 60

exam question answers, 116-119
  crash dumps, 116-119
  Flash Archives, 412-413
  JumpStart, 412-413
  name services, 269-270
  networks, 46
  NFS, 116-119
  PXE (Preboot Execution Environment), 412-413
  RBAC (Role-Based Access Control), 214-215
  SVM, 184
  swap spaces, 116-119
  syslog, 214-215
  zones, 312-313

exam questions
  autoFS, 109-114
  crash dumps, 109-114

e flags
  coreadm, 65
  r option, 194
Flash Archives, 407, 411
JumpStart, 407, 411
name services, 264-268
networks, 43-44
NFS, 109-114
PXE (Preboot Execution Environment), 407-411
RBAC (Role-Based Access Control), 211-213
SVM, 181-182
swap spaces, 109-114
syslog, 211-213
WAN boot, 463-466
   answers to, 466-467
for ZFS, 525-529, 531-533
zones, 308-310
exams
   advice for taking, 6-7
   CX-310-202, objectives reference, i-ii
   CX-310-203, ii
   practice exams, 565-581
   preparing for, 11-12
exec attr database, 201
execution attributes database, RBAC, 201
exercises
   autoFS, 108-109
   JumpStart, 404-407
   name services, 262-264
   networks, 41-42
   NFS, 106-108
   RBAC, 210-211
   SVM, 180
   swap spaces, 105-106
   for ZFS, 524-525
files
   wanboot.conf file, 428-430
   in ZFS storage pools, 482-483
filesystems mirror options, 353
finish scripts, JumpStart, 343
flags, Coreadm, 65
flar command, 560
flarcreate command, 383-386, 390, 560
Flash Archives, 560-561. See also Solaris Flash
   creating, 383, 385-386
   differential Flash Archives, 390
   exam question answers, 412-413
   exam questions, 407, 411
   installing with Solaris installation, 387-389
   JumpStart, 391-392
   overview, 382
forced_deployment keyword (class files), 354
G
gcore command, 544
geo keyword (class files), 354-355
getent command, 260
GIDs, resolving duplicates, 238
global zones, 558
   description of, 277. See also zones
   features of, 279
group files, creating, 238
GRUB menu, Alternate boot environments, 454
H
ahalting zones, 290
hardware, networks, 21
hardware addresses, 21
hardware requirements, for ZFS, 475
hierarchical namespaces, NIS+, 249
history of ZFS, 494
host files, creating, 239
hostnames, 21
   changing, 541
hosts
   client/server model, 20-21
   names, changing, 29-30
   overview, 20-21
hosts databases, 27
hot spare pools, 133, 137
hot spares, 133
ifconfig utility. See also network interfaces
configuring, 26
ifdef statements, 204. See also syslog
indirect maps, 93-97
inetcadm command, 31-33
inetcnv command, 31
initial zone login, 293
install servers (JumpStart), 329-331
sample installation, 376-377
installation setup (JumpStart), troubleshooting, 375
installations, JumpStart. See JumpStart
installing
Solaris Live Upgrade, 438-439
zones, 289
install_type keyword (class files), 354
interfaces (network), configuring, 540-541
interfaces, 134
internet addresses, 21
iostat command, 137
IP addresses, 22
ipnodes databases, 541
IPv4, client/server model, 21-22
IPv4 addresses
cautions, 22
overview, 21
planning for, 22
IPv4 network interfaces. See also network interfaces
configuring, 26-28
/etc/hostname.*interface* file, 26
/etc/inet/hosts file, 27-28
/lib/svc/method/net-physical file, 26

journaling process, ZFS file system, 473

JumpStart
begin scripts, 342-343
boot servers, 324-329
/etc/bootparams, 327
/etc/dfs/dfstab, 327
/etc/ethers, 326
/etc/hosts, 326
/tftpboot, 327
class files
archive_location, 344, 346
backup_media, 346-347
boot_createbe, 348
boot_device, 347
client_arch, 348-349
client_root, 349
client_swap, 349-350
cluster, 350
don'tuse, 351, 363
filesys, 351-354
forced_deployment, 354
geo, 354-355
install_type, 354
layout_constraint, 355-356
locale, 355-357
local_customization, 357
metadb, 357-358
no_content_check, 358
no_master_check, 358
num_clients, 358
overview, 343-344
package, 358-359
partitioning, 360
patch, 361-362
pool, 360-361
root_device, 362
system_type, 362
testing class files, 363-365
usedisk, 351, 363
clients, setting up, 372-376
commands, list of, 319
components of, 319-320, 559-560
configuration servers, 331-332
configuration diskettes, 332-333
custom installation process, 321
/etc/bootparams, 327
exam question answers, 412-413
exam questions, 407, 411
exercises, 404-407
files, configuring, 423-428
finish scripts, 343
Flash Archives, 391-392
install servers, 329-331
name service environments, 372
overview, 318-319
preparing for, 320-321
rules files
check script, 341-342
matches, 340
overview, 333-334, 336
requirements of, 337
syntax of, 336-337
rules keywords, 338-340
rules values, 338-340
sample installation
  clients, setting up, 379-381
  configuration servers, setting up, 378-379
  install servers, setting up, 376-377
  JumpStart directories, creating, 377-378
SPARC, 322
sysidcfg files, 366-367
  name service keywords, 367, 369
  network keywords, 369-370
  root passwords, setting, 370
  time servers, setting, 371
/tftpboot, 327
troubleshooting, 375-376
  client boot problems, 376
  installation setup, 375
X86/x64 systems, 323-324

K

keywords, dependent keywords, 367
Kill –HUP, 207

L

layers, network layers, 537-538
layout_constraint keyword (class files), 355-356
LDAP (Lightweight Directory Access Protocol)
  clients, 256-257
  listing client properties, 257
  modifying clients, 257
  overview, 254-255, 558. See also DNS
  Sun Java System Directory Server, 255-256
  uninitializing clients, 257
learning processes, 9
legacy mount points, ZFS file systems, 502-504
level field (syslog), 205
levels of RAID, 125
lib/svc/method/net-physical files, 26
listing LDAP client properties, 257
live Upgrade Patch, 438
locale keyword (class files), 355-357
local_customization keyword (class files), 357
lockfs command, 164, 171
logger command, 208
logical driver. See metadisk driver
logins, zones. See zlogin
lu command, 440
luactivate, 440
  activating new boot environments, 450-452
  SPARC platforms, 455
  X86/x64 platforms, 452-454
lucancel, 440
lucompare, 440
lucreate, 440
creating new boot environments, 441-445
  in another root pool, 445-446
lucurr, 440
ludelete, 440
ludesc, 440, 461
luftlist, 440
lumake, 440
lumount, 440
lunicode, 440
lustatus command, 447
luupgrade, 440
  upgrading new boot environments, 447-449

M

MAC addresses
  displaying, 23
  monitoring, 22
macro/micro study strategies, 10
maintaining boot environments (Solaris Live Upgrade), 456
  adding
    patches on OS installed boot environments, 459
    software packages, 457-458
  changing
    descriptions of, 460-461
    names, 460
  deleting inactive boot environments, 459-460
  removing
    patches on OS installed boot environments, 458
    software packages, 456-457
  viewing configuration of, 461
makefiles, preparing, 241
management commands, NIS, 235
mapping, description of, 251
maps. See also NIS
  AutoFS
    direct maps, 89-93, 97
    indirect maps, 93-97
master files, creating

master maps, 85-89
overview, 85
naming, 89
NIS, 229, 231-233
creating custom maps, 245-246
passwd maps, 246
where to generate, 230

master files, creating, 240
master group files, creating, 238
master host files, creating, 239
master maps, 85-89
master passwd files, creating, 236-237
master servers
NIS
configuring, 234-236
ypinit, 241-242
starting/stopping NIS, 242-243

menu.lst file, 454
metaclear command, options, 147
metadb command, 137, 357, 549-550
options, 142
state databases, creating, 142
metadb keyword (class files), 357-358
Metadisk driver, 132, 139
Metainit command, 146-147
options, 146-147
for creating soft partitions, 151
mirrors, creating, 156-157
metastat command, 137-138, 149-150
options, 150
migrating zones, 300-302
mirrored storage pools, 483
converting from nonredundant pools, 490-491
detaching devices, 491-492
replacing disks in, 516-517

mirroring, 128
root file systems, 162-166, 169
on x86-based systems, 166-172
unmirroring, 173-174
unmirroring systems, 159-160

mirrors, 134-135, 548
creating, 156-159
root file systems, troubleshooting, 174-176
submirrors, placing offline, 160-162
ZFS, 552

mkfile command, swap spaces, 60-62

modifying
existing zones, 299-300
LDAP clients, 257
monitoring swap resources, 55-57
mount command, NFS, 74-78
mounting ZFS file systems, 500-502
legacy mount points, 502-504

moving zones, 300

name Service Cache Daemon. See nscd
name service keywords (JumpStart), 367-369
name services. See also LDAP
DNS exception, 220
exam question answers, 269-270
exam questions, 264-268
exercises, 262-264
getent command, 260
JumpStart, setting up, 372
overview, 220-221
source status codes, 225
sources, 225
switch files, 222-226
Switch template files, 557

name-to-address resolutions. See mapping

names, changing in boot environments, 460
naming
ZFS file system, 478
ZFS snapshots, 510

naming services
DNS, 558
/etc files, 556
LDAP, 558
NIS, 556-557
NIS+, 557-558
overview, 555-556

native read-only ZFS properties, 495-496
netstat command, 37-38

network File System, 52
network hardware, overview, 21

network information, displaying with snoop, 42

network Interface Card (NIC), 538

network interfaces, 22
configuring, 28, 540-541
/etc/hostname.[lt]interface[gt] files, 26
/etc/inet/host files, 27
/etc/inet/netmasks files, 29
network keywords (JumpStart), 369-370
network layers, list of, 537-538
network maintenance
overview, 36-39
verifying operation of, 37-39
network services
overview, 31-34
RPC services, 34-36
networking zones, 281
networks
CIDR, 540
class A networks, 539
class B networks, 539
class C networks, 539
class D networks, 539
components of, 538
exam question answers, 46
exam questions, 43-44
exercises, 41-42
NFS services, 545-546
NFS (Network File System), 69
AutoFS, 547
overview, 82-85
automount command. See automount command
clients/servers, 69
daemons, 70
exam question answers, 116-119
exam questions, 109-114
exercises, 106-108
mount command, 75
overview, 68-69
remote file systems, mounting, 74-78
server logging, 78-79
servers and clients, 70
setting up, 71-74
swap spaces, 62
troubleshooting errors, 80
NFS server not responding, still trying message, 81
NFS: service not responding error, 80
No such file or directory error, 81
RPC: Program not registered error, 80
RPC: Unknown host error, 81
Server not responding error, 81
NFS daemons, list of, 546
NFS server not responding, still trying message, 81
NFS services, 545-546
NFS: service not responding error, 80
NIC (Network Interface Card), 538
NIS (Network Information Service)
binding problems, 247
clients, setting up, 243-244
daemons, 234
determining hosts as servers, 229
determining servers needed, 228-229
domains, planning, 233-234
makefiles, preparing, 241
maps. See maps
master files, creating, 240
master group files, creating, 238
master host files, creating, 239
master passwd files, creating, 236-237
master servers, configuring, 234-236
overview, 227, 556-557
security, 246-247
server problems, 248
slave servers, 244-245
SMF, 243
starting/stopping, 242-243
structure of, 227-228
Ypinit, 241-242
NIS commands, 235
NIS daemons, 234
NIS+
authorization, 250-251
hierarchical namespaces, 249
overview, 248, 557-558
principals, 249
security, 249
No such file or directory error, 81
nonglobal zones, 558
description of, 277. See also zones
features of, 279-280
root file system models, 280-281
nonredundant pools, converting to mirrored pools, 490-491
no_content_check keyword (class files), 358
no_master_check keyword (class files), 358
How can we make this index more useful? Email us at indexes@quepublishing.com
nscl (Name Service Cache Daemon). See also name services
  command options, 259
  overview, 258-260
num_clients keyword (class files), 358
nvalias command, 166

O

object sets, ZFS file systems, 480
objects, Volume Manager objects, 177
OBP (OpenBoot PROM), 419
  interactively booting WAN boot client, 434-436
  noninteractively booting WAN boot client, 436
offlining devices, 493
optional parameters field (filesys), 352
optional parameters options (filesys), 352
options
  coreadm command, 64
  logger command, 208
  metaclear command, 147
  metadb command, 142
  metainit command, 146-147
    mirrors, creating, 156-157
    soft partitions, creating, 151
  metastat command, 150
  nscl command options, 259
  roleadd command, 191
  rolemod command, 193
OS installed boot environments
  adding patches, 459
  removing patches from, 458

P

package keyword (class files), 358-359
pages, space allocation, 53
paging, description of, 53
parameters, wanboot.conf, 429
partitioning keyword (class files), 360
partitions, creating soft partitions, 150-152
passwd files, creating, 236-237
passwd maps, 246
patch keyword (class files), 361-362
performing Solaris Live Upgrade from local DVDs, 449
pfinstall command
  class files (testing), 363-365
  options, 364
physical memory, description of, 52-53. See also RAM
ping, 36
  conditions for success, 24
ping success, conditions for, 24
pool keyword (class files), 360-361
pool1 command, 477
pools, ZFS, 552
practice exam, 565-581
  answers to, 583-590
preparing for exam, 11-12
pretesting, importance of, 11
principals, NIS+, 249
processes, Solaris Live Upgrade, 439-440
  activating new boot environments, 441-445
  creating new boot environments, 441-446
  displaying the status of new boot environments, 447
  luactivate
    on SPARC platforms, 455
    on x86/x64 platforms, 452-454
  upgrading new boot environments, 447-449
prof attr database, 199
profile diskettes. See configuration diskettes
profile names, 199. See also /etc/prof_attr databases
properties
  listing LDAP client properties, 257
  ZFS, 494-497
    native read-only ZFS properties, 495-496
    settable ZFS properties, 496-497
    setting, 497-500
pseudo driver. See metadisk driver
PXE (Preboot Execution Environment)
  configuring DHCP servers, 393-401
  exam question answers, 412-413
  exam questions, 407-411
  overview, 392
  preparing for, 393
RAID (Redundant Array of Inexpensive Disks), 124-125
  levels of, 125
  overview, 125
RAID 0, 126-127, 136
RAID 0+1 (mirrored stripe), 130
RAID 1, 128, 136
RAID 1+0, 130
RAID 5, 129-130, 135-136
security

SVM, 548-549
ZFS, 476

RAID 0 (concatenated), creating volumes, 146-149
RAID 0+1 (mirrored stripe), 130
RAID 5, 548
RAID-Z, 476
storage pools, 484
ZFS, 552

RAM (random-access memory), 52
RARP (reverse address resolution protocol), 327
JumpStart, 325
raw devices, Solaris Volume Manager, 139

RBAC (Role-Based Access Control), 189
authorizations databases, 197-199
components of, 195
exam question answers, 214-215
exam questions, 211-213
execution attributes databases, 201-203
exercises, 210-211
extended user attributes databases, 195-197
overview, 189, 555
rights profiles databases, 199-201
utilizing, 190-194
read policies, 134-135
ready state, zones (transitioning to), 289-290
rebooting zones, 291
recipients (hosts), 20
Redundant Array of Inexpensive Disks. See RAID
Regional Internet registries (RIRs), 21
registering Sun Connection Services, 101
remote file systems, mounting, 74-78
removing
patches on OS installed boot environments, 458
software packages from boot environments, 456-457
ZFS file systems, 479-480
ZFS storage pools, 480-481
renaming
ZFS file system, 478
ZFS snapshots, 510
replacing
devices in storage pools (ZFS), 515-517
ZFS file systems with ZFS clones, 513-514
replicas, 90
requirements
for Solaris Live Upgrade, 438
for WAN boot, 418-419, 561
for ZFS, 553
hardware and software requirements, 475
resilvering ZFS, 552
resource management (containers), 275-276
restoring ZFS snapshots, 510
review exercises, zones, 306-307
rights profiles (prof attr) database, RBAC, 199
RIRs (regional Internet registries), 21
Role-Based Access Control. See RBAC
roleadd command, 190-192
roledel command, 194
rolemod command, 193-194
rolling back ZFS snapshots, 511-512
root file system models (zones)
overview, 280
sparse root zones, 281
whole root zones, 281
root file systems
mirroring, 162-166, 169
on x86-based systems, 166-172
mirrors, troubleshooting, 174-176
unmirroring, 173-174
ZFS, 517-518
root passwords (JumpStart), setting, 370
root pools, creating new boot environments, 445-446
root_device keyword (class files), 362
RPC (remote procedure calls), 34-36
RPC services, 34-36
RPC: Program not registered error, 80
RPC: Unknown host error, 81
rules files (JumpStart)
check script, 341-342
matches, 340
overview, 333-336
requirements of, 337
syntax of, 336-337
rules keywords (JumpStart), 338-340
rules values (JumpStart), 338-340
saving ZFS snapshots, 510
scores, 12
secure nets files, 246-247
security
NIS, 246-247
NIS+, 249

How can we make this index more useful? Email us at indexes@quepublishing.com
senders (hosts), 20
server logging (NFS), 78-79
Server not responding error, NFS, 81
server problems (NIS), 248
servers, 20
DHCP servers, Booting WAN boot clients, 436-437
NFS, 69-70
WAN boot servers, 421-422
configuring, 422-423
services (network)
overview, 31-34
RPC services, 34-36
settable ZFS properties, 496-497
share command, 72-74, 504
NFS, 71-73
shareable file systems, Solaris Live Upgrade, 440
shared resources, 71. See also NFS
sharefs property, 505
sharing ZFS file systems, 504-506
showmount command, 84-85
[lt]size[gt] values (filesys), 351-352
Sizing swap space, 4
Slave servers (NIS), 244-245
[lt]slice[gt] values (filesys), 351
SMC (Solaris Management Console), 140
SMF, NIS, 243
snapshots, ZFS, 508, 552
creating, 508
destroying, 510
listing, 509
renaming, 510
rolling back, 511-512
saving and restoring, 510
SNMP (Simple Network Management Protocol), trap generating daemon, 138-139
snoop, 25, 36
displaying network information, 42
soft partitions, 32-133
creating, 150-152
software packages
adding to boot environments, 457-458
removing from boot environments, 456-457
software requirements for ZFS, 475
Solaris Flash, 560-561. See also Flash Archives
Solaris Live Upgrade, 437-438, 563-564
commands, 440
installing, 438-439
maintaining boot environments, 456
adding patches on OS installed boot environments, 459
adding software packages, 457-458
changing descriptions of, 460-461
changing names, 460
deleting inactive boot environments, 459-460
removing patches on OS installed boot environments, 458
removing software packages, 456-457
viewing configuration of, 461
performing from local DVDs, 449
processes, 439-440
activating new boot environments, 450-452
creating new boot environments, 441-446
displaying the status of new boot environments, 447
luactivate on SPARC platforms, 455
luactivate on x86/x64 platforms, 452-454
upgrading new boot environments, 447-449
requirements for, 438
upgrading from Flash Archive from a DVD, 450
Solaris Management Console (SMC), 140
Solaris Volume Manager. See SVM
Solaris zones. See also zones
ZFS, 518-519
adding ZFS datasets to nonglobal zones, 519-521
delegating ZFS datasets to nonglobal zones, 521-522
source status codes, name services, 225
sources, name services, 225
SPARC, JumpStart, 322
SPARC platform, luactivate, 455
SPARC systems, as install servers, 393
Sparse root zones, 281
Spray services, enabling/disabling, 33
stale NFS file handle message, troubleshooting NFS errors, 80
state databases (SVM), 133, 549-550
creating, 141-143
monitoring, 143-144
recovering from problems, 144-146
statements, ifdef, 204
states
ZFS, 552-553
zones, 278
storage pools

mirrored storage pools, 483
RAID-Z, 484
ZFS, 472-473, 550-551
adding devices to, 488-489
converting nonredundant pools to mirrored pools, 490-491
devices, 489-490
removing, 480-481
replacing devices, 515-517
taking devices offline and online, 492-493

storage volumes, 124, 133. See also SVM

stripes, 126-127, 134-136, 548
concatenated stripes, 134

striping

with distributed parity (RAID 5), 129-130
with parity (RAID 5), 135

study strategies

active strategies, 10
commonsense strategies, 11
macro/micro strategies, 10
overview, 10

su command (RBAC), 190, 193

submirrors, 134-135, 548. See also mirrors
placing offline, 160-162

subnets, booting on, 327

Sun Connection Services, registering, 101

Sun Java System Directory Server, 256. See also LDAP
(Lightweight Directory Access Protocol)
Sun Update Connection, 98

Sun Update Connection Proxy, 98

Sun Update Connection service, 97
Update Manager, 98-103
Update Manager Proxy, 103

Sun Update Manager, 98-103

Sun Update Manager Proxy, 103

SunSolve Patch and Updates Portal, 98

supernetting. See CIDR

Superuser access, assigning with RBAC
(Role-Based Access Control), 190

Svcadm command, 207

Svccfg command, 31

SVM (Solaris Volume Manager), 130-132
commands, 139-140
concatenated volumes, creating, 146-147
configurations planning, 136-139
disk sets, 133

exam question answers, 184
exam questions, 181-182
exercises, 180
hot spare pool, 137
metadisk driver, 132, 139
mirroring root file systems, 162-166, 169
mirrors
creating, 156, 158-159
troubleshooting, 174-176
objects, 547
overview, 132, 547-548
placing submirror offline, 160-162
RAIDs, 548-549
SNMP trap generating daemon, 138-139
soft partitions, 132
creating, 150-152
state databases, 133, 549-550
creating, 141-143
monitoring, 143-144
recovering from problems, 144-146
unmirroring systems, 159-160
root file systems, 173-174
volume statuses, monitoring, 149-150
volumes, 132
concatenated stripes, 134
concatenations, 133-134
expanding, 153-156
mirrors, 134-135
overview, 133
RAID 5, 135
stripes, 134-136

swap -l command, 56
swap -s command, 57

swap command, 542

swap files, 542. See also core dumps

swap monitoring tools, 56

swap spaces

caluclations, 57

exam question answers, 116-119

exam questions, 109-114
exercises, 105-106

NFS, 62
overview, 52-53
permissions, 61
setting up, 58-62

How can we make this index more useful? Email us at indexes@quepublishing.com
sizing, 54
TMPFS, 53-54
troubleshooting, 55
swaps, 53.  See also swap spaces
swaps, definition of, 542
switch files, name services, 222-226
switch template files (name services), 557
sys-unconfig command, 541
sysidcfg files
installing, 294
JumpStart, 366-367
name service keywords, 367-369
name services, 372
network keywords, 369-370
root passwords, setting, 370
time servers, setting, 371
zones, configuring, 294
syslog
exam question answers, 214-215
exam questions, 211-213
logger command, 208
overview, 203-208
syslogd daemon, 203-204
system hostnames, changing, 29-30
system_type keyword (class files), 362
T

TFTPboot, JumpStart, 327
time limits, 11
time servers (JumpStart), setting, 371
tips for success, 12-14
TMPFS (temporary file system), 53-54.  See also swap spaces
tools, swap monitoring tools, 56
trigger nodes, 87
troubleshooting
JumpStart, 375
client boot problems, 376
installation setup, 375
NFS errors, 80
NFS server not responding, still trying message, 81
NFS: service not responding error, 80
No such file or directory error, 81
PRC: Program not registered error, 80
RPC: Unknown host error, 81

Server not responding error, 81
Stale NFS file handle message, 80

Trusted Solaris, 202

U

UFS (UNIX file systems), 53
UFS files, in ZFS storage pools, 482-483
UIDs, resolving duplicates, 237
umount command, 78.  See also mount command
uname command, 165
uninitializing LDAP clients, 257
uninstalling zones, 291
UNIX file systems (UFS), 53
UnixEd.com, 7
unmirroring root file systems, 173-174
Update Manager, Sun Update Connection service, 98-103
Update Manager Proxy, Sun Update Connection service, 103
upgrading
Flash Archive from DVD, Solaris Live Update, 450
new boot environments, 447-449

usedisk keyword (class files), 351, 363
user attr database, 196
usermod command, 192
usernames, resolving duplicates, 237

V

validating rules files, 341-342
verifying operation of networks, 37-39
Veritas Volume Manager, overview, 176-178
viewing configurations of boot environments, 461
virtual devices, ZFS, 552
virtual memory, 53, 542.  See also swap files; swap spaces
virtual swap spaces, description of, 53
virtual volume management, SVM, 133
virtual volumes, 124.  See also SVM
volume Manager objects, 177-178
volumes
defined, 132
managing.  See SVM
RAID 0 (concatenated), creating, 146-148
RAID 0 (stripe), creating, 149
Veritas Volume Manager, 176-178
ZFS, 552
W

WAN boot, 418, 561
   clients, 563
   components of, 420-421, 562
   exam questions, 463-466
      answers, 466-467
   files, configuring, 423-428
   processes, 421
   requirements for, 418-419, 561
   wanboot.conf file, 428-430
WAN boot client, booting, 431
   with DHCP servers, 436-437
   from local CD/DVD, 431-434
   interactively from OBP, 434-436
   noninteractively from OBP, 436
WAN boot file system, 420
WAN boot miniroot, 420
WAN boot servers, 421-422
   configuring, 422-423
wanboot program, 420
wanboot-cgi, 420
wanboot.conf, 420
wanboot.conf file, 428-430
WANs (wide area networks), 418
Web-based interfaces, ZFS, 506-507
whole root zones, 281
wide area networks. See WANs
write policies, 135

X

X64 systems, JumpStart, 323-324
X86 clients
   booting, 402
   DHCP, 401
X86 systems
   JumpStart, 323-324
   Preserve, 347
X86-based systems, mirroring root file systems, 166-172
X86/x64 platform, luactivate, 452-454

Y

Yellow Pages, 556
ypcat command, 233

zpinit, 241-242
ypserv, 242

Z

z option
   zlogin, 296
   etc files, 556
   overview, 226
   etc/auth_attr databases, 197-199
   etc/auto_masters, 85-86
ZFS (Zettabyte File System), 472
   administration, 474
   basic file systems, creating, 476-478
   clones, 512-513
      destroying, 513
      replacing ZFS file systems, 513-514
   components of, 481-482
      disks in storage pools, 482
      files in storage pools, 482-483
   copy-on-write semantics, 473-474
   disk scrubbing, 514
   exam questions, 525-533
   exercises for, 524-525
   file systems, 550
      managing, 553-554
   hardware and software requirements, 475
   history of, 494
   mirrored storage pools, 483
   object sets, 480
   overview, 472
   properties, 494-497
      native read-only ZFS properties, 495-496
      settable ZFS properties, 496-497
      setting, 497-500
   RAID configurations, 476
   requirements for, 553
   root pool, 518
   snapshots, 508
      creating, 508
      destroying, 510
      listing, 509
      renaming, 510
      rolling back, 511-512
      saving and restoring, 510
   Solaris zones, 518-519
      adding ZFS datasets to nonglobal zones, 519-521
      delegating ZFS datasets to nonglobal zones, 521-522

How can we make this index more useful? Email us at indexes@quepublishing.com
states, 552-553
storage pools, 472-473, 550-551
adding devices to, 488-489
attaching devices, 489-490
converting nonredundant pools to mirrored pools, 490-491
detaching devices from mirrored pools, 491-492
displaying information, 484-488
removing, 480-481
replacing devices, 515-517
taking devices offline and online, 492-493
terminology for, 474-475
terms for, 552
Web-based management GUI, 506-507
zfs destroy command, 513
ZFS file systems, 473
listing, 478-479
mounting, 500-502
legacy mount points, 502-504
removing, 479-480
renaming, 478
sharing, 504-506
zfs mount command, 500
zfs rename command, 510
zfs rollback command, 511
ZFS root file system, 517-518
zfs set command, 497
zlogin
z option, 296
initial logins, 293
overview, 292-293
zone console, logging in, 294-295
zone console, logging in, 294-295
zoneadm command, 300
zoneadmd, description of, 282
zonecfg command, 558
overview, 283-287
properties/parameters, 285-286
resource types, 284-285
subcommands, 283-284
zones
backing up, 304
booting, 289-290
cloning, 302-304
commands, running in zones, 296
configuration files, viewing, 299
configurations, viewing, 287-289
configuring. See zonecfg command
zpool attach command, 489
zpool create command, 477, 484
zpool destroy command, 480
zpool detach command, 491
zpool history command, 494
zpool offline command, 492
zpool replace command, 515
zpool scrub command, 514
zpool status command, 516
zpools, 473, 550
zsched, description of, 282