Creating and Administering Oracle® Solaris 11.2 Boot Environments

Part No: E36803
July 2014

Contents

Using This Documentation

- **Overview** – Describes how to manage and create boot environments
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Experience administrating a Oracle Solaris system

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://www.oracle.com/pls/topic/lookup?ctx=E36784.

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.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Introduction to Managing Boot Environments

This document describes how to use the `beadm` command 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` command 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 “Managing ZFS File Systems in Oracle Solaris 11.2 ”.

The following example shows the root dataset for a sample boot environment named BE1: `rpool/ROOT/BE1`. In this 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 the installer. The rpool/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.

```
# zfs list
NAME                    USED  AVAIL  REFER  MOUNTPOINT
rpool                  42.5G  24.4G  4.65M  /rpool
rpool/ROOT             25.6G  24.4G    31K  legacy
rpool/ROOT/BE1         8.91M  24.4G  4.17G  /
rpool/ROOT/BE1/var     3.96M  24.4G   276M  /var
```

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
rpool                      42.5G  24.4G  4.65M  /rpool
rpool/ROOT                 25.6G  24.4G    31K  legacy
rpool/ROOT/BE1             8.91M  24.4G  4.17G  /
rpool/ROOT/BE1/var         3.96M  24.4G   276M  /var
tank                        450K   457G    18K  /export
tank/home                   315K   457G    21K  /export/home
```

Note - For further information, see the `zpool`(1M) and the `zfs`(1M) man pages. See, also, “Querying ZFS Storage Pool Status” in “Managing ZFS File Systems in Oracle Solaris 11.2 ”.

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.

The `beadm` command should always be used to manipulate boot environments. For example, use the `beadm` command to make a reference copy of a boot environment before making changes. Even though the `beadm` command is based on zfs technology, it has additional functionalities specific to managing a BE's root dataset and the datasets under the root datasets. And, the command also has special logic to track and manage the associations between BEs within a global zone, within multiple zones, or across multiple zones. For more information, see the `beadm`(1M) man page.

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 command 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 Software in Oracle Solaris 11.2 ".
- 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.

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 `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 and is unaffected by the update.
- 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.

About the `beadm` Command

The `beadm` command 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 in 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` command has the following features:

- Aggregates all datasets in a boot environment and performs actions on the entire boot environment at once. You should not perform ZFS commands to modify each dataset individually.
- Manages the dataset structures within boot environments. For example, when the `beadm` command clones a boot environment that has shared datasets, the command 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` command to create a new boot environment, that environment is automatically added to the GRUB menu or boot menu.

◆ ◆ ◆ 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 "Creating and Using Oracle Solaris Zones ".

The beadm command 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 command and in related processes:

- The beadm command is supported inside a non-global zone.
- The root dataset for non-global zones must not be under 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 command affects the non-global zones on your system, the beadm command does not display zones information. Use the zoneadm command 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, seezoneadm(1M) the 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 the beadm(1M) man page.

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` command automatically handles all zones naming tasks related to the `beadm` processes. The `beadm` command 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` command 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:

`-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.

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.

How to Clone a Boot Environment

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in "Securing Users and Processes in Oracle Solaris 11.2 ".

2. **Clone 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` command 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 Cloning Boot Environments

The following examples illustrate how to clone boot environments, and how the cloning 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. The third example show what happens with shared datasets when you clone a system.

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

- "`beadm` in Non-Global Zones" on page 11
- "Zones and Shared Datasets" on page 12

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`.

```
# zfs list -r rpool
NAME                                    USED  AVAIL  REFER MOUNTPOINT
rpool                                  11.5G  3.89G  4.46M  /rpool
rpool/ROOT                             8.47G  3.89G    31K  legacy
rpool/ROOT/solaris                     2.98M  3.89G  2.49G  /
rpool/ROOT/solaris/var                  428K  3.89G   298M  /var
rpool/dump                             1.03G  3.92G  1.00G  -
rpool/export                            120K  3.89G    32K  /export
rpool/export/home                      88.5K  3.89G    32K  /export/home
rpool/export/home/user1                56.5K  3.89G  56.5K  /export/home/user1
rpool/swap                             1.03G  3.92G  1.00G  -
rpool/zones                             672M  3.89G    32K  /zones
rpool/zones/z1                          672M  3.89G    32K  /zones/z1
rpool/zones/z1/rpool                    671M  3.89G    31K  /rpool
rpool/zones/z1/rpool/ROOT               671M  3.89G    31K  legacy
rpool/zones/z1/rpool/ROOT/solaris       671M  3.89G   591M  /zones/z1/root
rpool/zones/z1/rpool/ROOT/solaris/var  79.3M  3.89G  78.4M  /zones/z1/root/var
rpool/zones/z1/rpool/export             62K  3.89G    31K  /export
rpool/zones/z1/rpool/export/home        31K  3.89G    31K  /export/home
```

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 test
# zfs list -r rpool
NAME                                      USED  AVAIL  REFER MOUNTPOINT
rpool                                    11.5G  3.89G  4.46M  /rpool
rpool/ROOT                               8.47G  3.89G    31K  legacy
rpool/ROOT/solaris                       2.98M  3.89G  2.49G  /
rpool/ROOT/solaris/var                    428K  3.89G   298M  /var
rpool/ROOT/test                            71K  3.89G  2.50G  /
rpool/ROOT/test/var                         1K  3.89G  2.88G  /var
rpool/dump                               1.03G  3.92G  1.00G  -
rpool/export                              120K  3.89G    32K  /export
rpool/export/home                        88.5K  3.89G    32K  /export/home
rpool/export/home/user1                  56.5K  3.89G  56.5K  /export/home/user1
rpool/swap                               1.03G  3.92G  1.00G  -
rpool/zones                               672M  3.89G    32K  /zones
rpool/zones/z1                            672M  3.89G    32K  /zones/z1
rpool/zones/z1/rpool                      671M  3.89G    31K  /rpool
rpool/zones/z1/rpool/ROOT                 671M  3.89G    31K  legacy
rpool/zones/z1/rpool/ROOT/solaris         671M  3.89G   591M  /zones/z1/root
rpool/zones/z1/rpool/ROOT/solaris/var    79.3M  3.89G  78.4M  /zones/z1/root/var
```

```
rpool/zones/z1/rpool/ROOT/solaris-1        2K  3.89G   591M  /
rpool/zones/z1/rpool/ROOT/solaris-1/var    1K  3.89G  78.4M  /var
rpool/zones/z1/rpool/export               62K  3.89G    31K  /export
rpool/zones/z1/rpool/export/home          31K  3.89G    31K  /export/home
```

The clone is named `test`, with a root dataset at `rpool/ROOT/test`.

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 Cloning a New 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`.

```
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    42.5G  24.4G  4.65M  /rpool
rpool/ROOT               25.6G  24.4G    31K  legacy
rpool/ROOT/BE1           8.91M  24.4G  4.17G  /
rpool/ROOT/BE1/var       3.96M  24.4G   276M  /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.

```
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    42.5G  24.4G  4.65M  /rpool
rpool/ROOT               25.6G  24.4G    31K  legacy
rpool/ROOT/BE1           8.91M  24.4G  4.17G  /
rpool/ROOT/BE1/var       3.96M  24.4G   276M  /var
rpool/ROOT/BE2           8.91M  24.4G  4.17G  /
rpool/ROOT/BE2/var       3.96M  24.4G   276M  /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
```

The shared datasets, `tank` and `tank/export`, 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:

```
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    42.5G  24.4G  4.65M  /rpool
rpool/ROOT               25.6G  24.4G    31K  legacy
rpool/ROOT/BE1           8.91M  24.4G  4.17G  /
rpool/ROOT/BE1/var       3.96M  24.4G   276M  /var
tank                      450K   457G    18K  /tank
tank/home                 315K   457G    21K  /tank/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, `tank` and `tank/home`, are unchanged.

```
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    42.5G  24.4G  4.65M  /rpool
rpool/ROOT               25.6G  24.4G    31K  legacy
rpool/ROOT/BE1           8.91M  24.4G  4.17G  /
rpool/ROOT/BE1/var       3.96M  24.4G   276M  /var
rpool/ROOT/BE2           8.91M  24.4G  4.17G  /
rpool/ROOT/BE2/var       3.96M  24.4G   276M  /var
tank                      450K   457G    18K  /tank
tank/home                 315K   457G    21K  /tank/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.

Cloning a Boot Environment From an Existing Snapshot

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

▼ How to Clone a Boot Environment From a Snapshot

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in "Securing Users and Processes in Oracle Solaris 11.2 ".

2. **Clone 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 26.

♦ ♦ ♦ C H A P T E R 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` command. The `beadm list` command output also displays boot environments that are created by the `pkg` command.

To view information for a specific boot environment, include a boot environment name on the command line. 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.

The following options are available:

`-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 Information About Boot Environments

The -a option shows full information for a specified boot environment or for all boot environments, including all dataset and snapshot information. This information includes flags to indicate whether the boot environment is active or not, the mountpoint for the dataset, the amount of space used by the dataset, the policy and the date the dataset was created.

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 2013-09-09 16:53
   p/ROOT/BE5/var      -      /var       24.55M  static 2013-09-09 16:53
   p/ROOT/BE5/var@boo  -      -          18.38M  static 2013-09-10 00:59
   p/ROOT/BE5/var@foo  -      -          18.38M  static 2013-06-10 16:37
   p/ROOT/BE5@boo      -      -          139.44M static 2013-09-10 00:59
   p/ROOT/BE5@foo      -      -          912.85M static 2013-06-10 16:37
```

Viewing Information About Boot Environments 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 for `beadm list -H`

Field	Description
1	BE name
2	UUID
3	Active
4	Mountpoint
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@2013-04-12-22:29:27 264.02M static 2013-04-12 16:29
   test-2@2013-06-02-20:28:51 32.50M  static 2013-06-02 14:28
   test-2@2013-06-03-16:51:01 16.66M  static 2013-06-03 10:51
   test-2@2013-07-13-22:01:56 25.93M  static 2013-07-13 16:01
   test-2@2013-07-21-17:15:15 26.00M  static 2013-07-21 11:15
   test-2@2013-07-25-19:07:03 13.75M  static 2013-07-25 13:07
   test-2@2013-07-25-20:33:41 12.32M  static 2013-07-25 14:33
   test-2@2013-07-25-20:41:23 30.60M  static 2013-07-25 14:41
   test-2@2013-08-06-15:53:15 8.92M   static 2013-08-06 09:53
   test-2@2013-08-06-16:00:37 8.92M   static 2013-08-06 10:00
   test-2@2013-08-09-16:06:11 193.72M static 2013-08-09 10:06
   test-2@2013-08-09-20:28:59 102.69M static 2013-08-09 14:28
   test-2@install             205.10M static 2013-03-16 19:04
```

Changing the Default Boot Environment

You can change an inactive boot environment into an active boot environment, which means that the named boot environment will be used when the system is next rebooted. 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 `beadm mount` 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` command 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 “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

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 /tank and /tank/home datasets. The datasets include the following:

  ```
  # zfs list
  NAME                      USED  AVAIL  REFER  MOUNTPOINT
  rpool                    42.5G  24.4G  4.65M  /rpool
  rpool/ROOT               25.6G  24.4G    31K  legacy
  rpool/ROOT/BE1           8.91M  24.4G  4.17G  /
  rpool/ROOT/BE1/var       3.96M  24.4G   276M  /var
  rpool/ROOT/BE2           8.91M  24.4G  4.17G  /
  rpool/ROOT/BE2/var       3.96M  24.4G   276M  /var
  tank                      450K   457G    18K  /export
  tank/home                 315K   457G    21K  /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:

  ```
  # zfs list
  NAME                      USED  AVAIL  REFER  MOUNTPOINT
  rpool                    42.5G  24.4G  4.65M  /rpool
  rpool/ROOT               25.6G  24.4G    31K  legacy
  rpool/ROOT/BE1           8.91M  24.4G  4.17G  /
  rpool/ROOT/BE1/var       3.96M  24.4G   276M  /var
  tank                      450K   457G    18K  /export
  tank/home                 315K   457G    21K  /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 Software in Oracle Solaris 11.2 ”.

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