

Creating and Administering Oracle® Solaris 11 Boot Environments

Copyright © 2008, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

Contents

Preface	5
1 Introduction to Managing Boot Environments	9
Advantages to Maintaining Multiple Boot Environments	10
Tools for Managing Boot Environments	11
About the Package Manager	11
About the beadm Utility	12
2 beadm Zones Support	15
beadm in Non-Global Zones	15
Unbootable Boot Environments	16
Zones Naming Conventions	16
3 Creating Boot Environments and Snapshots	19
Creating a Boot Environment	19
beadm create Command Options	19
▼ How to Create a Boot Environment	20
Examples of Creating Boot Environments	21
Creating and Copying Snapshots	23
Creating a Snapshot of a Boot Environment	23
Creating a Boot Environment From an Existing Snapshot	23
4 Administering Boot Environments	25
Listing Existing Boot Environments and Snapshots	25
Viewing Boot Environment Specifications	26
Viewing Specifications in Machine-Parsable Output	27
Viewing Snapshot Specifications	27

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

Preface

Creating and Administering Oracle Solaris 11 Boot Environments provides instructions about using the `beadm(1M)` utility to administer multiple boot environments on your Oracle Solaris system.

Related Information

Installing Oracle Solaris 11 Systems provides instructions for installing and configuring the Oracle Solaris operating system (OS) using any of the following methods:

- A LiveCD image
- An interactive text installer
- The Oracle Solaris Automated Installer (AI) feature
- The Oracle Solaris SCI Tool interactive system configuration tool
- The `sysconfig(1M)` command line system configuration tool

Creating a Custom Oracle Solaris 11 Installation Image explains how to use the Oracle Solaris Distribution Constructor (DC) tool to customize your installation image.

Chapter 6, “Managing Services (Overview),” in *Oracle Solaris Administration: Common Tasks* describes the Oracle Solaris Service Management Facility (SMF) feature. You can use SMF profiles to configure your system.

The `pkg(5)` man page describes the Oracle Solaris Image Packaging System (IPS) feature, which enables you to store and retrieve software packages for installation. The `pkg(1)` man page explains how to install IPS packages.

See the Oracle Solaris 11 System Administration documentation for more information about how to administer Oracle Solaris 11 systems.

Transitioning From Oracle Solaris 10 JumpStart to Oracle Solaris 11 Automated Installer provides information to help you migrate from JumpStart to AI, both of which are automated installation features of Oracle Solaris.

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	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name%</code> su Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> 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#

Introduction to Managing 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. You can use the `beadm(1M)` utility to create and administer additional boot environments on your system.

Note – In addition, the Package Manager GUI provides some options for managing boot environments.

Note the following distinctions relevant to boot environment administration:

- 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 *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.
- 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 *clone* of a boot environment is created by copying another boot environment. A clone is bootable.
- *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.

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 boot environment's *critical datasets* are included within the root dataset area for that environment.

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.

- 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. 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@snapshotNamedescription
```

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.

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 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.

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 described in [“About the `beadm` Utility” on page 12](#).

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.

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
- Active 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, “TPS Graphical User Interfaces,” in *Adding and Updating Oracle Solaris 11 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 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

Requirement	Description
Oracle Solaris operating system	Install the Oracle Solaris release on your system. For instructions, see Installing Oracle Solaris 11 Systems .
ZFS file systems	The <code>beadm</code> utility is designed for ZFS file systems.

beadm Limitations in the Current Release

Note the following `beadm` limitations:

- The `beadm` utility is implemented with ZFS support only.
- The `beadm` utility manages only boot environments and snapshots that were created by the `beadm` utility. If you manually create the datasets for a boot environment by using the `zfs (1M)` command, the `beadm` utility should recognize the boot environment, but the available management tasks for this boot environment might be undefined or not supported.

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 Solaris brand zones.
- **Starting with the Oracle Solaris 11/11 release,** the `beadm` utility is supported inside a non-global zone.
- Non-global zone support is limited to ZFS support.
- 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, so that, 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 Naming Conventions

The `beadm` utility automatically handles all zones naming tasks related to the `beadm` processes. When the `beadm` utility is operating on boot environments in a global zone that also contains non-global zones, the zones naming conventions that are automatically used by the `beadm` utility are as follows.

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

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

For example:

```
rpool/zones/zone1/rpool/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. Shared datasets are located outside the root dataset area of each boot environment.

Note – Shared datasets are user-defined directories, such as `/export`, that contain the same mount point in both the active and inactive boot environments.

A shared dataset is identified by using the following format:

```
<zonepath 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.”](#)

Creating Boot Environments and Snapshots

The `beadm` utility can be used 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.

Syntax: `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* – Enter a custom description 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.

Note – The -p option can be combined with the other options.

▼ How to Create a Boot Environment

1 Become the root role.

2 Create the boot environment.

```
# 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.

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

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

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.

4 (Optional) 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.

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.

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

- [“beadm in Non-Global Zones” on page 15](#)
 - [“Zones Naming Conventions” on page 16](#)
-

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 cloning a boot environment that contains two non-global zones.

- The original boot environment in the global zone is named `solaris` with the root dataset at `rpool/ROOT/solaris`.
- A zone named `z1` exists that has the dataset `rpool/zones/z1` as its zonepath. The original boot environment in the global zone has a copy of this zone's root dataset at `rpool/zones/z1/rpool/ROOT/solaris`.

Become the root role, and perform the following command to clone the boot environment and name the new boot environment `solaris-1`:

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

The root dataset for the new boot environment is at `rpool/ROOT/solaris-1`. The copy of zone `z1` for the new boot environment 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.

Become the root role and type the following command.

```
# beadm create BE2
```

EXAMPLE 3-2 Creating a New, Cloned Boot Environment with Datasets *(Continued)*

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, then those datasets are also cloned.

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

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

rpool in this example is the name of the storage pool. The pool was previously set up by the initial installation or upgrade and, therefore, already exists on the system. 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 3-3 Creating a New Boot Environment With Shared Datasets

This example illustrates a new boot environment that has shared datasets.

Become the root role and 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/<BeName> 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/var  
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/var  
rpool/export  
rpool/export/home
```

Creating and Copying Snapshots

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. 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`.

Syntax: `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.

Some snapshot names are:

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

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.

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 the root role.**
- 2 **Create a new boot environment from a snapshot.**

```
# beadm create -e BName@snapshotdescription BeName
```

Replace the variable, *BName@snapshotdescription*, with the name of an existing snapshot. Replace the variable, *BName*, with 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.

As a next step, you could activate this new boot environment. See [“Changing the Default Boot Environment”](#) on page 28.

Administering Boot Environments

This chapter describes administration tasks related to boot environments. It covers the following topics:

- “Listing Existing Boot Environments and Snapshots” on page 25
- “Changing the Default Boot Environment” on page 28
- “Mounting and Updating an Inactive Boot Environment” on page 28
- “Destroying a Boot Environment” on page 29
- “Creating Custom Names for Boot Environments” on page 30

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.

Note – Snapshots and boot environments can be created by utilities other than the `beadm` command. 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 syntax is:

```
Syntax: beadm list [-a | [-ds] [-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 – Prevents listing header information. Each field in the output is separated by a semicolon.

The output displays without a header as in this example:

```
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 blank line.

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 are represented by an exclamation point.

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 show 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 is separated by a semicolon. In this example, a boot environment was not specified in the command, so all boot environments are displayed. If 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. The second field is blank for BE2, BE3, and BE4, indicating that there boot environments do not have a UUID. 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 and, optionally, 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.

```
Syntax: 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 the root role.

2 Mount the a 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.

```
Syntax: 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:

```
Syntax: 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, the zones that belong to that boot environment are also destroyed.
- 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.

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
```

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 with 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:

Syntax: `beadm rename BeName newBeName`

The command changes the name of an existing boot environment to a new name.

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

Creating Additional Datasets for Boot Environments

To create additional datasets for a boot environment, use the `zfs` command to create a dataset located hierarchically under the root dataset of the boot environment. This dataset will not be shared with other boot environments.

When creating additional datasets for boot environments, the `canmount zfs` property of the dataset must be set to `noauto`.

For example, to create a new, non-shared dataset mounted at `/myfs` for the boot environment, `BE1`, the following command can be used.

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

The dataset's mountpoint is inherited from the boot environment's root dataset and, hence, will be mounted at `/myfs` when this boot environment is in use.

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

