Managing User Accounts and User Environments in Oracle® Solaris 11.1

Part No: E29010–01
October 2012

121119@25097

Contents

Preface

Managing User Accounts and User Environments in Oracle Solaris 11.1 is part of a documentation set that provides a significant portion of the Oracle Solaris system administration information. This guide contains information for both SPARC based and x86 based systems.

This book assumes you have completed the following tasks:

- Installed the Oracle Solaris software
- Set up all the networking software that you plan to use

For Oracle Solaris, new features that might be interesting to system administrators are covered in sections called *What's New in ... ?* in the appropriate chapters.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

For supported systems, see the *Oracle Solaris OS: Hardware Compatibility Lists*.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Oracle Solaris release. To use this book, you should have 1–2 years of UNIX system administration experience. Attending UNIX system administration training courses might be helpful.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface	Description	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`

TABLE P–2 Shell Prompts *(Continued)*

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

CHAPTER 1

Managing User Accounts and User Environments (Overview)

This is a list of the information in this chapter:

- "What's New or Changed in Managing User Accounts and User Environments?" on page 11
- "What Are User Accounts and Groups?" on page 14
- "Where User Account and Group Information Is Stored" on page 22
- "Commands That Are Used for Managing Users, Roles, and Groups" on page 28
- "Customizing a User's Work Environment" on page 29

For task-related information on managing user accounts and user environments, see Chapter 2, "Managing User Accounts by Using the Command-Line Interface (Tasks)," and Chapter 3, "Managing User Accounts by Using the User Manager GUI (Tasks)."

What's New or Changed in Managing User Accounts and User Environments?

The following features are new or changed in this release:

- "Security Changes That Impact User Account Management" on page 12
- "Introducing the User Manager GUI" on page 12
- "Administrative Editor (`pfedit`)" on page 13
- "`/var/user/$USER` Subdirectory" on page 13
- "`groupadd` Command Changes" on page 14
- "Failed Login Count Notification" on page 14

Security Changes That Impact User Account Management

The following feat ures have changed in this release:

- State transition refinements for the `password` command. This change clarifies which user accounts can and cannot be locked. The primary changes impact the `LK` and `NL` property definitions, and are as follows:

  `LK` The account is locked. The `passwd -l` command was run, or the account was automatically locked due to the number of authentication failures reaching the configured maximum that is allowed. See the `policy.conf(4)` and `user_attr(4)` man pages.

  `NL` The account is configured for non UNIX authentication. The `passwd -N` command was run. Starting with this release, accounts in this state can be locked by running the `passwd -l` command and unlocked by running the `passwd -u` command.
- Qualified Authorizations. Authorizations can be qualified to apply to specific objects like groups, zones, or file names. See "Administrative Editor (`pfedit`)" on page 13.
- The `profiles` command has been rewritten to manage rights profiles for local and LDAP scopes. Direct editing of role-based access control (RBAC) files is no longer supported.
- The ability to set a per-user Pluggable Authentication Module (PAM) policy (`pam_policy`) in a rights profile is available in this release. The `pam_policy` must be either an absolute path name to a `pam_conf`(4)-formatted file or the name of a `pam.conf`(4)-formatted file located in the `/etc/security/pam_policy` file. See `pam_user_policy(5)`.

  In addition to setting PAM policy in a rights profile, you can also directly set the `pam_policy` in the user's `user_attr` entry by using either the `useradd` or `usermod` command. See Example 2–1.
- Users and roles who are assigned the User Security rights profile can create new user accounts, as well as delegate some of their rights to other accounts, without becoming the `root` role.

For more information, see Part III, "Roles, Rights Profiles, and Privileges," in *Oracle Solaris 11.1 Administration: Security Services*.

Introducing the User Manager GUI

You can now set up and manage users, roles, and groups with the Oracle Solaris User Manager graphical user interface (GUI). The User Manager GUI is available in the desktop and is part of the Visual Panels project. The User Manager GUI replaces the Solaris Management Console GUI in this release. The tasks that you can perform with the User Manager GUI are essentially the same as those that can be performed by using the CLI, for example, the useradd, usermod, userdel, roleadd, rolemod, roledel commands.

For instructions on using the User Manager GUI, see Chapter 3, "Managing User Accounts by Using the User Manager GUI (Tasks)," and the online help.

Administrative Editor (pfedit)

An administrative editor (pfedit) can be used to edit system files in this release. If defined by the system administrator, the value of this editor is $EDITOR. If the editor is undefined, the editor defaults to the vi command.

Start the editor as follows:

```
$ pfedit system-filename
```

To edit system files by using the pfedit command, you or your role must have the solaris.admin.edit/*system-filename* authorization for the specific file that you are editing. Assigning this auth-sysfilename to an existing rights profile simplifies procedures that contain a mixture of Service Management Facility (SMF) commands and regular file edits. For example, if you are assigned the solaris.admin.edit/etc/security/audit_warn authorization, you can edit the audit_warn file.

The pfedit command can be used to edit most configuration files that are in the /etc directory, its subdirectories, and also application configuration files, for example, GNOME and Firefox files. The pfedit command *cannot* be used to edit system files that give a user power over a wide swath of a system, for example the ./etc/security/policy.conf file. You must have root access to edit such files. See the pfedit(1M) man page and Chapter 3, "Controlling Access to Systems (Tasks)," in *Oracle Solaris 11.1 Administration: Security Services*.

/var/user/$USER Subdirectory

Whenever a user logs in and successfully authenticates by using the pam_unix_cred module, a /var/user/$USER directory is explicitly created, if the directory does not already exist. This directory enables applications to store persistent data that is associated with a particular user on the host system. The /var/user/$USER directory is created upon initial credential establishment, as well during a secondary authentication when changing users by using the su, ssh, rlogin, and telnet commands. The /var/user/$USER directory does not require any administration. However, users should be aware of how the directory is created, its function, and that it is visible in the /var directory.

groupadd Command Changes

An administrator who has the solaris.group.manage authorization can create a group. At group creation, the system assigns the solaris.group.assign/*groupname* to the administrator, which gives the administrator complete control over that group. The administrator can then modify or delete that group, as needed. For more information, see the groupadd(1M) and groupmod(1M) man pages.

Failed Login Count Notification

The system now notifies users of failed authentication attempts, even if the user account is not configured to enforce failed logins. Users who fail to authenticate correctly, will see a message similar to following upon successful authentication:

```
Warning: 2 failed authentication attempts since last successful
authentication. The latest at Thu May 24 12:02 2012.
```

To suppress such notifications, create a ~/.hushlogin file.

What Are User Accounts and Groups?

The following information is described in this section:

- "User Account Components" on page 15
- "Guidelines for Assigning User Names, User IDs, and Group IDs" on page 21

One basic system administration task is to set up a user account for each user at a site. A typical user account includes the information a user needs to log in and use a system, without having the system's root password. User account components are described in "User Account Components" on page 15.

When you set up a user account, you can add the user to a predefined group of users. A typical use of groups is to set up group permissions on a file and directory, which allows access only to those users who are part of that group.

For example, you might have a directory containing confidential files that only a few users should be able to access. You could set up a group called topsecret that includes the users that are working on the topsecret project. In addition, you could set up the topsecret files with read permission for the topsecret group. That way, only the users in the topsecret group would be able to read the files.

A special type of user account, called a *role*, gives selected users special privileges. For more information, see "Role-Based Access Control (Overview)" in *Oracle Solaris 11.1 Administration: Security Services*.

User Account Components

The following sections describe the various components of a user account.

User (Login) Names

User names, also called *login names*, let users access their own systems and remote systems that have the appropriate access privileges. You must choose a user name for each user account that you create.

Consider establishing a standard way of assigning user names so that they are easier for you to track. Also, names should be easy for users to remember. A simple scheme when selecting a user name is to use the first name initial and first seven letters of the user's last name. For example, Ziggy Ignatz becomes `zignatz`. If this scheme results in duplicate names, you can use the first initial, middle initial, and the first six characters of the user's last name. For example, `Ziggy Top Ignatz` becomes `ztignatz`.

If this scheme still results in duplicate names, consider using the following scheme to create a user name:

- The first initial, middle initial, first five characters of the user's last name
- The number 1, or 2, or 3, and so on, until you have a unique name

Note – Each new user name must be distinct from any mail aliases that are known to the system or to a NIS domain. Otherwise, mail might be delivered to the alias rather than to the actual user.

For detailed guidelines on setting up user (login) names, see "Guidelines for Assigning User Names, User IDs, and Group IDs" on page 21.

User ID Numbers

Associated with each user name is a user identification number (UID). The UID number identifies the user name to any system on which the user attempts to log in. And, the UID number is used by systems to identify the owners of files and directories. If you create user accounts for a single individual on a number of different systems, always use the same user name and ID number. In that way, the user can easily move files between systems without ownership problems.

UID numbers must be a whole number that is less than or equal to 2147483647. UID numbers are required for both regular user accounts and special system accounts. The following table lists the UID numbers that are reserved for user accounts and system accounts.

TABLE 1–1 Reserved UID Numbers

UID Numbers	User or Login Accounts	Description
0 – 99	root, daemon, bin, sys, and so on	Reserved for use by the operating system
100 – 2147483647	Regular users	General purpose accounts
60001 and 65534	nobody and nobody4	NFS Anonymous users
60002	noaccess	Non-trusted users

Do not assign UIDs 0 through 99. These UIDs are reserved for allocation by Oracle Solaris. By definition, root always has UID 0, daemon has UID 1, and pseudo-user bin has UID 2. In addition, you should give uucp logins and pseudo user logins, such as who, tty, and ttytype, low UIDs so that they fall at the beginning of the passwd file.

For additional guidelines on setting up UIDs, see “Guidelines for Assigning User Names, User IDs, and Group IDs” on page 21.

As with user (login) names, you should adopt a scheme for assigning unique UID numbers. Some companies assign unique employee numbers. Then, administrators add a number to the employee number to create a unique UID number for each employee.

To minimize security risks, you should avoid reusing the UIDs from deleted accounts. If you must reuse a UID, “wipe the slate clean” so that the new user is not affected by attributes set for a former user. For example, a former user might have been denied access to a printer by being included in a printer deny list. However, that attribute might be inappropriate for the new user.

Using Large User IDs and Group IDs

UIDs and group IDs (GIDs) can be assigned up to the maximum value of a signed integer, or 2147483647.

The following table describes UID and GID limitations.

TABLE 1–2 Large UID and GID Limitation Summary

UID or GID	Limitations
262144 or greater	Users who use the cpio command with the default archive format to copy a file see an error message for each file. And, the UIDs and GIDs are set to nobody in the archive.

TABLE 1–2 Large UID and GID Limitation Summary (Continued)

UID or GID	Limitations
2097152 or greater	Users who use the `cpio` command with the `-H odc` format or the `pax -x cpio` command to copy files see an error message returned for each file. And, the UIDs and GIDs are set to `nobody` in the archive.
1000000 or greater	Users who use the `ar` command have their UIDs and GIDs set to `nobody` in the archive.
2097152 or greater	Users who use the `tar` command, the `cpio -H ustar` command, or the `pax -x tar` command have their UIDs and GIDs set to `nobody`.

UNIX Groups

A *group* is a collection of users who can share files and other system resources. For example, users who working on the same project could be formed into a group. A group is traditionally known as a UNIX group.

Each group must have a name, a group identification (GID) number, and a list of user names that belong to the group. A GID number identifies the group internally to the system.

The two types of groups that a user can belong to are as follows:

- **Primary group** – Specifies a group that the operating system assigns to files that are created by the user. Each user must belong to a primary group.
- **Supplemental groups** – Specifies one or more groups to which a user also belongs. Users can belong to up to 1024 supplemental groups.

For detailed guidelines on setting up group names, see “Guidelines for Assigning User Names, User IDs, and Group IDs” on page 21.

Sometimes, a user's secondary group is not important. For example, ownership of files reflect the primary group, not any secondary groups. Other applications, however, might rely on a user's secondary group memberships. For example, a user has to be a member of the `sysadmin` group (group 14) to use the Admintool software in previous Solaris releases. However, it does not matter if group 14 is the user's current primary group.

The `groups` command lists the groups that a user belongs to. A user can have only one primary group at a time. However, a user can temporarily change the user's primary group, with the `newgrp` command, to any other group in which the user is a member.

When adding a user account, you must assign a primary group for a user or accept the default group, `staff` (group 10). The primary group should already exist. If the primary group does not exist, specify the group by a GID number. User names are not added to primary groups. If user names were added to primary groups, the list might become too long. Before you can assign users to a new secondary group, you must create the group and assign it a GID number.

Groups can be local to a system or managed through a name service. To simplify group administration, you should use a name service such as NIS or a directory service such as LDAP. These services enable you to centrally manage group memberships.

User Passwords

You can specify a password for a user when you add the user. Or, you can force the user to specify a password when the user first logs in to the system. Although user names are publicly known, passwords must be kept secret and known only to users. Each user account should be assigned a password.

User passwords must comply with the following syntax:

- Password length must at least match the value that is identified by PASSLENGTH variable in the /etc/passwd file. By default, this value is set to 6.

  In this release, the default password hashing algorithm has changed to SHA256. As a result, there is no longer an eight character limitation for user passwords, as in previous Oracle Solaris releases. The eight character limitation only applies to passwords that use the older crypt_unix(5) algorithm, which has been preserved for backwards compatibility with any existing passwd file entries and NIS maps.

  The maximum number of characters for a password is dependent on the algorithm, either the crypt_unix algorithm for older passwords, and for all others, SHA256. If the password change is from an existing password and it is a crypt_unix password, the maximum length is set to 8, unless the policy.conf file requires a password algorithm change.

  The new password must match the complexity rules within the maximum number of characters that are allowed for the password algorithm. Thus, if using the crypt_unix algorithm, and you type a 20 character password, the password must match the complexity rules within the first 8 characters. If the password algorithm is any of the other algorithms, the password must match the complexity rules within the full password that is entered, which is 20 in this example.
- Each password must meet the configured complexity constraints that are specified in the /etc/default/passwd file.
- Each password must not be a member of the configured dictionary, as specified in the /etc/default/passwd file.
- For user accounts in a name services that support password history checking, if prior password history is defined, new passwords must not be contained in the prior password history.

Password rules are explained in detail in the passwd(1) man page.

To make your computer systems more secure, users should change their passwords periodically. For a high level of security, you should require users to change their passwords every six weeks. Once every three months is adequate for lower levels of security. System administration logins (such as `root` and `sys`) should be changed monthly, or whenever a person who knows the `root` password leaves the company or is reassigned.

Many breaches of computer security involve guessing a legitimate user's password. You should make sure that users avoid using proper nouns, names, login names, and other passwords that a person might guess just by knowing something about the user.

Good choices for passwords include the following:

- Phrases (`beammeup`).
- Nonsense words made up of the first letters of every word in a phrase. For example, `swotrb` for `SomeWhere Over The RainBow`.
- Words with numbers or symbols substituted for letters. For example, `sn00py` for `snoopy`.

Do not use these choices for passwords:

- Your name (spelled forwards, backwards, or jumbled)
- Names of family members or pets
- Car license numbers
- Telephone numbers
- Social Security numbers
- Employee numbers
- Words related to a hobby or interest
- Seasonal themes, such as Santa in December
- Any word in the dictionary

Home Directories

The home directory is the portion of a file system that is allocated to a user for storing private files. The amount of space you allocate for a home directory depends on the kinds of files the user creates, their size, and the number of files that are created.

A home directory can be located either on the user's local system or on a remote file server. In either case, by convention the home directory should be created as `/export/home/`*username*. For a large site, you should store home directories on a server. Use a separate file system for each user. For example, `/export/home/alice` or `/export/home/bob`. By creating separate file systems for each user, you can set properties or attributes based on each user's needs.

Regardless of where their home directory is located, users usually access their home directories through a mount point named, `/home/`*username*. When AutoFS is used to mount home directories, you are not permitted to create any directories under the `/home` mount point on any system. The system recognizes the special status of `/home` when AutoFS is active. For more information about auto-mounting home directories, see "Task Overview for Autofs Administration" in *Managing Network File Systems in Oracle Solaris 11.1*.

To use a home directory from anywhere on the network, you should always refer to the home directory as `$HOME`, not as `/export/home/`*username*. The latter is machine-specific. In addition, any symbolic links that are created in a user's home directory should use relative paths (for example, `../../../x/y/x`) so that the links are valid no matter where the home directory is mounted.

For more information about how home directories are added when you create user accounts by using the CLI, see "Guidelines for Setting Up User Accounts" on page 45.

Naming Services

If you are managing user accounts for a large site, you might want to consider using a name or directory service such as LDAP, or NIS. A name or directory service enables you to store user account information in a centralized manner instead of storing user account information in every system's `/etc` files. When you use a name service or a directory service for user accounts, users can move from system to system using the same user account, without having their information duplicated on every system. Using a naming service or a directory service also ensures consistent user account information.

User's Work Environment

Besides having a home directory to create and store files, users need an environment that gives them access to the tools and resources they need to do their work. When a user logs in to a system, the user's work environment is determined by initialization files. These files are defined by the user's startup shell, which can vary, depending on the release.

A good strategy for managing the user's work environment is to provide customized user initialization files, such as `.bash_profile`, `.bash_login`, `.kshrc`, or `.profile`, in the user's home directory.

Note – Do not use system initialization files, such as `/etc/profile` or `/etc/.login`, to manage a user's work environment. These files reside locally on systems and are not centrally administered. For example, if AutoFS is used to mount the user's home directory from any system on the network, you would have to modify the system initialization files on each system to ensure a consistent environment whenever a user moved from system to system.

For detailed information about customizing user initialization files for users, see "Customizing a User's Work Environment" on page 29.

For information about how to customize user accounts through the RBAC, see "Role-Based Access Control (Overview)" in *Oracle Solaris 11.1 Administration: Security Services* for more information.

Guidelines for Assigning User Names, User IDs, and Group IDs

User names, UIDs, and GIDs should be unique within your organization, which could span multiple domains.

Keep the following guidelines in mind when creating user or role names, UIDs, and GIDs:

- **User names –** Should contain from two to eight letters and numerals. The first character should be a letter. At least one character should be a lowercase letter.

  Note – Even though user names can include a period (.), underscore (_), or hyphen (-), using these characters is not recommended because they can cause problems with some software products.

- **System accounts –** Do not use any of the user names, UIDs, or GIDs that are contained in the default `/etc/passwd` and `/etc/group` files. Do not use the UIDs and GIDs, 0-99. These numbers are reserved for allocation by Oracle Solaris and should not be used by anyone. Note that this restriction also applies to numbers not currently in use.

  For example, `gdm` is the reserved user name and group name for the GNOME Display Manager daemon and should not be used for another user. For a complete listing of the default `/etc/passwd` and `/etc/group` entries, see Table 1–3 and Table 1–4.

  The `nobody` and `nobody4` accounts should never be used for running processes. These two accounts are reserved for use by NFS. Use of these accounts for running processes could lead to unexpected security risks. Processes that need to run as non-root should use the `daemon` or `noaccess` accounts.
- **System account configuration** – The configuration of the default system accounts should never be changed. This includes changing the login shell of a system account that is currently locked. The only exception to this rule is the setting of a password and password aging parameters for the `root` account.

  Note – Changing a password for a locked user account changes the password, but no longer unlocks the account at the same time. A second step to unlock the account by using the `passwd -u` command is now required.

Where User Account and Group Information Is Stored

The following information is described in this section:

- "Fields in the passwd File" on page 22
- "Default passwd File" on page 23
- "Fields in the shadow File" on page 25
- "Fields in the group File" on page 25
- "Default group File" on page 25
- "Commands for Obtaining User Account Information" on page 27

Depending on your site policy, user account and group information can be stored in your local system's /etc files or in a name or directory service as follows:

- The NIS name service information is stored in maps.
- The LDAP directory service information is stored in indexed database files.

Note – To avoid confusion, the location of the user account and group information is generically referred to as a *file* rather than as a *database*, *table*, or *map*.

Most user account information is stored in the passwd file. Password information is stored as follows:

- In the passwd file when you are using NIS
- In the /etc/shadow file when you are using /etc files
- In the people container when you are using LDAP

Password aging is available when you are using LDAP, but not NIS.

Group information is stored in the group file for NIS, and files. For LDAP, group information is stored in the group container.

Fields in the passwd File

The fields in the passwd file are separated by colons and contain the following information:

username:*password*:*uid*:*gid*:*comment*:*home-directory*:*login-shell*

For example:

```
kryten:x:101:100:Kryten Series 4000 Mechanoid:/export/home/kryten:/bin/csh
```

For a complete description of the fields in the passwd file, see the passwd(1) man page.

Default passwd File

The default passwd file contains entries for standard daemons. Daemons are processes that are usually started at boot time to perform some system-wide task, such as printing, network administration, or port monitoring.

```
root:x:0:0:Super-User:/root:/usr/bin/bash
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
dladm:x:15:65:Datalink Admin:/:
netadm:x:16:65:Network Admin:/:
netcfg:x:17:65:Network Configuration Admin:/:
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/var/lib/gdm:
zfssnap:x:51:12:ZFS Automatic Snapshots Reserved UID:/:/usr/bin/pfsh
upnp:x:52:52:UPnP Server Reserved UID:/var/coherence:/bin/ksh
xvm:x:60:60:xVM User:/:
mysql:x:70:70:MySQL Reserved UID:/:
openldap:x:75:75:OpenLDAP User:/:
webservd:x:80:80:WebServer Reserved UID:/:
postgres:x:90:90:PostgreSQL Reserved UID:/:/usr/bin/pfksh
svctag:x:95:12:Service Tag UID:/:
unknown:x:96:96:Unknown Remote UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
ftp:x:21:21:FTPD Reserved UID:/:
dhcpserv:x:18:65:DHCP Configuration Admin:/:
aiuser:x:60003:60001:AI User:/:
pkg5srv:x:97:97:pkg(5) server UID:/:
```

TABLE 1–3 Default passwd File Entries

User Name	User ID	Description
root	0	Reserved for superuser account
daemon	1	Umbrella system daemon associated with routine system tasks
bin	2	Administrative daemon associated with running system binaries to perform some routine system task
sys	3	Administrative daemon associated with system logging or updating files in temporary directories
adm	4	Administrative daemon associated with system logging
lp	71	Reserved for the Line printer daemon

TABLE 1–3 Default passwd File Entries *(Continued)*

User Name	User ID	Description
uucp	5	Assigned to the daemon that is associated with uucp functions
nuucp	9	Assigned to another daemon associated with uucp functions
dladm	15	Reserved for datalink administration
netadm	16	Reserved for network administration
netcfg	17	Reserved for network configuration administration
smmsp	25	Assigned to the Sendmail message submission program daemon
listen	37	Assigned to the Network Listener daemon
gdm	50	Assigned to the GNOME Display Manager daemon
zfssnap	51	Reserved for automatic snapshots
upnp	52	Reserved for UPnP server
xvm	60	Reserved for xVM user
mysql	70	Reserved for MySQL user
openldap	75	Reserved for OpenLDAP user
webservd	80	Reserved for WebServer access
postgres	90	Reserved for PostgresSQL access
svctag	95	Reserved for Service Tag Registry access
unknown	96	Reserved for unmappable remote users in NFSv4 ACLs
nobody	60001	Reserved for NFS Anonymous Access user
noaccess	60002	Reserved for No Access user
nobody4	65534	Reserved for SunOS 4.x NFS Anonymous Access user
ftp	21	Reserved for FTP access
dhcpserv	18	Reserved for DHCP server user
aiuser	60003	Reserved for AI user
pkg5srv	97	Reserved for pkg(5) depot server

Fields in the shadow File

The fields in the shadow file are separated by colons and contain the following information:

username:*password*:*lastchg*:*min*:*max*:*warn*:*inactive*:*expire*

The default password hashing algorithm is SHA256. The password hash for the user is similar to the following:

```
$5$cgQk2iUy$AhHtVGx5Qd0.W3NCKjikb8.KhOiA4DpxsW55sP0UnYD
```

For a complete description of the fields in the shadow file, see the shadow(4) man page.

Fields in the group File

The fields in the group file are separated by colons and contain the following information:

group-name:*group-password*:*gid*:*user-list*

For example:

```
bin::2:root,bin,daemon
```

For a complete description of the fields in the group file, see the group(4) man page.

Default group File

The default group file contains the following system groups that support some system-wide task, such as printing, network administration, or electronic mail. Most of these groups have corresponding entries in the passwd file.

```
root::0:
other::1:root
bin::2:root,daemon
sys::3:root,bin,adm
adm::4:root,daemon
uucp::5:root
mail::6:root
tty::7:root,adm
lp::8:root,adm
nuucp::9:root
staff::10:
daemon::12:root
sysadmin::14:
games::20:
smmsp::25:
gdm::50:
upnp::52:
```

```
xvm::60:
netadm::65:
mysql::70:
openldap::75:
webservd::80:
postgres::90:
slocate::95:
unknown::96:
nobody::60001:
noaccess::60002:
nogroup::65534:
ftp::21
pkg5srv::97:
```

TABLE 1–4 Default `group` File Entries

Group Name	Group ID	Description
`root`	`0`	Superuser group
`other`	`1`	Optional group
`bin`	`2`	Administrative group associated with running system binaries
`sys`	`3`	Administrative group associated with system logging or temporary directories
`adm`	`4`	Administrative group associated with system logging
`uucp`	`5`	Group associated with `uucp` functions
`mail`	`6`	Electronic mail group
`tty`	`7`	Group associated with `tty` devices
`lp`	`8`	Line printer group
`nuucp`	`9`	Group associated with `uucp` functions
`staff`	`10`	General administrative group.
`daemon`	`12`	Group associated with routine system tasks
`sysadmin`	`14`	Administrative group that is useful for system administrators
`smmsp`	25	Daemon for Sendmail message submission program
`gdm`	50	Group reserved for the GNOME Display Manager daemon
`upnp`	52	Group reserved for UPnP server
`xvm`	`60`	Group reserved for xVM user
`netadm`	65	Group reserved for network administration
`mysql`	`70`	Group reserved for MySQL user

TABLE 1–4 Default group File Entries *(Continued)*

Group Name	Group ID	Description
openldap	75	Reserved for OpenLDAP user
webservd	80	Group reserved for WebServer access
postgres	90	Group reserved for PostgresSQL access
slocate	95	Group reserved for Secure Locate access
unknown	96	Group reserved for unmappable remote groups in NFSv4 ACLs
nobody	60001	Group assigned for anonymous NFS access
noaccess	60002	Group assigned to a user or a process that needs access to a system through some application but without actually logging in
nogroup	65534	Group assigned to a user who is not a member of a known group
ftp	21	Group assigned for FTP access
pkg5srv	97	Group assigned to pkg(5) depot server

Commands for Obtaining User Account Information

The following table describes the commands that system administrators can use to obtain information about user accounts. This information is stored in various files within the /etc directory. Using these commands to obtain user account information is preferred over using the cat command to view similar information.

TABLE 1–5 Commands to Use to Obtain Information About Users

Command	Description	Man Page Reference
auths	Lists and manages authorizations.	auths(1)
getent	Gets a list of entries from the administrative database. The information generally comes from one or more of the sources that are specified for the /etc/nsswitch.conf database.	getent(1M)

TABLE 1–5 Commands to Use to Obtain Information About Users *(Continued)*

Command	Description	Man Page Reference
`logins`	Displays information about users, roles, and system logins. The output is controlled by the command options that are specified and can include user, role, system login, UID, `passwd` account field value, primary group, primary group ID, multiple group names, multiple group IDs, home directory, login shell, and password-aging parameters.	`logins`(1M)
`profiles`	Lists and manages rights profiles.	`profiles`(1)
`roles`	Displays the roles that are assigned to a user.	`roles`(1)
`userattr`	Displays the first value that is found for `attribute_name`. If a user is not specified, the user is taken from the real user ID of the process. Attribute names are defined in `user_attr`(4) and `prof_attr`(4). **Note –** This command is new in Oracle Solaris 11.	Example 2–1

Commands That Are Used for Managing Users, Roles, and Groups

Note – The Solaris Management Console GUI, and the CLI that is associated with this GUI, are no longer supported.

The following commands are available for managing users, roles, and groups.

TABLE 1–6 Commands That Are Used to Manage Users, Roles, and Groups

Man Page for Command	Description	For Additional Information
`useradd`(1M)	Creates users locally or in an LDAP repository.	"How to Add a User" on page 46

TABLE 1–6 Commands That Are Used to Manage Users, Roles, and Groups *(Continued)*

Man Page for Command	Description	For Additional Information
`usermod`(1M)	Changes user properties locally or in an LDAP repository. If the user properties are security-relevant, such as role assignment, this task might be restricted to your security administrator or to the `root` role.	"How to Modify a User" on page 47 "How to Change the Security Attributes of a User" in *Oracle Solaris 11.1 Administration: Security Services*
`userdel`(1M)	Deletes a user from the system or from the LDAP repository. Can involve additional cleanup, such as `cron` job removal.	"How to Delete a User" on page 48
`roleadd`(1M) `rolemod`(1M) `roledel`(1M)	Manages roles locally or in an LDAP repository. Roles cannot log in. Users assume an assigned role to perform administrative tasks.	"How to Create a Role" in *Oracle Solaris 11.1 Administration: Security Services* "Initially Configuring RBAC (Task Map)" in *Oracle Solaris 11.1 Administration: Security Services*
`groupadd`(1M) `groupmod`(1M) `groupdel`(1M)	Manages groups locally or in an LDAP repository.	"How to Add a Group" on page 49

Customizing a User's Work Environment

The following information is described in this section:

- "Using Site Initialization Files" on page 30
- "Avoiding Local System References" on page 31
- "Shell Features" on page 31
- "Bash and `ksh93` Shell History" on page 32
- "Bash and `ksh93` Shell Environment Variables" on page 33
- "Customizing the Bash Shell" on page 35
- "`MANPATH` Environment Variable" on page 36
- "`PATH` Environment Variable" on page 36
- "Locale Variables" on page 37
- "Default File Permissions (`umask`)" on page 38
- "Customizing a User Initialization File" on page 39

Part of setting up a user's home directory is providing user initialization files for the user's login shell. A *user initialization file* is a shell script that sets up a work environment for a user after the user logs in to a system. Basically, you can perform any task in a user initialization file that you can do in a shell script. However, a user initialization file's primary job is to define the characteristics of a user's work environment, such as a user's search path, environment variables, and windowing environment. Each login shell has its own user initialization file, or files, which are listed in the following table. Note that the default user initialization file for both the bash and ksh93 shells is /etc/skel/local.profile.

TABLE 1–7 Bash and ksh93 User Initialization Files

Shell	User Initialization File	Purpose
bash	$HOME/.bash_profile $HOME/.bash_login $HOME/.profile	Defines the user's environment at login
ksh93	/etc/profile $HOME/.profile	Defines the user's environment at login
	$ENV	Defines the user's environment at login within the file and is specified by the Korn shell's ENV environment variable

You can use these files as a starting point and then modify them to create a standard set of files that provide the work environment common to all users. You can also modify these files to provide the working environment for different types of users.

For step-by-step instructions on how to create sets of user initialization files for different types of users, see “How to Customize User Initialization Files” on page 43.

Using Site Initialization Files

The user initialization files can be customized by both the administrator and the user. This important task can be accomplished with centrally located and globally distributed user initialization files that are called, *site initialization files*. Site initialization files enable you to continually introduce new functionality to the user's work environment, while enabling the user to customize the user's initialization file.

When you reference a site initialization file in a user initialization file, all updates to the site initialization file are automatically reflected when the user logs in to the system or when a user starts a new shell. Site initialization files are designed for you to distribute site-wide changes to users' work environments that you did not anticipate when you added the users.

You can customize a site initialization file the same way that you customize a user initialization file. These files typically reside on a server, or set of servers, and appear as the first statement in a user initialization file. Also, each site initialization file must be the same type of shell script as the user initialization file that references it.

To reference a site initialization file in a bash or ksh93 user initialization file, place a line at the beginning of the user initialization file similar to the following line:

```
. /net/machine-name/export/site-files/site-init-file
```

Avoiding Local System References

Do not add specific references to the local system in the user initialization file. The instructions in a user initialization file should be valid, regardless of which system the user logs into.

For example:

- To make a user's home directory available anywhere on the network, always refer to the home directory with the variable `$HOME`. For example, use `$HOME/bin` instead of `/export/home/`*username*`/bin`. The `$HOME` variable works when the user logs in to another system, and the home directories are auto-mounted.
- To access files on a local disk, use global path names, such as `/net/`*system-name*/*directory-name*. Any directory referenced by `/net/`*system-name* can be mounted automatically on any system on which the user logs in, assuming the system is running AutoFS.

Shell Features

This Oracle Solaris release supports the following shell features and behavior:

- The user account that is created when you install the Oracle Solaris release is assigned the GNU Bourne-Again Shell (bash) by default.
- The standard system shell (`bin/sh`) is now the Korn Shell 93 (`ksh93`).
- The default interactive shell is the Bourne-again (`bash`) shell (`/usr/bin/bash`).
- Both the bash and `ksh93` shells feature command-line editing, which means you can edit commands before executing them.
- There are a few ways in which you can display default shell and path information:
  - Use the `echo $SHELL` and `which` commands:

    ```
    $ grep root /etc/passwd
    root:x:0:0:Super-User:/root:/usr/bin/bash

    $ echo $SHELL /usr/bin/bash
    $ which ksh93 /usr/bin/ksh93
    ```
  - Use the `pargs` command:

    ```
    ~$ pargs -l $$
    /usr/bin/i86/ksh93
    ```
- The `ksh93` shell also has a built-in variable called `.sh.version`, which can be displayed as follows:

  ```
  ~$ echo ${.sh.version}
  Version jM 93u 2011-02-08
  ```

- To change to a different shell, type the path of the shell that you want to use.
- To exit a shell, type `exit`.

The following table describes the shell options that are supported in Oracle Solaris.

TABLE 1–8 Basic Shell Features in the Oracle Solaris Release

Shell	Path	Comments
Bourne-Again Shell (bash)	`/usr/bin/bash`	Default shell for users that are created by an installer, as well as the `root` role The default (interactive) shell for users that are created with the `useradd` command, as well as the `root` role, is `/usr/bin/bash`. The default path is `/usr/bin:/usr/sbin`.
Korn Shell	`/usr/bin/ksh`	`ksh93` is the default shell in this Oracle Solaris release
C Shell and enhanced C Shell	`/usr/bin/csh` and `/usr/bin/tcsh`	C Shell and enhanced C Shell
POSIX-compliant Shell	`/usr/xpg4/bin/sh`	POSIX-compliant shell
Z Shell	`/usr/bin/zsh`	Z Shell

Note – The Z Shell (`zsh`) and the enhanced C Shell (`tsch`) are not installed on your system by default. To use either of these shells, you must first install the required software packages.

Bash and ksh93 Shell History

Both the `bash` and `ksh93` shells record a history of all of the commands that you run. This history is kept on a per-user basis, which means the history is persistent between login sessions, as well as representative of all your login sessions.

For example, if you are in a bash shell, you can display the complete history of the commands that you have run, as follows:

```
$ history
1 ls
2 ls -a
3 pwd
4 whoami
.
```

```
.
.
```

To display a number of previous commands, include an integer in the command:

```
$ history 2
12 date
13 history
```

For more information, see the history(1) man page.

Bash and ksh93 Shell Environment Variables

The bash and ksh93 shells store special variable information that is known to the shell as an*environment variable*. To view a complete list of the current environment variables for the bash shell, use the declare command:

```
$ declare
BASH=/usr/bin/bash
BASH_ARGC=()
BASH_ARGV=()
BASH_LINEND=()
BASH_SOuRCE=()
BASH_VERSINFO=([0]=’’3’’ [1]=’’2’’ [2]=’’25’’ [3]=’’1’’
[4]=’’release’’ [5]’’
.
.
.
```

For the ksh93 shell, use the set command, which is the bash shell's declare command equivalent:

```
$ set
  COLUMNS=80
  ENV=’$HOME/.kshrc’
  FCEDIT=/bin/ed
  HISTCMD=3
  HZ=’’
  IFS=$’ \t\n’
  KSH_VERSION=.sh.version
  LANG=C
  LINENO=1
  .
  .
  .
```

To print environment variables for either shell, use the echo or printf command. For example:

```
$ echo $SHELL
/usr/bin/bash
$ printf ’’$PATH\n’’
/usr/bin:/usr/sbin
```

Note – Environment variables do not persist between sessions. To set up environment variables that remain consistent between logins, you must make the changes in the .bashrc file.

A shell can have two types of variables:

Environment variables
: Specifies variables that are exported to all processes that are spawned by the shell. The export command is used to export a variable. For example:

```
export VARIABLE=value
```

These settings can be displayed by using the env command. A subset of environment variables, such as PATH, affects the behavior of the shell itself.

Shell (local) variables
: Specifies variables that affect only the current shell.

In a user initialization file, you can customize a user's shell environment by changing the values of the predefined variables or by specifying additional variables.

The following table provides more details about the shell and environment variables that are available in the Oracle Solaris release.

TABLE 1–9 Shell and Environment Variable Descriptions

Variable	Description
CDPATH	Sets a variable that is used by the cd command. If the target directory of the cd command is specified as a relative path name, the cd command first searches for the target directory in the current directory (.). If the target is not found, the path names that are listed in the CDPATH variable are searched consecutively until the target directory is found and the directory change is completed. If the target directory is not found, the current working directory is left unmodified. For example, the CDPATH variable is set to /home/jean, and two directories exist under /home/jean, bin, and rje. If you are in the /home/jean/bin directory and type cd rje, you change directories to /home/jean/rje, even though you do not specify a full path.
HOME	Sets the path to the user's home directory.
LANG	Sets the locale.
LOGNAME	Defines the name of the user that is currently logged in. The default value of LOGNAME is automatically set by the login program to the user name that is specified in the passwd file. You should only need to refer to, not reset, this variable.
MAIL	Sets the path to the user's mailbox.

TABLE 1–9 Shell and Environment Variable Descriptions *(Continued)*

Variable	Description
`MANPATH`	Sets the hierarchies of man pages that are available. **Note –** Starting with Oracle Solaris 11, the `MANPATH` environment variable is no longer required. The `man` command determines the appropriate `MANPATH`, based on the `PATH` environment variable setting.
`PATH`	Specifies, in order, the directories that the shell searches to find the program to run when the user types a command. If the directory is not in the search path, users must type the complete path name of a command. As part of the login process, the default `PATH` is automatically defined and set as specified in `.profile`. The order of the search path is important. When identical commands exist in different locations, the first command that is found with that name is used. For example, suppose that `PATH` is defined in the shell syntax as `PATH=/usr/bin:/usr/sbin:$HOME/bin` and a file named `sample` resides in both `/usr/bin` and `/home/jean/bin`. If the user types the command `sample` without specifying its full path name, the version that is found in `/usr/bin` is used.
`PS1`	Defines the shell prompt for the bash or `ksh93` shell.
`SHELL`	Sets the default shell used by `make`, `vi`, and other tools.
`TERMINFO`	Names a directory where an alternate `terminfo` database is stored. Use the `TERMINFO` variable in either the `/etc/profile` or `/etc/.login` file. For more information, see the `terminfo(4)`man page. When the `TERMINFO` environment variable is set, the system first checks the `TERMINFO` path defined by the user. If the system does not find a definition for a terminal in the `TERMINFO` directory defined by the user, it searches the default directory, `/usr/share/lib/terminfo`, for a definition. If the system does not find a definition in either location, the terminal is identified as "dumb."
`TERM`	Defines the terminal. This variable should be reset in either the `/etc/profile` or `/etc/.login` file. When the user invokes an editor, the system looks for a file with the same name that is defined in this environment variable. The system searches the directory referenced by `TERMINFO` to determine the terminal characteristics.
`TZ`	Sets the time zone. The time zone is used to display dates, for example, in the `ls -l` command. If `TZ` is not set in the user's environment, the system setting is used. Otherwise, Greenwich Mean Time is used.

Customizing the Bash Shell

To customize your bash shell, add the information to the `.bashrc` file that is located in your home directory. The initial user that is created when you install Oracle Solaris has a `.bashrc` file that sets the `PATH`, `MANPATH`, and command prompt. For more information, see the `bash(1)` man page.

MANPATH Environment Variable

The MANPATH environment variable specifies where the man command looks for reference manual (man) pages. The MANPATH is set automatically based on a user's PATH value, but it generally includes /usr/share/man and usr/gnu/share/man.

Note that a user's MANPATH environment variable can be modified independent of the PATH environment variable. A one to one equivalent of the associated man page locations, with directories in the user's $PATH, is not required.

PATH Environment Variable

When the user executes a command by using the full path, the shell uses that path to find the command. However, when users specify only a command name, the shell searches the directories for the command in the order specified by the PATH variable. If the command is found in one of the directories, the shell executes the command.

A default path is set by the system. However, most users modify it to add other command directories. Many user problems related to setting up the environment and accessing the correct version of a command or a tool can be traced to incorrectly defined paths.

Setting Path Guidelines

Here are some guidelines for setting up efficient PATH variables:

- If you must include the current directory (.) in your path, it should be placed last. Including the current directory in your path is a security risk because some malicious person could hide a compromised script or executable in the current directory. Consider using absolute path names instead.
- Keep the search path as short as possible. The shell searches each directory in the path. If a command is not found, long searches can slow down system performance.
- The search path is read from left to right, so you should put directories for commonly used commands at the beginning of the path.
- Make sure that directories are not duplicated in the path.
- Avoid searching large directories, if possible. Put large directories at the end of the path.
- Put local directories before NFS mounted directories to lessen the chance of “hanging” when the NFS server does not respond. This strategy also reduces unnecessary network traffic.

Locale Variables

The LANG and LC environment variables specify the locale-specific conversions and conventions for the shell. These conversions and conventions include time zones, collation orders, and formats of dates, time, currency, and numbers. In addition, you can use the stty command in a user initialization file to indicate whether the terminal session will support multibyte characters.

The LANG variable sets all possible conversions and conventions for the given locale. You can set various aspects of localization separately through these LC variables: LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_NUMERIC, LC_MONETARY, and LC_TIME.

Note – By default, Oracle Solaris 11 installs UTF-8 based locales only.

The following table describes the environment variable values for the core Oracle Solaris 11 locales.

TABLE 1–10 Values for LANG and LC Variables

Value	Locale
en_US.UTF-8	English, United States (UTF-8)
fr_FR.UTF-8	French, France (UTF-8)
de_DE.UTF-8	German, Germany (UTF-8)
it_IT.UTF-8	Italian, Italy (UTF-8)
ja_JP.UTF-8	Japanese, Japan (UTF-8)
ko_KR.UTF-8	Korean, Korea (UTF-8)
pt_BT.UTF-8	Portuguese, Brazil (UTF-8)
zh_CN.UTF-8	Simplified Chinese, China (UTF–8)
es_ES.UTF-8	Spanish, Spain (UTF-8)
zh_TW.UTF-8	Traditional Chinese, Taiwan (UTF-8)

EXAMPLE 1–1 Setting the Locale Using the LANG Variables

In a Bourne or Korn shell user initialization file, you would add the following:

```
LANG=de_DE.ISO8859-1; export LANG
```

```
LANG-de_DE.UTF-8; export LANG
```

Default File Permissions (umask)

When you create a file or directory, the default file permissions assigned to the file or directory are controlled by the *user mask*. The user mask is set by the umask command in a user initialization file. You can display the current value of the user mask by typing umask and pressing Return.

The user mask contains the following octal values:

- The first digit sets permissions for the user
- The second digit sets permissions for group
- The third digit sets permissions for other, also referred to as world

Note that if the first digit is zero, it is not displayed. For example, if the user mask is set to 022, 22 is displayed.

To determine the umask value that you want to set, subtract the value of the permissions you want from 666 (for a file) or 777 (for a directory). The remainder is the value to use with the umask command. For example, suppose you want to change the default mode for files to 644 (rw-r--r--). The difference between 666 and 644 is 022, which is the value you would use as an argument to the umask command.

You can also determine the umask value you want to set by using the following table. This table shows the file and directory permissions that are created for each of the octal values of umask.

TABLE 1–11 Permissions for umask Values

umask Octal Value	File Permissions	Directory Permissions
0	rw-	rwx
1	rw-	rw-
2	r--	r-x
3	r--	r--
4	-w-	-wx
5	-w-	-w-
6	--x	--x
7	--- (none)	--- (none)

The following line in a user initialization file sets the default file permissions to rw-rw-rw-.

```
umask 000
```

Customizing a User Initialization File

The following is an example of the `.profile` user initialization file. You can use this file to customize your own user initialization files. This example uses system names and paths that you will need to modify for your particular site.

EXAMPLE 1–2 The `.profile` File

```
(Line 1) PATH=$PATH:$HOME/bin:/usr/local/bin:/usr/gnu/bin:.
(Line 2) MAIL=/var/mail/$LOGNAME
(Line 3) NNTPSERVER=server1
(Line 4) MANPATH=/usr/share/man:/usr/local/man
(Line 5) PRINTER=printer1
(Line 6) umask 022
(Line 7) export PATH MAIL NNTPSERVER MANPATH PRINTER
```

1. Defines the user's shell search path.
2. Defines the path to the user's mail file.
3. Defines the user's time/clock server.
4. Defines the user's search path for man pages.
5. Defines the user's default printer.
6. Sets the user's default file creation permissions.
7. Sets the listed environment variables.

◆ ◆ ◆ CHAPTER 2

Managing User Accounts by Using the Command-Line Interface (Tasks)

This chapter provides basic information for setting up and managing user accounts by using the command-line interface (CLI).

For overview information about managing user accounts and user environments, see Chapter 1, "Managing User Accounts and User Environments (Overview)."

For information about managing users and roles by using the User Manager graphical user interface (GUI), see Chapter 3, "Managing User Accounts by Using the User Manager GUI (Tasks)."

Setting Up and Managing User Accounts by Using the CLI

The following tasks describe how to set up and manage user accounts by using the CLI.

Setting Up and Managing User Accounts by Using the CLI (Task Map)

Task	Description	For Instructions
Gather user information.	Use a standard form to gather user information to help you keep user information organized.	"Gathering User Information" on page 42
Customize user initialization files.	You can set up user initialization files to provide new users with consistent environments.	"How to Customize User Initialization Files" on page 43
Change account defaults for all roles.	Change the default home directory and skeleton directory for all roles.	"How to Change Account Defaults For All Roles" on page 44

Task	Description	For Instructions
Create a user account.	Using the account defaults that you set up, create a local user by using the `useradd` command.	"How to Add a User" on page 46
Modify a user account.	Modify a user's login information on the system.	"How to Modify a User" on page 47
Delete a user account.	Delete a user account by using the `userdel` command.	"How to Delete a User" on page 48
Create, then assign a role to perform an administrative task.	Using the account defaults that you set up, create a local role to enable the user to perform a specific administrative command or task.	"How to Create a Role" in *Oracle Solaris 11.1 Administration: Security Services* "How to Assign a Role" in *Oracle Solaris 11.1 Administration: Security Services*
Create a group.	Create a new group by using the `groupadd` command.	"How to Add a Group" on page 49
Add security attributes to a user account.	After you set up a local user account, you can add the required security attributes.	"How to Change the Security Attributes of a User" in *Oracle Solaris 11.1 Administration: Security Services*
Share a user's home directory.	You must share the user's home directory so that the directory can be remotely mounted from the user's system.	"How to Share Home Directories That Are Created as ZFS File Systems" on page 50
Manually mount a user's home directory.	Typically, you do not need to manually mount user home directories that are created as a ZFS file system. The home directory is mounted automatically when it is created and also at boot time from the SMF local file system service.	"Manually Mounting a User's Home Directory " on page 51

Gathering User Information

When setting up user accounts you can create a form similar to the following form to gather information about users before setting up their accounts.

Item	Description
User Name:	

Item	Description
Role Name:	
Profiles or Authorizations:	
UID:	
Primary Group:	
Secondary Groups:	
Comment:	
Default Shell:	
Password Status and Aging:	
Home Directory Path Name:	
Mounting Method:	
Permissions on Home Directory:	
Mail Server:	
Add to These Mail Aliases:	
Desktop System Name:	

▼ How to Customize User Initialization Files

1 **Assume the `root` role or a role that has the User Management rights profile.**

```
$ su -
Password:
#
```

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Create a skeleton directory for each type of user.**

```
# mkdir /shared-dir/skel/user-type
```

shared-dir The name of a directory that is available to other systems on the network.

user-type The name of a directory to store initialization files for a type of user.

3 **Copy the default user initialization files into the directories that you created for different types of users.**

4. **Edit the user initialization files for each user type and customize them based on your site's needs.**

   For a detailed description on the ways to customize the user initialization files, see "Customizing a User's Work Environment" on page 29.

5. **Set the permissions for the user initialization files.**

   ```
   # chmod 744 /shared-dir/skel/user-type/.*
   ```

6. **Verify that the permissions for the user initialization files are correct.**

   ```
   # ls -la /shared-dir/skel/*
   ```

▼ How to Change Account Defaults For All Roles

In the following procedure, the administrator has customized a `roles` directory . The administrator changes the default home directory and skeleton directory for all roles.

1. **Assume the `root` role or a role that has the User Management rights profile.**

   See "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Create a custom roles directory. For example:**

   ```
   # roleadd -D
   group=other,1  project=default,3  basedir=/home
   skel=/etc/skel  shell=/bin/pfsh  inactive=0
   expire=  auths=  profiles=All  limitpriv=
   defaultpriv=  lock_after_retries=
   ```

3. **Change the default home directory and skeleton directory for all roles. For example:**

   ```
   # roleadd -D -b /export/home -k /etc/skel/roles
   # roleadd -D
   group=staff,10  project=default,3  basedir=/export/home
   skel=/etc/skel/roles  shell=/bin/sh  inactive=0
   expire=  auths=  profiles=  roles=  limitpriv=
   defaultpriv=  lock_after_retries=
   ```

   Future uses of the **`roleadd`** command create home directories in `/export/home`, and populate the roles' environment from the `/etc/skel/roles` directory.

Guidelines for Setting Up User Accounts

Note the following guidelines for setting up user accounts by using the CLI:

- In this release, user accounts are created as Oracle Solaris ZFS file systems. As an administrator, when you create user accounts, you are giving users their own file system and their own ZFS dataset. Every home directory that is created by using the `useradd` and `roleadd` commands places the home directory of the user on the `/export/home` file system as an *individual* ZFS file system. As a result, users have the ability to back up their home directories, create ZFS snapshots of their home directories, and replace files in their current home directory from the ZFS snapshots that they created.
- To set up user accounts, you must assume the `root` role or a role that has the appropriate rights profile, for example, the User Management rights profile. See "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.
- When you create a user account with the `useradd` command, you must specify the `-m` option in the command syntax. Otherwise, a home directory will not be created for the user.

  For example, the following command will create a home directory for the user `jdoe`:

  ```
  # useradd -m jdoe
  ```

  But, the following syntax will *not* create a home directory for the user:

  ```
  # useradd jdoe
  ```

  Note – The only exception to this rule is if you want the `pam_zfs_key` module to create an encrypted home directory for the user. In this case, you would *not* specify the `-m` option with the `useradd` command. See the `pam_zfs_key(5)` and `zfs_encrypt(1M)` man pages.

- The `useradd` command creates entries in the `auto_home` map *only* if the `-d` option is specified with `hostname:/pathname`. Otherwise, the path name that is specified is updated as the home directory for the user in the `passwd` database, and no `auto_home` map entry is created. Home directories that are specified in the `auto_home` automounter map are only mounted if the `autofs` service is enabled.

  For example, if you specify the `-d` option to create a user as follows, the user is created without an `auto_home` entry, and the `passwd` entry specifies `/export/home/user1` as the user's home directory:

  ```
  # useradd -d /export/home/user1 user1
  ```

  But, if you use the `-d` option to create the user as follows, the user with have an `auto_home` entry, and the `passwd` database will contain `/home/user1`, indicating a dependency on the `autofs` service:

  ```
  # useradd -d localhost:/export/home/user1 user1
  ```

- If the pathname of the home directory includes a remote host specification, for example, `foobar:/export/home/jdoe`, then the home directory for `jdoe` must be created on the system `foobar`. The default pathname is `localhost:/export/home/`*username*.
- When the file system is a ZFS dataset, which is the case for all of Oracle Solaris 11, the user's home directory is created as a child ZFS dataset, with the ZFS permission to take snapshots delegated to the user. If a pathname is specified that does not correspond to a ZFS dataset, then a regular directory is created. If the `-S ldap` option is specified, then the `auto_home` map entry is updated on the LDAP server instead of the local `auto_home` map.

▼ How to Add a User

In this release, user accounts are created as Oracle Solaris ZFS file systems. Every home directory that is created by using the `useradd` and `roleadd` commands places the home directory of the user on the `/export/home` file system as an *individual* ZFS file system.

The `useradd` command creates entries in the `auto_home` map *only* if the `-d` option is specified with `hostname:/pathname`. Otherwise, the pathname that is specified is updated as the home directory for the user in the `passwd` database, and no `auto_home` map entry is created. Home directories that are specified in the `auto_home` automounter map are only mounted if the `autofs` service is enabled.

1 **Assume the `root` role or a role that has the User Management rights profile.**

See "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Create a local user.**

By default, the user is created locally. If you include the `-S ldap` option, the user is created in an existing LDAP repository.

```
# useradd -d dir -m username
```

`useradd` Creates an account for the specified user.

`-d` Specifies the location of the home directory of the user.

Use the `-d localhost:/export/home/`*username* instead of `-d /export/home/`*username* to force the entry to be written to `auto_home`.

`-m` Creates a local home directory on the system for the user.

If you specify the `-d` *dir* option as follows, the user is created without an `auto_home` entry, and the `passwd` entry specifies `/export/home/user1` as the user's home directory:

```
# useradd -d /export/home/user1 user1
```

If you specify the `-d` *dir* option as follows, the user with have an `auto_home` entry, and the `passwd` database will contain `/home/user1`, indicating a dependency on the `autofs` service:

```
# useradd -d localhost:/export/home/user1 user1
```

Note – If you want the `pam_zfs_key` module to create an encrypted home directory for the user. In this case, do *not* specify the `-m` option with the `useradd` command. See "Guidelines for Setting Up User Accounts" on page 45.

For a detailed description of all of the options and arguments that you can specify with the `useradd` command, see the `useradd`(1M) man page.

Note – The account is locked until you assign the user a password.

---

3 **Assign the user a password.**

```
# passwd username
New password:      Type user password
Re-enter new password:      Retype password
```

For more command options, see the `useradd`(1M) and `passwd`(1) man pages.

See Also After creating a user, you might need to perform some additional tasks, including adding and assigning roles to a user, listing and changing the rights profiles of a user, and changing the RBAC properties of a user. For more information, see the following references:

- "How to Create a Role" in *Oracle Solaris 11.1 Administration: Security Services* and "How to Assign a Role" in *Oracle Solaris 11.1 Administration: Security Services*
- "How to View All Defined Security Attributes" in *Oracle Solaris 11.1 Administration: Security Services*
- "How to Create a Rights Profile" in *Oracle Solaris 11.1 Administration: Security Services*
- "How to Change the Security Attributes of a User" in *Oracle Solaris 11.1 Administration: Security Services*

▼ How to Modify a User

The `usermod` command is used to change the definition of a user's login and make appropriate login-related file system changes for the user.

1 **Assume the `root` role or a role that has the User Management rights profile.**

See "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Modify the user account, as required.**

See the `usermod`(1M) man page for details about the arguments and options that you can specify with the `usermod` command.

For example, to add a role to a user, you would type:

```
# usermod -R role username
```

Example 2–1 Setting Per-User PAM Policy by Modifying a User's Account

The following example shows how to modify a user to set PAM policy. This particular modification specifies that user `jdoe` should only be authenticated with the Kerberos V5 protocol for all PAM services. See `pam_user_policy(5)` for more information.

```
# usermod -K pam_policy=krb5_only jdoe
```

See Also See the following references for additional examples of modifying a user:

- "How to Assign a Role" in *Oracle Solaris 11.1 Administration: Security Services*
- "How to Change the Security Attributes of a User" in *Oracle Solaris 11.1 Administration: Security Services*

▼ How to Delete a User

1 **Assume the `root` role**

```
$ su -
Password:
#
```

Note – This method works whether `root` is a user account or a role.

2 **Archive the user's home directory.**

3 **Run one of the following commands:**

- **If the user has a local home directory, delete the user and the home directory.**

  ```
  # userdel -r username
  ```

  `usesrdel` Deletes the account of the specified user.

-r Removes the account from the system.

Because user home directories are now ZFS datasets, the preferred method for removing a local home directory for a deleted user is to specify the -r option with the userdel command.

- **Otherwise, delete the user only.**

  ```
  # userdel username
  ```

  You must manually delete the user's home directory on the remote server.

For a full list of command options, see the userdel(1M) man page.

Next Steps Additional cleanup might be required if the user that you deleted had administrative responsibilities, for example creating cron jobs, or if the user had additional accounts in non-global zones.

▼ How to Add a Group

When an administrator creates a group, the system assigns the solaris.group.assign/*groupname* to that administrator, giving the administrator complete control over that group. If another administrator who has the same authorization creates a group, that administrator has the control over that group. An administrator who has control of one group cannot administer the group of the other administrator. For more information, see the groupadd(1M) and groupmod(1M) man pages.

1 **Assume the root role or an administrator who has the solaris.group.manage authorization.**

See "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **List the existing groups.**

```
# cat /etc/group
```

3 **Create a new group.**

```
$ groupadd -g 18 exadata
```

groupadd Creates a new group definition on the system by adding the appropriate entry to the /etc/group file.

-g Assigns the group ID for the new group.

For more information, see the groupadd(1M) man page.

Example 2–2 Setting Up a Group and User With the groupadd and useradd Commands

The following example shows how to use the groupadd and useradd commands to add the group scutters and the user scutter1 to files on the local system.

```
# groupadd -g 102 scutters
# useradd -u 1003 -g 102 -d /export/home/scutter1 -s /bin/csh \
-c "Scutter 1" -m -k /etc/skel scutter1
64 blocks
```

For more information, see the groupadd(1M) and useradd(1M) man pages.

▼ How to Share Home Directories That Are Created as ZFS File Systems

In this Oracle Solaris release, you can share a ZFS file system by setting the share.nfs property or the share.smb property. Or, you can create a file system share by using the zfs share command. By default, all file systems are unshared.

By default, the pool/export/home dataset is already mounted on /export/home. The useradd command automatically creates per-user datasets as children of this dataset. As an administrator, you can choose to create a new pool for user home directories. The following procedure describes these steps.

For more information about sharing and unsharing file systems, see "Sharing and Unsharing ZFS File Systems" in *Oracle Solaris 11.1 Administration: ZFS File Systems*.

1 **Assume the root role.**

See "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Create a separate pool for the user home directories. For example:**

```
# zpool create users mirror c1t1d0 c1t2d0 mirror c2t1d0 c2t2d0
```

3 **Create a container for the home directories. For example:**

```
# zfs create users/home
```

4 **Set the share properties for the home directory. For example, to create an NFS share and set the share.nfs property for users/home, you would type:**

```
# zfs set share.nfs=on users/home
```

When using this new syntax, each file system contains an "auto share" that is created as soon as the share.nfs property (or the share.smb property) is set to on for that file system. The previous command shares a file system named users/home and all of its children.

5 **Confirm that the descendent file system shares are also published. For example:**

```
# zfs get -r share.nfs users/home
```

The `-r` option displays all of the descendent file systems.

Manually Mounting a User's Home Directory

User accounts that are created as ZFS file systems do not typically need to be manually mounted. With ZFS, file systems are automounted when they are created and then mounted at boot time from the SMF local file system service.

When creating user accounts, make sure home directories are set up as they are in the name service, at `/home/`*username*. Then, make sure that the `auto_home` map indicates the NFS path to the user's home directory. For task-related information, see "Task Overview for Autofs Administration" in *Managing Network File Systems in Oracle Solaris 11.1*.

If you need to manually mount a user's home directory, use the `zfs mount` command. For example:

```
# zfs mount users/home/alice
```

Note – Make sure that the user's home directory is shared. For more information, see "How to Share Home Directories That Are Created as ZFS File Systems" on page 50.

◆ ◆ ◆ CHAPTER 3

Managing User Accounts by Using the User Manager GUI (Tasks)

This chapter provides overview and task-related information for setting up and managing users by using the Oracle Solaris User Manager GUI. You can use the User Manager GUI to perform most of the tasks that can be performed by using the equivalent CLI (`useradd`, `usermod`, `userdel`, and so on). For more information about the User Manager GUI, refer to the online help.

This is a list of the information in this chapter:

- "Introducing the User Manager GUI" on page 53
- "Adding, Modifying, and Deleting Users and Roles by Using the User Manager GUI" on page 57
- "Administering Advanced Settings With the User Manager GUI" on page 60

For overview information about managing user accounts, see Chapter 1, "Managing User Accounts and User Environments (Overview)."

For information about managing user accounts by using the CLI, see Chapter 2, "Managing User Accounts by Using the Command-Line Interface (Tasks)."

Introducing the User Manager GUI

The following information is described in this section:

- "Starting the User Manager GUI" on page 54
- "Organization of the User Manager Panel" on page 54
- "Selecting a Default Name-Service Scope and Type" on page 56
- "Assuming a Role or Changing User Credentials" on page 56

The User Manager GUI is based on the Visual Panels framework and is provided as a Visual Panels interface. The remote management of users and roles is made possible through the Remote Administration Daemon (RAD). The GUI depends on the User/Role Manager RAD module to perform all of its operations. The RAD module works by invoking role-based access control (RBAC) CLIs that perform all of the administrative functions of the GUI.

User authentication and role assumption is provided by the Visual Panels framework itself and is available to all of the panels, including the User Manager panel. The User Manager GUI replaces the Solaris Management Console's User and Roles tool that is supported in Oracle Solaris 10. Although not identical to the Solaris Management Console, the GUI has some of the same functionality. Note that the Solaris Management Console is *not* supported in this release.

The User Manager GUI presents a simple, clear interface that is easy to use. To minimize the possibility of errors, the GUI presents only those choices that are valid, based on the authorizations and rights profiles of the authenticated user or role. The tasks that can be performed with the GUI are the same as the tasks that you can perform by using the CLI, for example, `useradd`, `usermod`, `userdel`, `roleadd`, `groupadd`, and so on. For information about managing users and roles by using the CLI, see Chapter 2, "Managing User Accounts by Using the Command-Line Interface (Tasks)."

The User Manager GUI is delivered by the `pkg:/system/management/visual-panels/panel-usermgr` IPS package.

Starting the User Manager GUI

▼ How to Start the User Manager GUI

1 **Assume the `root` role or log in as a user who is assigned the User Management rights profile.**

See "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 **Start the User Manager GUI by choosing one of the following methods:**

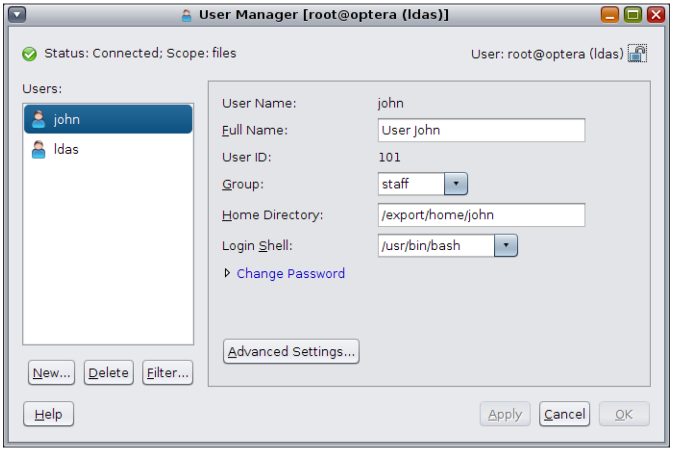
- **Start the User Manager GUI from the desktop by choosing System → Administration → User Manager.**
- **Start the User Manager GUI from the command line as follows:**

```
# vp usermgr &
```

Organization of the User Manager Panel

When you Start the User Manager GUI, the main User Manager panel is displayed. The User Manager panel is used to administer users and roles. On the left side of the panel is a Status field that displays the status of the services that are currently running on the local host. On the right of the panel is a User field. The User field displays the credential that is currently being used by the User Manager GUI. To change credentials, click the Lock button on the far right side of the panel. See “Assuming a Role or Changing User Credentials” on page 56.

In the following figure, the main User Manager panel is displayed.



User Manager [root@optera (ldas)]

Status: Connected; Scope: files

User: root@optera (ldas)

User Name: john

Full Name: User John

User ID: 101

Group: staff

Home Directory: /export/home/john

Login Shell: /usr/bin/bash

The User Manager panel includes the following components:

- Users and Roles list – Contains a list of users from which you can select to administer
- Basic Settings – Displays the basic settings for a user, such as user name and full name

To view or modify information for an existing user, select the user from the list of users that is displayed. After you select a user, that user's information is displayed on the right side of the panel.

The following actions are available to you from within the User Manager panel:

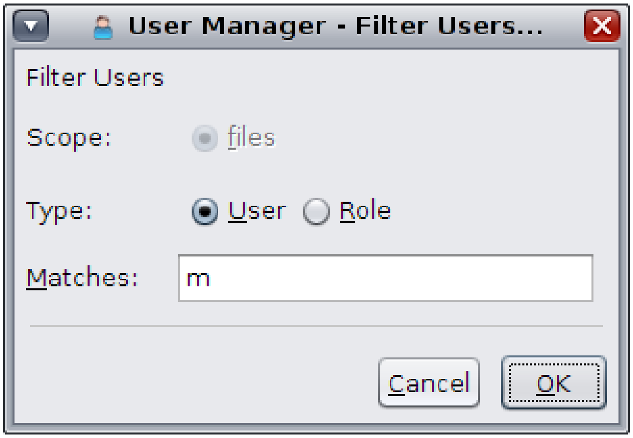
- Create a new user or role. See “How to Add a User or Role With the User Manager GUI” on page 57.
- Delete an existing user or role. See “Deleting a User or Role With the User Manager GUI” on page 60
- Filter a user's information. See “Selecting a Default Name-Service Scope and Type” on page 56.
- Administer advanced settings for an existing user. See “How to Modify a User or Role With the User Manager GUI” on page 59.

Selecting a Default Name-Service Scope and Type

The default name-service scope and type for the User Manager GUI is `files` and `User`. To administer the User Manager GUI within a different scope, for example `ldap` and `roles`, click the Filter button. Clicking the Filter button launches a dialog box that enables you to change the default scope, type, or both.

- Choices for the Scope option are `files` and `ldap`.
- Choices for Type option are `User` and `Role`. Click OK to save the changes.

Click Cancel to cancel the operation.



User Manager - Filter Users...

Filter Users

Scope: ● files

Type: ● User ○ Role

Matches: m

Note – If the system is not configured as an `ldap` client, only the `files` scope is available.

Assuming a Role or Changing User Credentials

A user with the User Management rights profile can create new users, as long as the advanced attributes of the user or role to be created are a subset of those of the user who is performing the administration. If the user who is performing the administration does not have sufficient authorizations, but has an administrative role with sufficient authorizations, the user can assume that role to perform the necessary administration by clicking the Lock button in the main User Manager panel.

▼ How to Change a User's Credentials

1 Start the User Manager GUI.

See "How to Start the User Manager GUI" on page 54.

2 In the main User Manager panel, Click the Lock icon to open a submenu that contains the following options:

- Change Role
- Change User
- Administer New Host
- Clear History

3 Select the Change Role option.

An authentication dialog box is displayed. The authentication dialog box contains a drop-down menu that lists the roles that are available for the specified user.

4 Select the appropriate role, then click Log In to change the role.

After assuming the role, you can perform the required administrative tasks.

Adding, Modifying, and Deleting Users and Roles by Using the User Manager GUI

Adding, modifying, and deleting users by using the User Manger GUI is equivalent to using the `useradd`, `usermod`, and `userdel` commands, respectively. For more information about adding users from the command line, see Chapter 2, "Managing User Accounts by Using the Command-Line Interface (Tasks)."

The following information is described in this section:

- "How to Add a User or Role With the User Manager GUI" on page 57
- "How to Modify a User or Role With the User Manager GUI" on page 59
- "Deleting a User or Role With the User Manager GUI" on page 60

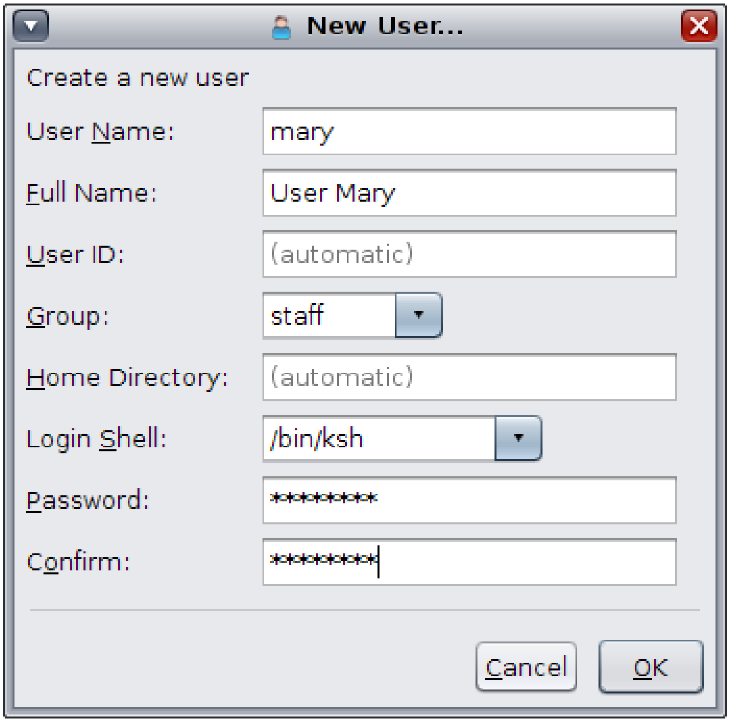
▼ How to Add a User or Role With the User Manager GUI

1 Start the User Manager GUI.

See "How to Start the User Manager GUI" on page 54.

2 **To add a new user or role within the scope of the filter that is currently being used by the GUI, click the New button in the main User Manager panel.**

The New User dialog box is displayed.



New User...

Create a new user

User Name:	mary
Full Name:	User Mary
User ID:	(automatic)
Group:	staff
Home Directory:	(automatic)
Login Shell:	/bin/ksh
Password:	********
Confirm:	********

3 **In the New User dialog box, complete the following fields:**

- User Name
- Full Name
- User ID

  This field is optional. If you don't provide any information, the system automatically assigns a default value.
- Group

  Available choices for the Group field vary depending on your system's configuration.
- Home Directory

  This field is optional. If you don't provide any information, the system automatically assigns a default value.

  If you want the home directory of the user to be automounted, precede the path name with a host name or a local host. For example, `localhost:/export/home/test1`.

- Login Shell

  Choices for the Login Shell field vary, depending on your system's configuration.
- Password

  Assign a temporary password to the user.
- Confirm

  Confirm the temporary password that you assigned to the user.

Note – You must complete all of the fields, with the exception of optional fields.

---

4 **To create the new user or role and add the user or role to the list of users that is displayed in the main User Manager panel, click OK.**

To cancel the operation, click Cancel.

▼ How to Modify a User or Role With the User Manager GUI

1 **Start the User Manager GUI.**

See "How to Start the User Manager GUI" on page 54.

2 **To modify an existing user or role, in the main User Manager panel, select the user or role that you want to modify from the list that is displayed.**

After selecting the user, the right side of the panel is populated with information about the current user.

3 **Modify any or all of the information for the current user or role.**

Note – If a field is modified, an indicator is displayed next to the field that has been modified.

---

4 **Click Apply to save the changes.**

5 **(Optional) Click the Advanced Settings button to modify additional security attributes for the user or role. See "Administering Advanced Settings With the User Manager GUI" on page 60.**

6 **Click OK to save the changes and close the User Manager panel.**

Click Cancel to discard any unsaved changes and close the panel.

Deleting a User or Role With the User Manager GUI

To delete a user or role within the scope of the filter that is currently being used by the User Manager GUI, select the user or role in the main User Manager panel, then click the Delete button. To save the changes, click OK when the confirmation dialog box is displayed. To cancel the operation, click Cancel.

Administering Advanced Settings With the User Manager GUI

The following information is described in this section:

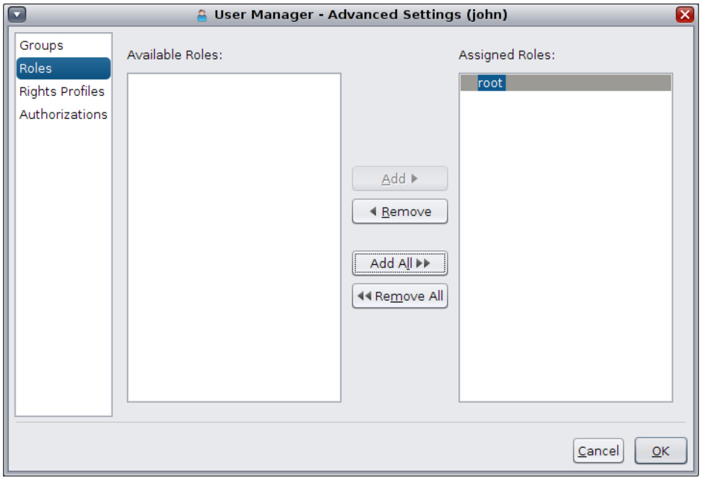
- "Administering Groups With the User Manager GUI" on page 61
- "Administering Roles With the User Manager GUI" on page 62
- "Administering Rights Profiles With the User Manager GUI" on page 63
- "Administering Authorizations With the User Manager GUI" on page 65

Use the Advanced Settings dialog box of the User Manager GUI to assign additional security attributes to a user, for example, rights profiles, roles, and authorizations.

For an overview of the security features that are supported in Oracle Solaris, see Part I, "Security Overview," in *Oracle Solaris 11.1 Administration: Security Services*. For a detailed explanation of how role-based access control (RBAC) works in this release, see Part III, "Roles, Rights Profiles, and Privileges," in *Oracle Solaris 11.1 Administration: Security Services*.

To administer advanced attributes for a user or role, select the user or role in the main User Manager panel, then click the Advanced Settings button. The Advanced Settings panel for the current user or role is displayed. The current user's name is displayed in parentheses at the top of the panel.

The following figure shows the Advanced Settings panel, with the Roles security attribute of the user john selected.



The following security attributes can be administered in the Advanced Settings panel:

- Groups
- Roles
- Rights Profiles
- Authorizations

Administering Groups With the User Manager GUI

Groups are administered in the main User Manager dialog box of the User Manager GUI by clicking the Advanced Settings button.

How to Administer Groups

1. **Start the User Manager GUI.**

   See "How to Start the User Manager GUI" on page 54.

2. **Select a user in the main User Manager panel, then click the Advanced Settings button.**

   The Advanced Settings panel is displayed.

3. **Click the Groups attribute on the left side of the panel.**

   A list of the available groups and a list of the groups that the current user belongs to are displayed.

   - **To assign a group (or multiple groups) to a user, select the group (or groups) from the Available Groups list, then click Add.**

     The added group is displayed in the Assigned Groups list.

- **To remove a group from the Assigned Groups list, select the group (or groups) from the list, then click Remove.**
- **To add or remove all of the groups for the current user, click the Add All or Remove All button.**

4 **Click OK to save the settings.**

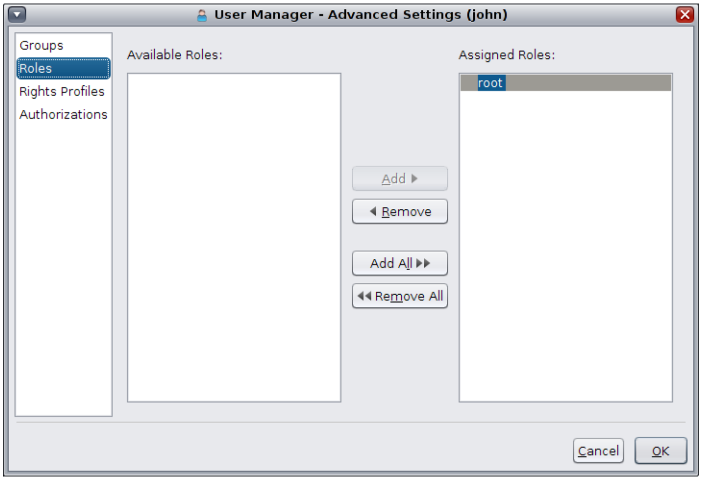
The changes are not applied until you click Apply or OK in the main User Manager panel.

Administering Roles With the User Manager GUI

Roles are administered in the main User Manager dialog box of the User Manager GUI by clicking the Advanced Settings button.

Note – The Roles attribute is available only for a user, not for a role, because roles can only be assigned to users.

The following figure shows the Advanced Settings panel, with the Roles security attribute of the user john selected.



User Manager - Advanced Settings (john)

Available Roles:

Assigned Roles: root

▼ How to Administer Roles With the User Manager GUI

1. **Start the User Manager GUI.**

   See "How to Start the User Manager GUI" on page 54.

2. **Select a user in the main User Manager panel, then click the Advanced Settings button.**

   The Advanced Settings panel is displayed.

3. **Click the Roles attribute on the left side of the panel.**

   A list of the available roles and a list of the roles that are assigned to the current user are displayed.

   - **To assign a role (or multiple roles) to a user, select the role (or roles) from the Available Roles list, then click Add.**

     The added role is displayed in the Assigned Roles list.

   - **To remove a role from the Assigned Roles list, select the role (or roles) from the list, then click Remove.**

   - **To add or remove all of the roles for the current user, click the Add All or Remove All button.**

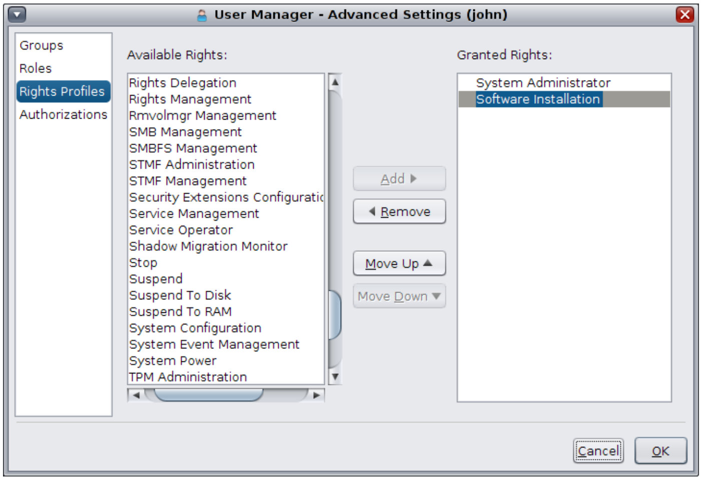
4. **Click OK to save the settings.**

   The changes are not applied until you click Apply or OK in the main User Manager panel.

Administering Rights Profiles With the User Manager GUI

Rights profiles are administered in the main User Manager dialog box of the User Manager GUI by clicking the Advanced Settings button.

The following figure shows the Advanced Settings panel, with the Rights Profile security attribute of the user john selected.



User Manager - Advanced Settings (john)

Available Rights:

- Rights Delegation
- Rights Management
- Rmvolmgr Management
- SMB Management
- SMBFS Management
- STMF Administration
- STMF Management
- Security Extensions Configuratic
- Service Management
- Service Operator
- Shadow Migration Monitor
- Stop
- Suspend
- Suspend To Disk
- Suspend To RAM
- System Configuration
- System Event Management
- System Power
- TPM Administration

Granted Rights:

- System Administrator
- Software Installation

Note – The assignment of rights profiles has an order precedence. Use the Move Up and Move Down buttons to change the order of the rights profiles that are granted to the current user, as desired.

▼ How to Administer Rights Profiles With the User Manager GUI

1 Start the User Manager GUI.

See "How to Start the User Manager GUI" on page 54.

2 Select a user in the main User Manager panel, then click the Advanced Settings button.

The Advanced Settings panel is displayed.

3 Click the Rights Profile attribute on the left side of the panel.

A list of the available rights profiles and a list of the rights profiles that are granted to the current user are displayed.

- **To assign a rights profile (or multiple rights profiles) to a user, select the rights profile (or rights profiles) from the Available Rights Profiles list, then click Add.**

  The added rights profile is displayed in the Granted Rights Profiles list.

- **To remove a rights profile from the Granted Rights Profiles list, select the rights profile (or rights profiles) from the list, then click Remove.**

- **To add or remove all rights profiles for the current user, click the Add All or Remove All button.**

4 **Click OK to save the settings.**

The changes are not applied until you click Apply or OK in the main User Manager panel.

Administering Authorizations With the User Manager GUI

A user generally is granted authorizations indirectly through a rights profile. Authorization settings can be used to grant a specific authorization to a user or role. Some authorizations might have additional attributes, such as an object name. For example, when an administrator creates the group `games`, the administrator is granted an implicit authorization: `solaris.group.manage/games`. The object names are then displayed in the Granted Authorizations list.

▼ How to Administer Authorizations With the User Manager GUI

1 **Start the User Manager GUI.**

See "How to Start the User Manager GUI" on page 54.

2 **Select a user in the main User Manager panel, then click the Advanced Settings button.**

The Advanced Settings panel is displayed.

3 **Click the Authorization attribute on the left side of the panel.**

A list of the available authorizations and a list of the authorizations that are granted to the current user are displayed.

- **To assign an authorization (or multiple authorizations) to a user, select the authorization (or authorizations) from the Available Authorizations list, then click Add.**

  The added authorization is displayed in the Granted Authorizations list.

- **To remove an authorization from the Granted Authorizations list, select the authorization (or authorizations) from the list, then click Remove.**

- **To add or remove all authorizations for the current user, click the Add All or Remove All button.**

4 **Click OK to save the settings.**

The changes are not applied until you click Apply or OK in the main User Manager panel.

Index