Managing Boot Environments With Oracle® Solaris 11 Express

Part No: 820–6565–12
November 2010

110425@25097

Contents

CHAPTER 1

Introduction to Boot Environments

A boot environment is a bootable instance of the Oracle Solaris operating system image plus any other application software packages installed into that image. System administrators can maintain multiple boot environments on their systems, and each boot environment can have different software versions installed.

Upon the initial installation of the Oracle Solaris release onto a system, a boot environment is created. Use the `beadm`(1M) utility to administer additional boot environments on your system.

Why Use Multiple Boot Environments?

With multiple boot environments, the process of updating software becomes a low risk operation because system administrators can create backup boot environments before making any software updates to your system. If needed, they have the option of booting a backup boot environment.

Here are some specific examples where having more than one Oracle Solaris boot environment, and managing them with the `beadm` utility, is useful:

- If you are modifying a boot environment, you can take a snapshot of that environment at any stage during modifications by using the `beadm create` command. A snapshot is a read-only image of a dataset or boot environment at a given point in time.

  You can create custom names for each snapshot that identify when or why the snapshot was created. For example, if you are doing monthly upgrades to your boot environment, you can capture snapshots for each monthly upgrade. You can use the `beadm list -s` command to view the available snapshots for a boot environment.

  A snapshot is not bootable. But, you can create a boot environment, based on that snapshot, by using the `-e` option for the `beadm create` command. Then you can use the `beadm activate` command to specify that this boot environment will become the default boot environment on the next reboot.

- You can maintain more than one boot environment on your system, and perform various updates on each of them as needed. For example, you can clone a boot environment by using the `beadm create` command. A clone is a bootable copy of a boot environment. Then, you can install, test, and update different software packages on the original boot environment and on its clone.

  Although only one boot environment can be active at a time, you can mount an inactive boot environment by using the `beadm mount` command. Then, you could use the `pkg install` *packagename* with the `-R` option to update specific packages on that environment.
- When you use either the Package Manager tool, the Update Manager tool, or the `pkg` command to install or update packages in your active Oracle Solaris boot environment, this process may create a clone of that boot environment. If a clone is created, packages are installed or updated in the clone rather than in the original boot environment. After successfully completing the changes, the new clone is activated. Then, the clone will become the new default boot environment on the next reboot. The original boot environment remains on the GRUB menu for x86 systems, or the boot menu for SPARC systems, as an alternate selection.

  You can use the `beadm list` command to see a list of all the boot environments on the system, including the backup boot environment that still has its original, unchanged software. If you are not satisfied with the changes made to the environment, you can use the `beadm activate` command to specify that the backup will become the default boot environment on the next reboot.

  For more information, see "Images and Boot Environments" in *Oracle Solaris 11 Express Image Packaging System Guide* .

Package Manager and Update Manager GUIs

You can manage the boot environments on your system either by using the `beadm` command or by using the Package Manager or the Update Manager.

The Package Manager and Update Manager are graphical user interfaces that enable you to install, update, and manage packages on your installed system. The Package Manager and Update Manager are available on the menu bar on the desktop of the Oracle Solaris operating system. On the desktop menu bar, go to System>Administration and select either the Package Manager or Update Manager.

If you use the Package Manager or Update Manager to install or update the packages on your system, a clone of the active boot environment may be created with the changes made in that clone. If a clone is created, that clone becomes the active boot environment on reboot. You can use the `beadm list` command to see a list of all the boot environments on the system. You have the option to reactivate the original boot environment.

You can use the Package Manager and Update Manager to manage your boot environments as follows.

- You can delete old and unused boot environments, in order to make the disk space available.
- You can change the default boot environment on your system.
- You can activate a boot environment.

For instructions, see the online help that is available in the Package Manager and Update Manager. See, also, Chapter 2, "IPS Graphical User Interfaces," in *Oracle Solaris 11 Express Image Packaging System Guide* .

The Package Manager and Update Manager, however, do not provide the full range of options for managing your boot environments that is available by using the `beadm` command as described in the following sections.

Features of `beadm` Utility

The `beadm` utility has the following features:

- The `beadm` utility aggregates all datasets in a boot environment and performs actions on the entire boot environment at once. You no longer need to perform ZFS commands to modify each dataset individually.
- The `beadm` utility manages the dataset structures within boot environments. For example, when the `beadm` utility clones a boot environment that has shared datasets, the utility automatically recognizes and manages those shared datasets for the new boot environment.
- The `beadm` utility enables you to perform administrative tasks on your boot environments. These tasks can be performed without upgrading your system.
- The `beadm` utility automatically manages and updates the GRUB menu for x86 systems, or the boot menu for SPARC systems. For example, when you use the `beadm` utility to create a new boot environment, that environment is automatically added to the GRUB menu or boot menu.

The `beadm` utility enables you to perform the following tasks:

- Create a new boot environment based on the active boot environment
- Create a new boot environment based on an inactive boot environment
- Create a snapshot of an existing boot environment
- Create a new boot environment based on an existing snapshot
- Create a new boot environment and add a custom title to the x86 GRUB menu or the SPARC boot menu
- Activate an existing, inactive boot environment
- Mount a boot environment

- Unmount a boot environment
- Destroy a boot environment
- Destroy a snapshot of a boot environment
- Rename an existing, inactive boot environment
- Display information about your boot environment snapshots and datasets

beadm Utility Requirements

For the Oracle Solaris 11 Express release, in order to use the `beadm` utility to manage your boot environments, your system must meet the following requirements.

TABLE 1–1 System Requirements

Requirement	Description
Oracle Solaris operating system	Install the Oracle Solaris release on your system. For instructions, see Chapter 3, “Installing Oracle Solaris 11 Express,” in *Getting Started With Oracle Solaris 11 Express* . For instructions about upgrading an OpenSolaris operating system to the Oracle Solaris 11 Express release, see *Oracle Solaris 11 Express Release Notes* .
ZFS file systems	The `beadm` utility is designed for ZFS file systems.

Limitations in Current Release

Note the following limitations.

- The installation of the Oracle Solaris OS supports one pool on one disk. After installation, you can attach another disk to the root pool and set up another pool.
- The `beadm` utility is implemented with ZFS support only.
- The `beadm` utility manages boot environments and snapshots that were created by the `beadm` utility. If a user manually creates the datasets for a boot environment by using the `zfs(1M)` command, the `beadm` utility should recognize the boot environment, but management of this boot environment might be undefined.

CHAPTER 2

Using beadm Utility (Tasks)

You can use the beadm utility to create and manage snapshots and clones of your boot environments.

Note the following distinctions relevant to boot environment administration:

- A *snapshot* is a read-only image of a dataset or boot environment at a given point in time. A snapshot is not bootable.
- A *boot environment* is a bootable Oracle Solaris environment, consisting of a root dataset and, optionally, other datasets mounted underneath it. Exactly one boot environment can be active at a time.
- A *clone* of a boot environment is created by copying another boot environment. A clone is bootable.

  Note – A clone of the boot environment includes everything hierarchically under the main root dataset of the original boot environment. Shared datasets are not under the root dataset and are not cloned. Instead, the boot environment accesses the original, shared dataset.

- A *dataset* is a generic name for ZFS entities such as clones, file systems, or snapshots. In the context of boot environment administration, the dataset more specifically refers to the file system specifications for a particular boot environment or snapshot.
- *Shared datasets* are user-defined directories, such as `/export`, that contain the same mount point in both the active and inactive boot environments. Shared datasets are located outside the root dataset area of each boot environment.
- A boot environment's *critical datasets* are included within the root dataset area for that environment.

Note – You must assume the root role on your system to use the beadm utility.

For detailed instructions about the beadm utility, see the beadm(1M) man page. See also Chapter 4, "Appendix: beadm Reference."

Listing Existing Boot Environments and Snapshots

All snapshots, boot environments, and datasets that were created by the beadm command can be displayed with the beadm list subcommand.

Snapshots and boot environments can be created by the beadm command. Snapshots and boot environments can also be created by other utilities. For example, the pkg command may automatically create a clone of a boot environment when you install or update packages using that command. The beadm list command output also displays boot environments that are created by the pkg command.

▼ How to Display Information About Your Boot Environments, Snapshots, and Datasets

1 **Choose the information that you want to view, as described in this step.**

- **Choose an option for the beadm list command from the following list:**

  -a – Lists all available information about the boot environment. This option includes subordinate datasets and snapshots.

  -d – Lists information about a boot environment's datasets.

  -s – Lists information about a boot environment's snapshots.

  -H – Omits the header information from the display. Choosing this option results in a display that can be more easily parsed for scripts or other programs.

- **You can also choose whether to specify a boot environment name in the beadm list command.**

  By default, if you do not include a boot environment name in the command, the results include all boot environments.

  But, if you include a boot environment name in the beadm list command, the results include only information about that boot environment.

2 **Type the beadm list command with preferred options.**

For example, include the -a option and specify the BE1 environment as follows:

```
$ beadm list -a BE1
```

3 **Review the information that is displayed about the available boot environments.**

Example 2–1 Viewing Boot Environment Specifications

This example includes the `-a` option and specifies the BE2 environment as follows:

```
$ beadm list -a BE2
```

Sample results are displayed. The BE2 environment will be active on reboot, as indicated by the R in the Active column.

```
BE/Dataset             Active Mountpoint Space  Policy Created
----------             ------ ---------- -----  ------ -------
BE2 rpool/ROOT/BE2     R      /mnt       89.67M static 2008-09-17 20:17
    rpool/ROOT/BE2/var -      /mnt/var   0      static 2008-09-17 20:17
```

Two datasets are included in BE2, the dataset, `rpool/ROOT/BE2`, and the dataset, `rpool/ROOT/BE2/var`. The values for the Active column are as follows:

- *R* – Active on reboot
- *N* – Now active
- "-" – Inactive

Example 2–2 Viewing Snapshot Specifications

This example includes the `-s` option, which displays information for any snapshots that exist on the current image. The status of those snapshots is the same list that the `zfs`(1M) command displays.

```
$ beadm list -s test-2
```

Sample results are displayed.

```
BE/Snapshot     Space Policy Created
----------- ----- ------ -------
test-2
test-2@2010-04-12-22:29:27 264.02M static 2010-04-12 16:29
test-2@2010-06-02-20:28:51 32.50M static 2010-06-02 14:28
test-2@2010-06-03-16:51:01 16.66M static 2010-06-03 10:51
test-2@2010-07-13-22:01:56 25.93M static 2010-07-13 16:01
test-2@2010-07-21-17:15:15 26.00M static 2010-07-21 11:15
test-2@2010-07-25-19:07:03 13.75M static 2010-07-25 13:07
test-2@2010-07-25-20:33:41 12.32M static 2010-07-25 14:33
test-2@2010-07-25-20:41:23 30.60M static 2010-07-25 14:41
test-2@2010-08-06-15:53:15 8.92M static 2010-08-06 09:53
test-2@2010-08-06-16:00:37 8.92M static 2010-08-06 10:00
test-2@2010-08-09-16:06:11 193.72M static 2010-08-09 10:06
test-2@2010-08-09-20:28:59 102.69M static 2010-08-09 14:28
test-2@install 205.10M static 2010-03-16 19:04
```

In the sample results, each snapshot title includes a timestamp, indicating when that snapshot was taken.

Creating a Boot Environment

If you want to create a backup of an existing boot environment, for example, prior to modifying the original boot environment, you can use the `beadm` command to create and mount a new boot environment that is a clone of your active boot environment. This clone is inactive and displays as an alternate in the GRUB menu for x86 systems or the boot menu for SPARC systems on reboot. The SPARC boot menu is located at `/rpool/boot/menu.lst`.

Note – When you create a new boot environment, the process first automatically takes a snapshot of the original boot environment, then the process makes a new clone that is derived from this snapshot. Both the new snapshot and the new clone are retained by the system.

▼ How to Create a Boot Environment

1 **At the command line, type the following:**

```
$ beadm create beName
```

`beName` is a variable for the name of the new boot environment. This new boot environment is inactive.

Note – `beadm create` does not create a partial boot environment. The command either successfully creates a full boot environment, or the command fails.

---

2 **(Optional) You can use the `beadm mount` command to mount the new boot environment as follows:**

```
$ beadm mount beName mountpoint
```

Note – If the directory for the mount point does not exist, the `beadm` utility creates the directory, then mounts the boot environment on that directory.

If the boot environment is already mounted, the `beadm mount` command fails and does not remount the boot environment at the newly specified location.

The boot environment is mounted, but remains inactive. You can upgrade a mounted, inactive boot environment.

Note – Unmount the boot environment before rebooting.

---

3 **(Optional) If you want to boot to this new boot environment, use the following command to activate the boot environment:**

```
$ beadm activate beName
```

beName is a variable for the name of the boot environment to be activated.

On reboot, the newly active boot environment is displayed as the default selection in the x86 GRUB menu or the SPARC boot menu.

Note – The GRUB menu or boot menu always displays the most recently activated boot environment as the default.

Example 2–3

Creating a New, Cloned Boot Environment with Datasets (BE2)

The following example illustrates the datasets in a newly created boot environment.

Type the following command.

```
$ beadm create BE2
```

The original boot environment in this example is BE1. The new boot environment, BE2, contains separate datasets cloned from BE1. If BE1 contains separate datasets for traditional file systems, such as `/var` or `/opt`, then those datasets are also cloned.

```
rpool/ROOT/BE1
rpool/ROOT/BE1/usr
rpool/ROOT/BE1/var
rpool/ROOT/BE1/opt

rpool/ROOT/BE2
rpool/ROOT/BE2/usr
rpool/ROOT/BE2/var
rpool/ROOT/BE2/opt
```

`rpool` in this example is the name of the storage pool. The pool already exists on the system, previously set up by the initial installation or upgrade. ROOT is a special dataset that was also created previously by the initial installation or upgrade. ROOT is reserved exclusively for use by boot environment roots.

Example 2–4

Creating a New Boot Environment With Shared Datasets

The following example illustrates a newly-created boot environment that has shared datasets.

Shared datasets are user-defined directories, such as `/export`, that contain the same mount point in both the active and inactive boot environments. Shared datasets are located outside the root dataset area of each boot environment.

Type the following command.

```
$ beadm create BE2
```

The shared datasets, `rpool/export` and `rpool/export/home`, are not cloned when the boot environment is cloned. The shared datasets are located outside the `rpool/ROOT/<BE>` datasets and are referenced at their original locations by the cloned boot environment.

The original boot environment, BE1 and datasets are as follows:

```
rpool/ROOT/BE1
rpool/ROOT/BE1/usr
rpool/ROOT/BE1/var
rpool/ROOT/BE1/opt
rpool/export
rpool/export/home
```

The cloned boot environment, BE2, has new root datasets, but the original shared datasets, `rpool/export` and `rpool/export/home`, are unchanged.

```
rpool/ROOT/BE2
rpool/ROOT/BE2/usr
rpool/ROOT/BE2/var
rpool/ROOT/BE2/opt
rpool/export
rpool/export/home
```

▼ How to Create a Boot Environment From an Inactive Boot Environment

By default, a new boot environment is created from the active boot environment. However, you can use the `beadm create` command to create a new boot environment that is based on an existing, inactive boot environment.

1 **Become superuser or assume an equivalent role.**

2 **To create a new, inactive BE from an existing, inactive boot environment, type the following command:**

```
$ beadm create -e non-activeBeName beName
```

non-activeBeName is a variable for the existing inactive BE that is cloned by this command.

beName is the variable for the new BE name.

By default, the new boot environment is inactive.

Taking a Snapshot of a Boot Environment

You can manually create a snapshot of an existing boot environment for your reference. This snapshot is a read-only image of a dataset or boot environment at a given point in time. You can create a custom name for the snapshot that indicates when the snapshot was created or what it contains.

▼ How to Create a Snapshot of a Boot Environment

● **Use the following command to create a snapshot.**

```
$ beadm create BEname@snapshotdescription
```

Replace the variable, *BEname@snapshotdescription*, with a custom name for your snapshot. The custom name must use the format *BEname@snapshotdescription*, where *BEname* is the name of an existing boot environment that you want to make a snapshot from. Type a custom *snapshotdescription* to identify the date or purpose of the snapshot.

Sample snapshot names include:

- `BE1@0312200.12:15pm`
- `BE2@backup`
- `BE1@march132008`

Note – Unless you use the `beadm create` command to assign a custom title to a snapshot, titles for snapshots automatically include a timestamp that indicates when the snapshot was taken.

Using an Existing Snapshot

A snapshot of a boot environment is not bootable. However, you can create a new boot environment from an existing snapshot. Then you can activate and boot that new boot environment.

▼ How to Create a Boot Environment From an Existing Snapshot

● **Use the following command to create a new boot environment from a snapshot.**

```
$ beadm create -e BEname@snapshotdescription beName
```

Replace the variable, *BEname@snapshotdescription*, with the name of an existing snapshot. As described in the previous task, snapshot names use the format *BEname@snapshotdescription*, where *BEname* is the name of an existing boot environment, and *snapshotdescription* is a description of a specific snapshot made from the existing boot environment.

Replace the variable, *BEname*, with a custom name for your new boot environment.

See the following example.

```
$ beadm create -e BE1@now BE2
```

This command creates a new boot environment, named BE2, from the existing snapshot named BE1@now. You can active this boot environment by using the next procedure.

Changing the Default Boot Environment

You can change an inactive boot environment into an active boot environment. Only one boot environment can be active at a time. The newly activated boot environment becomes the default environment upon reboot.

▼ How to Activate an Existing Boot Environment

1 **Use the following command to activate an existing, inactive boot environment:**

```
$ beadm activate beName
```

beName is a variable for the name of the boot environment to be activated.

Note the following specifications.

- `beadm activate` *beName* activates a boot environment by setting the bootable pool property, `bootfs`, to the value of the ROOT dataset of the boot environment that is being activated.
- `beadm activate` sets the newly activated boot environment as the default in the `menu.lst` file.

Note – When a boot environment is created, whether it is active or inactive, an entry is created for the boot environment on the x86 GRUB menu or the SPARC boot menu. The default boot environment is the last boot environment that was activated.

---

2 **Reboot.**

The newly activated boot environment is now the default on the x86 GRUB menu or SPARC boot menu.

Note – If the boot environment fails to boot, reboot and select the previous boot environment from the GRUB menu or the boot menu.

Mounting and Updating an Inactive Boot Environment

If you want to update packages on an existing, inactive boot environment, you can mount that environment and, optionally, update packages on it.

▼ How to Mount a Boot Environment

1 **You can use the `beadm mount` command to mount a boot environment as follows:**

```
$ beadm mount beName mountpoint
```

Note – If the directory for the mount point does not exist, the `beadm` utility creates the directory, then mounts the boot environment on that directory.

If the boot environment is already mounted, the `beadm mount` command fails and does not remount the boot environment at the newly specified location.

The boot environment is mounted but remains inactive.

2 **(Optional) You can update packages on the boot environment by using the `pkg` command.**

For example, you can use the `pkg install` command with the `-R` option to update specific packages on the boot environment.

```
$ pkg -R /mnt install packagename
```

Where */mnt* is the mount point for the boot environment.

Note – Unmount the boot environment before rebooting.

For further information, see the `pkg`(1) man page.

Unmounting Boot Environments

You can use the `beadm` command to unmount an existing boot environment.

Note – You cannot unmount the boot environment that is currently booted.

How to Unmount an Existing Boot Environment

- **To unmount the boot environment, type the following:**

```
$ beadm unmount beName
```

Destroying a Boot Environment

If you want to make more room available on your system, you can use the `beadm` command to destroy an existing boot environment.

Note the following specifications:

- You cannot destroy the boot environment that is currently booted.
- The `beadm destroy` command automatically removes the destroyed boot environment's entry from the x86 GRUB menu or the SPARC boot menu.
- The `beadm destroy` command destroys only the critical or nonshared datasets of the boot environment. Shared datasets are located outside of the boot environment root dataset area and are not affected when a boot environment is destroyed.

  See the following example, where BE1 and BE2 share the `rpool/export` and `rpool/export/home` datasets. The datasets include the following:

  ```
  rpool/ROOT/BE1
  rpool/ROOT/BE2
  rpool/export
  rpool/export/home
  ```

  Destroy BE2 by using the following command:

  ```
  beadm destroy BE2
  ```

  The shared datasets, `rpool/export` and `rpool/export/home`, are not destroyed when the boot environment BE2 is destroyed. The following datasets remain.

  ```
  rpool/ROOT/BE1
  rpool/export
  rpool/export/home
  ```

How to Destroy an Existing Boot Environment

- **To destroy a boot environment, type the following:**

```
$ beadm destroy beName
```

Note – The `beadm destroy` command asks for confirmation before destroying the boot environment. Add the `-F` option to the `beadm destroy` command to force destruction of the boot environment without a confirmation request. Add the `-f` option to force destruction of the boot environment even if its mounted.

Creating Custom Names for Boot Environments

The `beadm rename` command enables you to rename an existing boot environment. This command enables you to create a custom name for a boot environment, a name that is more meaningful for your particular situation. For example, you could rename boot environments to specify how you customized that environment. The boot environment's dataset name is also changed to conform with the new boot environment name.

Note – An active boot environment cannot be renamed. Only an inactive boot environment can be renamed. And, you cannot rename the boot environment that is currently booted. If you want to rename the active boot environment, first, make a different boot environment active and boot that environment. Then, you can rename the inactive boot environment.

▼ How to Rename a Boot Environment

● **Use the following command to rename the boot environment:**

```
$ beadm rename beName newBeName
```

Note – If the new name is already in use, the `beadm rename` command fails.

◆ ◆ ◆ CHAPTER 3

beadm Zones Support

Zones partitioning technology is used to virtualize operating system services and provide an isolated and secure environment for running applications. Each Oracle Solaris system is a global zone. Within a global zone, specific non-global zones can be created.

For information about creating and administering non-global zones on your system, see Part II, "Zones," in *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones* .

The beadm utility includes support for non-global zones.

Zones Support Limitations

Note the following limitations of support for non-global zones in the beadm utility and in related processes:

- When you use the pkg command, the command only upgrades ipkg branded zones.
- The beadm utility is not supported inside a non-global zone.
- Non-global zone support is limited to ZFS support. Zones are not supported unless they are on ZFS.
- Zones are not supported in the rpool/ROOT namespace. Non-global zones are cloned or copied only when the original zone is within the shared area for the global zone, for example, within rpool/export or within rpool/zones.
- Although the beadm utility affects the non-global zones on your system, the beadm utility does not display zones information. Use the zoneadm utility to view changes in the zones in your boot environment. For example, use the zoneadm list command to view a list of all current zones on the system.

  For further information, see the zoneadm(1M) man page.

Zones Support Specifications

The beadm command impacts the non-global zones in your boot environments as follows.

beadm create — When you clone a boot environment by using the beadm create command, all supported zones in that boot environment are copied into the new boot environment.

beadm destroy — When you destroy an inactive boot environment, the zones that belong to that boot environment are also destroyed.

beadm mount — When you mount a boot environment, the supported zones in that environment are mounted relative to the mount points for the environment.

beadm unmount — When you unmount a boot environment, the zones in that environment are also unmounted. All mount points are returned to their states prior to being mounted.

beadm rename — When you rename a boot environment, that change does not impact the names of the zones or the names of the datasets that are used for those zones in that boot environment. The change does not impact the relationships between the zones and their related boot environments.

Zones Naming Conventions

The beadm utility automatically handles all zones naming tasks related to the beadm processes. Just for your reference, this section describes the zones naming conventions that are automatically used by the beadm utility.

A zone root dataset name is stated in the following format:

```
<zonepath dataset>/ROOT/<zone root>
```

For example:

```
rpool/zones/zone1/ROOT
```

When a zone is copied from one boot environment to another boot environment, only the datasets that are under the zone's root dataset are copied.

A dataset can be shared between zone boot environments. A shared dataset is identified by using the following format:

```
<zonepath dataset>/export
```

For example:

```
rpool/zones/zone1/export
```

A shared dataset must be explicitly added during zones configuration. A shared dataset is not cloned when the zone dataset is cloned.

EXAMPLE 3–1 Non-Global Zones Example

This example shows the zones impact of the `beadm create` command when cloning a boot environment that contains two non-global zones.

- The original boot environment is named `opensolaris` with the dataset at `rpool/ROOT/opensolaris`.
- A zone named `z1` exists that has the dataset, `rpool/zones/z1` as its zonepath. The boot environment has a copy of the dataset at `rpool/zones/z1/ROOT/zbe`.

Perform the following command to clone the boot environment and name the new boot environment `opensolaris-1`:

```
beadm create opensolaris-1
```

The root dataset for the new boot environment is at `rpool/ROOT/opensolaris-1`. The copy of zone `z1` for the new boot environment has its root dataset at `rpool/zones/z1/ROOT/zbe-1`.

Note – ZFS properties for each dataset track the relationships between original and cloned datasets, and track the relationships between non-global zones and their parent boot environments.

Additional Resources

For further information about the Oracle Solaris operating system, see the following references.

TABLE 3–1 Oracle Solaris Documentation

Document	Description
Getting Started With Oracle Solaris 11 Express	Instructions for installing and upgrading the Oracle Solaris operating system
Oracle Solaris 11 Express Image Packaging System Guide	Instructions for using the Package Manager and the `pkg` command to install and update packages on your Oracle Solaris system.

◆ ◆ ◆ CHAPTER 4

Appendix: beadm Reference

The beadm command, /usr/sbin/beadm, enables users to manage their boot environments. The following table summarizes the subcommands and options for the beadm utility.

beadm Command Reference

TABLE 4–1 beadm Command

Subcommands	Description
beadm	Displays command usage.
beadm activate	Makes beName the active boot environment on next reboot. Syntax: beadm activate beName

TABLE 4–1 beadm Command *(Continued)*

Subcommands	Description
`beadm create`	Creates a new boot environment name, beName. Unless the `-e` option is provided, the new boot environment is created as a clone of the currently running boot environment. Syntax: `beadm create [-a] [-d description] [-e non-activeBeName | beName@snapshot] [-o property=value] beName` `-a` – Activate the newly created boot environment upon creation. The default is to not activate the newly created boot environment. ■ `-d description` – Use this description as the title in the x86 GRUB menu or the SPARC boot menu that describes this new boot environment. If this option is not used, beName is used for the title. `-e non-activeBeName` – Create a new boot environment from an existing, inactive boot environment. The default is to create the boot environment from the active boot environment. `-e beName@snapshot` – Create a new boot environment from an existing snapshot of the boot environment named beName. `-o property=value` – Create the datasets for new boot environment with specific ZFS properties. Multiple `-o` options can be specified. See `zfs(1M)` for more information on the `-o` option. `beName` – Name of the boot environment to create. `beName@snapshot` – Name of an existing snapshot of the boot environment named beName.
`beadm create beName@snapshot`	Creates a snapshot of the existing boot environment named beName. Syntax: `beadm create beName@snapshot` `beName@snapshot` – The snapshot name must use the format, `beName@snapshotdescription`, where `beName` is the name of an existing boot environment that you want to make a snapshot from. Enter a custom `snapshotdescription` to identify the date or purpose of the snapshot.
`beadm destroy`	Destroys the boot environment named `beName` or destroys an existing snapshot, `beName@snapshot`, of a boot environment. Prompts for confirmation before destroying the boot environment. Syntax: `beadm destroy [-fF] beName | beName@snapshot` `-f` – Forces destruction of the boot environment even if it is mounted. `-F` – Forces destruction of the boot environment without prompting for confirmation.

TABLE 4–1 beadm Command *(Continued)*

Subcommands	Description

beadm list	Lists information about the existing boot environment, which is beName, or lists information for all boot environments if beName is not provided. The default is to list boot environments without any additional information.
	Syntax: `beadm list [-a | [-ds] [-H] [beName]`
	`-a` – Lists all available information about the boot environment. This information includes subordinate datasets and snapshots.
	`-d` – Lists information about all subordinate datasets hat belong to the boot environment.
	`-s` – Lists information about the snapshots of the boot environment.
	`-H` – Prevents listing header information. Each field in the output is separated by a semicolon. The example is displayed without a header:
	`BE2:no:yes:mounted:/pool1/BE/BE2:6.2G;;;`
	`Without header information, the display information is identified by the following delimiters:`
	■ `;` – Delimits boot environments, datasets, zones, and snapshots.
	■ `:` – Delimits attributes for boot environments, datasets, zones, and snapshots.
	■ `,` – Delimits multiple datasets, zones, and snapshots.
	■ Multiple boot environments are delimited with a return, a blank line.
	Note – The `-p` option can be combined with the other options.
	beName – If beName is not provided, the command results display information for all boot environments.

beadm mount	Mounts a boot environment named beName at mount point. The mount point must be an already existing, empty directory. Syntax: `beadm mount beName mountpoint` **Note –** Unmount the boot environment before rebooting.
beadm rename	Renames the boot environment named beName to newBeName. Syntax: `beadm rename beName newBeName`
beadm unmount	Unmounts the boot environment named beName. Syntax: `beadm unmount [-f] beName` `-f` – Forcefully unmounts the boot environment even if it is currently busy.

For detailed instructions about the beadm utility, see the beadm(1M) man page.