Creating and Administering Oracle® Solaris 11.1 Boot Environments

Part No: E29052–01
October 2012

ORACLE®

121203@25097

Contents

Preface 5

1 Introduction to Managing Boot Environments 7
About Boot Environments and Datasets 7
Advantages to Maintaining Multiple Boot Environments 9
Tools for Managing Boot Environments 10
About the Package Manager 10
About the `beadm` Utility 11

2 `beadm` Zones Support 13
`beadm` in Non-Global Zones 13
Unbootable Boot Environments 14
Zones and Shared Datasets 14

3 Creating Boot Environments and Snapshots 17
Creating a Boot Environment 17
`beadm create` Command Options 17
▼ How to Create a Boot Environment 18
Examples of Creating Boot Environments 19
Creating and Copying Snapshots 21
Creating a Snapshot of a Boot Environment 21
Creating a Boot Environment From an Existing Snapshot 22

4 Administering Boot Environments 25
Listing Existing Boot Environments and Snapshots 25
Viewing Boot Environment Specifications 26
Viewing Specifications in Machine-Parsable Output 26

Viewing Snapshot Specifications 27
Changing the Default Boot Environment 27
Mounting and Updating an Inactive Boot Environment 28
▼ How to Mount and Update a Boot Environment 28
Unmounting Boot Environments 29
Destroying a Boot Environment 29
Creating Custom Names for Boot Environments 30
Creating Additional Datasets for Boot Environments 31

Index 33

Preface

Creating and Administering Oracle Solaris 11.1 Boot Environments provides instructions about using the `beadm`(1M) utility to administer multiple boot environments on your Oracle Solaris system.

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

◆ ◆ ◆ CHAPTER 1

Introduction to Managing Boot Environments

This document describes how to use the `beadm` utility to manage boot environments. The `beadm` command is specifically designed to perform tasks on a boot environment structure including a root dataset and all the datasets nested under that root dataset. This chapter describes datasets and provides an overview of managing boot environments.

About Boot Environments and Datasets

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. You can use the `beadm`(1M) utility to create and administer additional boot environments on your system. In addition, the Package Manager GUI provides some options for managing boot environments. Exactly one boot environment can be active at a time.

In terms of file structure, each boot environment consists of a *root dataset* and, optionally, other datasets nested under that root dataset.

Note – 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.

For more information about ZFS datasets, see "ZFS Terminology" in *Oracle Solaris 11.1 Administration: ZFS File Systems*.

The following example shows the root dataset for a sample boot environment named BE1:

```
rpool/ROOT/BE1
```

In the rpool/ROOT/BE1 root dataset example, rpool is the name of the storage pool (zpool). The pool was previously set up and therefore already exists on the system. ROOT is a special dataset that was created by a prior installation. The ROOT dataset is reserved exclusively for use by boot environment root datasets.

The root dataset and any other datasets nested beneath it are included in the BE1 boot environment. These datasets are sometimes referred to as the *critical datasets* for a boot environment.

Shared datasets, in contrast, are located outside the root dataset area of each boot environment. Shared datasets are user-defined directories, such as /export. An example of a shared dataset might be a dataset where user accounts are kept; these user accounts can be accessed regardless of which boot environment is booted.

See the following example:

```
# zfs list
          NAME                      USED  AVAIL  REFER  MOUNTPOINT
          pool                      450K   457G    18K  /pool
          pool/home                 315K   457G    21K  /export/home
          pool/home/anne             18K   457G    18K  /export/home/anne
          pool/home/bob             276K   457G   276K  /export/home/bob
```

Note – For further information, see the zpool(1M) and the zfs(1M) man pages. See, also, "Querying ZFS Storage Pool Status" in *Oracle Solaris 11.1 Administration: ZFS File Systems*.

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

Use the beadm command when you specifically want to create or modify a boot environment, that is, when you want to create or modify a root dataset and the datasets under that root dataset. For example, use the beadm command to make a reference copy of a dataset before making changes to that dataset. The beadm command is based on the zfs technology, but the beadm command has functionality specific to managing root datasets and the datasets under those root datasets. And, you can also use the beadm command to track and manage the associations between multiple root datasets within a global zone or across multiple zones.

You can use the beadm utility to perform actions such as the following on a boot environment:

- **Create a new boot environment or clone an existing boot environment,** A *clone* of a boot environment is created by copying an existing boot environment. A clone is bootable. A clone of the boot environment copies the root dataset and everything hierarchically under the main root dataset of the original boot environment.

  In contrast, shared datasets are not under the root dataset and are not cloned when a boot environment is cloned. Instead, the clone accesses, as needed, the original, shared dataset.

- **Create a snapshot of a boot environment.** A snapshot is a read-only image of a dataset or boot environment at a given point in time. A snapshot usually references some of the files in the original boot environment instead of completely copying those files, thereby saving space in the snapshot. Because snapshots do not include the complete set of files from the boot environment, a snapshot is not bootable.
- **List existing boot environments and snapshots.**
- **Rename a boot environment.** In the renaming process, the `beadm` command will retain any existing associations between global zone boot environments and non-global zone boot environments. Such relationships are based on ZFS properties which the `beadm` command recognizes and maintains during the renaming process.
- **Delete a boot environment.** When a boot environment is deleted, the `beadm` command also deletes associated zone boot environments in the global zone or non-global zones. The `beadm` command tracks the ZFS properties that describe such associations.

Advantages to Maintaining Multiple Boot Environments

Multiple boot environments reduce risk when updating software because system administrators can create backup boot environments before making any software updates to the system. If needed, they have the option of booting a backup boot environment.

The following specific examples show how having more than one Oracle Solaris boot environment and managing them with the `beadm` utility can be useful.

- 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. (The clone you create is a bootable copy of the original.) 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` command with the alternate root (`-R`) option to install or update specific packages on that environment. For more information, see "Installing a Package into a New Boot Environment" in *Adding and Updating Oracle Solaris 11.1 Software Packages*.
- 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 and specifying a snapshot name. For example, if you are doing monthly upgrades to your boot environment, you can capture snapshots for each monthly upgrade.

  Use the command as follows:

  ```
  # beadm create BeName@snapshotdescription
  ```

  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. Provide a custom snapshot description to identify the date or purpose of the snapshot.

Note – You can use the `beadm list -s` command to view the available snapshots for a boot environment.

Although a snapshot is not bootable, you can create a boot environment based on that snapshot by using the `-e` option in 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.

- Using the Package Manager tool or the `pkg` command to install or update packages in your active Oracle Solaris boot environment might 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.

Tools for Managing Boot Environments

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

The Package Manager does not provide the full range of options for managing your boot environments that is available by using the `beadm` command. For more information, see "About the `beadm` Utility" on page 11.

About the Package Manager

The Package Manager is a graphical user interface that enables you to install and manage packages on your installed system. The Package Manager is 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 the Package Manager.

If you use the Package Manager to install 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.

Note – You can boot any active or inactive boot environment by manually selecting it from the x86 GRUB menu or the SPARC boot menu.

You can use the Package Manager to manage your boot environments as follows:

- Delete old and unused boot environments in order to make the disk space available
- Activate a boot environment so that boot environment will be the new default on reboot

For instructions, see the online help that is available in the Package Manager. Or, see Chapter 2, "IPS Graphical User Interfaces," in *Adding and Updating Oracle Solaris 11.1 Software Packages*.

About the `beadm` Utility

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 copy it to a different zpool
- Create a new boot environment and add a custom title and description 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

The `beadm` utility has the following features:

- 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.
- 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.

- Enables you to perform administrative tasks on your boot environments in a global zone or in a non-global zone.
- 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.

`beadm` Utility Requirements

In order to use the `beadm` utility to manage your boot environments, your system must meet the requirements listed in the following table.

TABLE 1–1 System Requirements for Using the `badm` Utility

Requirement	Description
Oracle Solaris operating system	Install the Oracle Solaris release on your system. For instructions, see *Installing Oracle Solaris 11.1 Systems*.
ZFS file systems	The `beadm` utility is designed for ZFS file systems.

◆ ◆ ◆ CHAPTER 2

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 has 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, "Oracle Solaris Zones," in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

The `beadm` utility includes support for creating and administering non-global zone boot environments.

beadm in Non-Global Zones

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

- When you use the `pkg` command, the command only upgrades Oracle Solaris brand zones.
- The `beadm` utility is supported inside a non-global zone.
- Non-global 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.
- Not all of the `beadm` command options can be used in non-global zones. See the specific limitations for each command option in Chapter 3, "Creating Boot Environments and Snapshots," and Chapter 4, "Administering Boot Environments."

Unbootable Boot Environments

Both global zones and non-global zones contain boot environments. Each boot environment in a non-global zone is associated with a parent boot environment in the global zone. If a global zone boot environment is inactive, the related non-global zone boot environment is unbootable. However, if you boot into that parent boot environment in the global zone, the related boot environment in the non-global zone becomes bootable.

Note – If the boot environment is unbootable, it is marked with an exclamation point (!) in the Active column in the `beadm list` output.

The `beadm` command restricts actions on unbootable boot environments as follows:

- You cannot activate an unbootable boot environment.
- You cannot destroy a boot environment that is both unbootable and marked as active on reboot.
- You cannot create a snapshot of an unbootable boot environment.
- You cannot use an unbootable boot environment or boot environment snapshot with the `-e` option of `beadm create`.
- You cannot rename an unbootable boot environment.

Zones and Shared Datasets

The `beadm` utility automatically handles all zones naming tasks related to the `beadm` processes. The `beadm` utility can operate on boot environments in a global zone that also contains non-global zones.

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

zone-path dataset`/rpool/ROOT/`*BE-name*

For example:

```
rpool/zones/zone1/rpool/ROOT/BE1
```

In this example, `rpool/zones/zone1` is the path for a zone root dataset. Nested under that dataset is `rpool/ROOT/BE1`, which is the root dataset for the BE1 boot environment.

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.

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

A shared dataset is identified by using the following format:

zone-path dataset`/rpool/export`

For example:

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

A shared dataset must be explicitly added during zones configuration. A shared dataset is not cloned when the zone dataset is cloned. See the examples in Chapter 3, "Creating Boot Environments and Snapshots."

◆ ◆ ◆ CHAPTER 3

Creating Boot Environments and Snapshots

Use the `beadm` utility to create and copy boot environments and snapshots of boot environments.

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 listed as an alternate boot environment in the GRUB menu for x86 systems or in the boot menu for SPARC systems.

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 create` Command Options

The `beadm create` command has the following options, where *BeName* specifies the name of the boot environment to be created.

`beadm create [-a] [-d` *description*`] [-e` *non-activeBeName* `|` *BeName@snapshot*`] [-o` *property=value*`]...[-p` *zpool*`]` *BeName*

`-a` – Activate the newly created boot environment upon creation. The default is to not activate the newly created boot environment.

`-d` *description* – Provide a custom description to be used as the title in the x86 GRUB menu or the SPARC boot menu to describe the new boot environment. If this option is not used, *BeName* is used for the title.

`-e` *non-activeBeName* – Create a new boot environment from a specified existing, but 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 a specified, existing snapshot of the boot environment.

`-o` *property=value* – Create the datasets for a new boot environment with specific ZFS properties. Multiple `-o` options can be specified. See the `zfs`(1M) man page for more information on the `-o` option.

`-p` *zpool* – Create the datasets for a new boot environment within a specified zpool. If this option is not provided, the default behavior is to create the new boot environment in the same pool as the original boot environment. The `-p` option is not supported within a non-global zone. This option can be combined with the other options.

▼ How to Create a Boot Environment

1 Become an administrator.

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 Create the boot environment.

```
# beadm create BeName
```

BeName is the name of the new boot environment. This new boot environment is inactive.

Note the following:

- *BeName* cannot be a boot environment name that is already being used.
- This command clones the active boot environment, unless the `-e` option is used to specify an inactive boot environment.
- `beadm create` does not create a partial boot environment. The command either successfully creates a full boot environment, or the command fails.

3 (Optional) Use the `beadm mount` command to mount the new boot environment.

```
# beadm mount BeName mount-point
```

You might mount the new boot environment, for example, if you want to modify some configuration files inside the new boot environment before rebooting into it.

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

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.

4 **(Optional) Activate the boot environment.**

```
# beadm activate BeName
```

BeName is 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.

Examples of Creating Boot Environments

The following examples illustrate how to create boot environments, and how the creation process varies depending on the zone and dataset structure. The first example illustrates how cloning operates in a system that includes global and non-global zones. The second example shows dataset specifics related to cloning.

Note – For zones and dataset information, see the following:

- "`beadm` in Non-Global Zones" on page 13
- "Zones and Shared Datasets" on page 14

EXAMPLE 3–1 Cloning a Boot Environment in a Global Zone That Contains Non-Global Zones

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

If the boot environment being cloned has an associated zone boot environment in a non-global zone, that associated boot environment is also cloned. For example, BE1 has an associated zone boot environment, BE2, in a non-global zone. If BE1 is cloned, BE2 is also cloned.

- In this example, the original boot environment in the global zone is named `solaris` with its root dataset at `rpool/ROOT/solaris`.
- A non-global zone named `z1` exists that has the dataset `rpool/zones/z1` as its zonepath. The original `solaris` boot environment in the global zone has an associated boot environment in the `z1` non-global zone. This associated zone boot environment is named `solaris`, with a root dataset at `rpool/zones/z1/rpool/ROOT/solaris`.

EXAMPLE 3–1 Cloning a Boot Environment in a Global Zone That Contains Non-Global Zones *(Continued)*

In this example, you would issue the following command as root to clone the boot environment in the global zone and name the new boot environment `solaris-1`:

```
# beadm create solaris-1
```

The clone is named `solaris-1`, with a root dataset at `rpool/ROOT/solaris-1`.

Note that because the `solaris` boot environment in the global zone has an associated zone boot environment in the `z1` non-global zone, the cloning process also clones the associated zone boot environment in `z1`. The new `solaris-1` clone in zone `z1` has its root dataset at `rpool/zones/z1/rpool/ROOT/solaris-1`.

EXAMPLE 3–2 Creating a New, Cloned Boot Environment with Datasets

This example illustrates how datasets are set up in a newly created boot environment. This example does not involve multiple zones.

As root, you would type the following command.

```
# beadm create BE2
```

The original boot environment in this example is BE1 with a root dataset at `rpool/ROOT/BE1` containing another dataset, `var`.

```
rpool/ROOT/BE1
rpool/ROOT/BE1/var
```

After BE1 is cloned, the new clone, BE2, contains a root dataset and other nested datasets, all cloned from BE1. Because BE1 contains the `/var` file system under the root dataset, `/var` was also cloned.

```
rpool/ROOT/BE2
rpool/ROOT/BE2/var
```

In contrast, if there was a shared file system outside of the root dataset, that shared file system would not have been cloned. The original boot environment and the clone would both "share" the original shared file system, as shown in the next example.

EXAMPLE 3–3 Creating a New Boot Environment With Existing Shared Datasets

This example illustrates creating a new boot environment when there are existing shared datasets. In this example, the original boot environment is BE1, and the shared datasets are `rpool/export` and `rpool/export/home`. This example does not involve multiple zones.

As root, you would type the following command to clone BE1 and name the clone BE2:

```
# beadm create BE2
```

EXAMPLE 3–3 Creating a New Boot Environment With Existing Shared Datasets *(Continued)*

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/`*BeName* datasets and are referenced at their original locations by the cloned boot environment, BE2.

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

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

The root dataset is at `rpool/ROOT/BE1` and a `/var` dataset is located under the root dataset. The root dataset and `/var` are both cloned.

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

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

Creating and Copying Snapshots

You can manually create a snapshot of an existing boot environment for 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. You can then copy that snapshot.

Creating a Snapshot of a Boot Environment

The following command creates a snapshot of the existing boot environment named `BeName`.

`beadm create` *BeName@snapshotdescription*

The snapshot name must use the format *BeName@snapshotdescription*. *BeName* is the name of an existing boot environment that you want to make a snapshot from. If the existing boot environment name is not valid, the command fails. `snapshotdescription` is a custom description to identify the date or purpose of the snapshot.

Note – If you do not use the snapshot name format, the `beadm create` command will try to make a bootable clone instead of a unbootable snapshot. A clone is a complete bootable copy of an image that can be much larger than a mere snapshot. The snapshot merely records what's changed in the datasets instead of copying all the datasets contents.

Note the following snapshot sample names and descriptions:

- `BE1@0312200.12:15pm` – The name for a snapshot of the existing BE1 boot environment. The custom description, `0312200.12:15pm`, records the date and time that the snapshot was taken for future reference.
- `BE2@backup` – The name for a snapshot of an original boot environment named BE2. The snapshot description merely notes that this is a backup of BE2.
- `BE1@march132008` – The name for a snapshot of an original boot environment named BE1. The snapshot description records the date that the snapshot was taken.

Some other system functions automatically take snapshots of a boot environment. Names for such snapshots automatically include a timestamp that indicates when the snapshot was taken. You must use the `beadm create` command if you want to customize a snapshot name.

Creating a Boot Environment From 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 a Snapshot

1 Become an administrator.

For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2 Create a new boot environment from a snapshot.

```
# beadm create -e BEname@snapshotdescription  NewName
```

BEname@snapshotdescription is the name of an existing snapshot and description. *NewName* is a custom name for your new boot environment.

For example:

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

This command creates a new boot environment named `BE2` from the existing snapshot named `BE1@now`.

Next Steps You can activate this new boot environment. See "Changing the Default Boot Environment" on page 27.

◆ ◆ ◆ CHAPTER 4

Administering Boot Environments

This chapter describes administration tasks related to boot environments, covering the following topics:

- Listing Existing Boot Environments and Snapshots
- Changing the Default Boot Environment
- Mounting and Updating an Inactive Boot Environment
- Destroying a Boot Environment
- Creating Custom Names for Boot Environments

Listing Existing Boot Environments and Snapshots

You can display information about snapshots, boot environments, and datasets that were created by the `beadm` command by using the `beadm list` subcommand. The `beadm list` command output also displays boot environments that are created by the `pkg` command.

The `beadm list` command syntax is:

`beadm list [[-a | [-d] [-s]] [-H] [`*`beName`***`]`**

The command lists information about the existing boot environment. To view information for a specific boot environment, replace *beName* with a boot environment name. If a specific boot environment is not specified, the command lists information about all boot environments. The default is to list boot environments without additional information.

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

`-d` – Lists information about all subordinate datasets that belong to the boot environment.

`-s` – Lists information about the snapshots of the boot environment.

`-H` – Lists information in machine-parseable format. Each field in the output is separated by a semicolon.

Viewing Boot Environment Specifications

The `-a` option shows full information for a specified boot environment or for all boot environments, including all dataset and snapshot information.

The values for the Active column are as follows:

- `R` – Active on reboot.
- `N` – Active now.
- `NR` – Active now and active on reboot.
- `-` – Inactive.
- `!` – Unbootable boot environments in a non-global zone

The following example displays full information for the `BE5` boot environment.

```
# beadm list -a BE5
BE/Dataset/Snapshot    Active Mountpoint Space   Policy Created
-------------------    ------ ---------- -----   ------ -------
BE5
   p/ROOT/BE5          NR     /          6.10G   static 2011-09-09 16:53
   p/ROOT/BE5/var      -      /var       24.55M  static 2011-09-09 16:53
   p/ROOT/BE5/var@boo  -      -          18.38M  static 2011-09-10 00:59
   p/ROOT/BE5/var@foo  -      -          18.38M  static 2011-06-10 16:37
   p/ROOT/BE5@boo      -      -          139.44M static 2011-09-10 00:59
   p/ROOT/BE5@foo      -      -          912.85M static 2011-06-10 16:37
```

Viewing Specifications in Machine-Parsable Output

The `-H` option suppresses header titles and displays results separated by semicolons. The following example shows information for all boot environments.

```
# beadm list -H
BE2;4659d6ee-76a0-c90f-e2e9-a3fcb570ccd5;;;55296;static;1211397974
BE3;ff748564-096c-449a-87e4-8679221d37b5;;;339968;static;1219771706
BE4;1efe3365-02c5-6064-82f5-a530148b3734;;;16541696;static;1220664051
BE5;215b8387-4968-627c-d2d0-f4a011414bab;NR;/;7786206208;static;1221004384
```

Each field in the output is separated by a semicolon. The output fields, in display order, are as follows.

TABLE 4–1 Output Fields

Field	Description
1	BE name
2	UUID
3	Active
4	Mountpoint

TABLE 4–1 Output Fields *(Continued)*

Field	Description
5	Space; size in bytes
6	Policy
7	Creation time (in seconds since 00:00:00 UTC, Jan 1, 1970)

Each field is separated by a semicolon. In this example, a boot environment was not specified in the command, so all boot environments are displayed. Because no other options were used with the command, the universally unique identifier (UUID) for the boot environment is provided in the second field. In this example, the UUID for BE5 is `215b8387-4968-627c-d2d0-f4a011414bab`. For a boot environment in a non-global zone, the UUID field represents the parent ID with which that boot environment is associated.

Viewing Snapshot Specifications

The `-s` option displays information for any snapshots that exist.

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

```
# beadm list -s test-2
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
```

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.

Use the `beadm activate` command as follows to activate an existing, inactive boot environment:

```
# beadm activate BeName
```

`beadm activate` sets the specified boot environment as the default in the `menu.lst` file.

Note – When a boot environment is created, regardless of 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.

Mounting and Updating an Inactive Boot Environment

To update packages on an existing, inactive boot environment, mount that environment first, in order to gain access to it, Then, you can update packages on it.

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

The command syntax is as follows.

`beadm mount` *BeName mount-point*

The command mounts a specified boot environment at a specified mount point. If the mount point already exists, it must be empty. If the directory for the mount point does not exist, the `beadm` utility creates the directory, then mounts the boot environment on that directory. Although the boot environment is mounted, it remains inactive.

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

▼ How to Mount and Update a Boot Environment

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Mount the boot environment.**

   ```
   # beadm mount BeName mount-point
   ```

3. **(Optional) 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 package-name
   ```

where `/mnt` is the mount point for the boot environment.

Unmounting Boot Environments

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

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

The command syntax is as follows.

`beadm unmount [-f]` *BeName*

The command unmounts the specified boot environment.

The `-f` option forcefully unmounts the boot environment even if it is currently busy.

Destroying a Boot Environment

To make more room available on your system, use the `beadm` command to destroy an existing boot environment. The command syntax is as follows:

`beadm destroy [-fF]` *BeName* | *BeName@snapshot*

The command destroys the specified boot environment or snapshot. The command prompts the user to provide confirmation before destroying the boot environment.

`-f` – Forces destruction of the boot environment even if it is mounted.

`-F` – Forces destruction of the boot environment without prompting for confirmation.

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.
- When you destroy an inactive boot environment, any zone boot environments that are associated with that inactive boot environment are also destroyed.
- The `beadm destroy` command destroys only the 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.

  In the following example, 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
```

You would 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
```

Creating Custom Names for Boot Environments

The `beadm rename` command enables you to rename an existing boot environment so you can supply 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 to the new boot environment name.

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 affect the relationships between the zones and their related boot environments.

An active boot environment cannot be renamed. Only an inactive boot environment can be renamed.

You cannot rename the boot environment that is currently booted. If you want to rename the active boot environment, you must first make a different boot environment active and boot that environment. Then you can rename the inactive boot environment.

The command syntax is as follows:

`beadm rename` *BeName newBeName*

The command renames Bename to newBeName.

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

Note – A new boot environment or a backup boot environment might be created when you install, update, or uninstall a package using the `pkg` command. You can create custom names for new or backup boot environments by using the `-be-name` or `-backup-be-name` options with the `pkg` command. For information, see "Boot Environment Options" in *Adding and Updating Oracle Solaris 11.1 Software Packages*.

Creating Additional Datasets for Boot Environments

You can create additional private datasets for an existing boot environment by using the `zfs` command to create a dataset located hierarchically under the root dataset of the boot environment.

This new dataset will not be automatically cloned to existing, associated zone boot environments. Also, because the new dataset is under the root dataset of a boot environment, the new dataset will be private to that boot environment. The new dataset will not be a shared dataset.

Once this new dataset has been created under the root dataset, if you then clone that whole boot environment, the new dataset would be included in the new, cloned boot environment.

Note – When creating additional datasets for boot environments, the `canmount` ZFS property of the dataset must be set to `noauto`. The dataset's mountpoint is inherited from the boot environment's root dataset and hence should be mounted at `/myfs` when this boot environment is in use.

For example, to create a new, non-shared dataset mounted at `/myfs` for the boot environment, `BE1`:

```
# zfs create -o canmount=noauto rpool/ROOT/BE1/myfs
```

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

Index

A

Active column in boot environment listing, 26
administering boot environments, 25–31
advantages to maintaining multiple boot environments, 9–10

B

`beadm activate` command, 27–28
`beadm create` command
  cloning boot environments and zones, 19
  options, 17–18
`beadm destroy` command, 29–30
`beadm list` command, 25–27
  displaying snapshot information with, 27
  options, 25
  output example, 26
  suppressing header titles in output, 26–27
  unbootable environments and, 14
  values for Active column in output, 26
`beadm mount` command, 28
`beadm rename` command, 30–31
`beadm unmount` command, 29
`beadm` utility
  overview, 11–12
  syntax and options, 17–18
  system requirements for using, 12
  when to use, 8
  zones support
    in non-global zones, 13
    unbootable environments and, 14
`beadm` utility, zones support *(Continued)*
    zones naming conventions, 14–15
boot environments
  adding datasets, 31
  administering, 25–31
  administering with `beadm` utility, 11–12
  advantages to maintaining multiple, 9–10
  changing default, 27–28
  clone of, 8
  creating, 17–21
    cloning a snapshot, 22–23
    examples, 19–21
    with existing shared datasets, 20
    with new datasets, 20
  definition of, 7
  destroying, 29–30
  listing, 25–27
  making an inactive boot environment active, 27–28
  mounting and updating an inactive, 28–29
  overview, 7–9
  renaming, 30–31
  snapshot of, 9
  unbootable
    displayed in `beadm list` output, 14
    restricted actions, 14
    zones support and, 14
  unmounting, 29
  using Package Manager to administer, 10–11
  viewing specifications
    in machine-parsable output, 26–27
    of snapshots, 27
  viewing specifications for, 26

C

clone
- *See also* boot environments, creating
- definition of, 8
- setting up datasets in new, 20
- shared datasets and, 8

D

datasets
- creating additional, 31
- definition of, 7
- root
  - definition of, 8
  - naming conventions, 14–15
- setting up in new boot environment, 20
- shared
  - creating new boot environment with existing, 20
  - definition of, 8
  - identifying, 15

destroying a boot environment, 29–30

I

inactive boot environments
- making active, 27–28
- mounting and updating, 28–29

L

listing boot environments information, 25–27

M

mounting an inactive boot environment, 28–29

N

non-global zones, `beadm` utility and, 13

P

Package Manager, using to manage boot environments, 10–11

R

renaming boot environments, 30–31

root datasets
- definition of, 8
- naming conventions, 14–15

S

shared datasets
- creating new boot environment with existing, 20
- definition of, 8
- identifying, 15

snapshots
- creating, 21–22
- creating a boot environment from existing, 22–23
- definition of, 9, 21
- naming, 21
- viewing specifications of, 27

system requirements for using `beadm` utility, 12

U

unbootable boot environments
- in `beadm list` output, 14
- zones and, 14

unmounting boot environments, 29

updating packages on an inactive boot environment, 28–29

Z

zones
- how to view information related to, 13
- restricted `beadm` actions on unbootable environments, 14
- root dataset naming conventions, 14–15

zones *(Continued)*
support for `beadm` utility
in non-global zones, 13
zones naming conventions, 14–15