

**NAME**

**zfs-rename** - rename ZFS dataset

**SYNOPSIS**

**zfs rename** [-f] *filesystem|volume|snapshot filesystem|volume|snapshot*

**zfs rename -p** [-f] *filesystem|volume filesystem|volume*

**zfs rename -u** [-f] *filesystem filesystem*

**zfs rename -r** *snapshot snapshot*

**DESCRIPTION**

**zfs rename** [-f] *filesystem|volume|snapshot filesystem|volume|snapshot*

**zfs rename -p** [-f] *filesystem|volume filesystem|volume*

**zfs rename -u** [-f] *filesystem filesystem*

Renames the given dataset. The new target can be located anywhere in the ZFS hierarchy, with the exception of snapshots. Snapshots can only be renamed within the parent file system or volume. When renaming a snapshot, the parent file system of the snapshot does not need to be specified as part of the second argument. Renamed file systems can inherit new mount points, in which case they are unmounted and remounted at the new mount point.

**-f** Force unmount any file systems that need to be unmounted in the process. This flag has no effect if used together with the **-u** flag.

**-p**  
Creates all the nonexistent parent datasets. Datasets created in this manner are automatically mounted according to the **mountpoint** property inherited from their parent.

**-u**  
Do not remount file systems during rename. If a file system's **mountpoint** property is set to **legacy** or **none**, the file system is not unmounted even if this option is not given.

**zfs rename -r** *snapshot snapshot*

Recursively rename the snapshots of all descendent datasets. Snapshots are the only dataset that can be renamed recursively.

**EXAMPLES****Example 1: Promoting a ZFS Clone**

The following commands illustrate how to test out changes to a file system, and then replace the original file system with the changed one, using clones, clone promotion, and renaming:

```
# zfs create pool/project/production
  populate /pool/project/production with data
# zfs snapshot pool/project/production@today
# zfs clone pool/project/production@today pool/project/beta
  make changes to /pool/project/beta and test them
# zfs promote pool/project/beta
# zfs rename pool/project/production pool/project/legacy
# zfs rename pool/project/beta pool/project/production
  once the legacy version is no longer needed, it can be destroyed
# zfs destroy pool/project/legacy
```

**Example 2:** Performing a Rolling Snapshot

The following example shows how to maintain a history of snapshots with a consistent naming scheme. To keep a week's worth of snapshots, the user destroys the oldest snapshot, renames the remaining snapshots, and then creates a new snapshot, as follows:

```
# zfs destroy -r pool/users@7daysago
# zfs rename -r pool/users@6daysago @7daysago
# zfs rename -r pool/users@5daysago @6daysago
# zfs rename -r pool/users@4daysago @5daysago
# zfs rename -r pool/users@3daysago @4daysago
# zfs rename -r pool/users@2daysago @3daysago
# zfs rename -r pool/users@yesterday @2daysago
# zfs rename -r pool/users@today @yesterday
# zfs snapshot -r pool/users@today
```