NAME

zfs-set - set properties on ZFS datasets

SYNOPSIS

zfs set [**-u**] *property=value* [*property=value*]<?> *filesystem|volume|snapshot*<?>
zfs get [**-r**|**-d** *depth*] [**-Hp**] [**-o** *field*[,*field*]<?>] [**-s** *source*[,*source*]<?>] [**-t** *type*[,*type*]<?>]
    all|*property*[,*property*]<?> [*filesystem|volume|snapshot|bookmark*]<?>
zfs inherit [**-rS**] *property filesystem|volume|snapshot*<?>

DESCRIPTION

zfs set [**-u**] *property=value* [*property=value*]<?> *filesystem|volume|snapshot*<?>
Only some properties can be edited. See zfsprops(7) for more information on what properties can be
set and acceptable values. Numeric values can be specified as exact values, or in a human-readable
form with a suffix of **B**, **K**, **M**, **G**, **T**, **P**, **E**, **Z** (for bytes, kilobytes, megabytes, gigabytes, terabytes,
petabytes, exabytes, or zettabytes, respectively). User properties can be set on snapshots. For more
information, see the *User Properties* section of zfsprops(7).

-u Update mountpoint, sharenfs, sharesmb property but do not mount or share the dataset.

zfs get [**-r**|**-d** *depth*] [**-Hp**] [**-o** *field*[,*field*]<?>] [**-s** *source*[,*source*]<?>] [**-t** *type*[,*type*]<?>]
all|*property*[,*property*]<?> [*filesystem|volume|snapshot|bookmark*]<?>
Displays properties for the given datasets. If no datasets are specified, then the command displays
properties for all datasets on the system. For each property, the following columns are displayed:

    name   Dataset name
    property Property name
    value   Property value
    source   Property source **local**, **default**, **inherited**, **temporary**, **received**, or **-** (none).

All columns are displayed by default, though this can be controlled by using the **-o** option. This
command takes a comma-separated list of properties as described in the *Native Properties* and *User
Properties* sections of zfsprops(7).

The value **all** can be used to display all properties that apply to the given dataset's type (**filesystem**,
volume, **snapshot**, or **bookmark**).

-H   Display output in a form more easily parsed by scripts. Any headers are omitted, and fields
            are explicitly separated by a single tab instead of an arbitrary amount of space.

-d *depth* Recursively display any children of the dataset, limiting the recursion to *depth*. A depth of **1**
            will display only the dataset and its direct children.

-o *field*   A comma-separated list of columns to display, defaults to **name**,**property**,**value**,**source**.

-p   Display numbers in parsable (exact) values.

-r   Recursively display properties for any children.

-s *source* A comma-separated list of sources to display. Those properties coming from a source other than those in this list are ignored. Each source must be one of the following: **local**, **default**, **inherited**, **temporary**, **received**, or **none**. The default value is all sources.

-t *type*   A comma-separated list of types to display, where *type* is one of **filesystem**, **snapshot**, **volume**, **bookmark**, or **all**.

zfs inherit [**-rS**] *property filesystem|volume|snapshot*<?>
Clears the specified property, causing it to be inherited from an ancestor, restored to default if no ancestor has the property set, or with the **-S** option reverted to the received value if one exists. See zfsprops(7) for a listing of default values, and details on which properties can be inherited.

-r Recursively inherit the given property for all children.

-S

   Revert the property to the received value, if one exists; otherwise, for non-inheritable properties, to the default; otherwise, operate as if the **-S** option was not specified.

EXAMPLES
Example 1: Creating a ZFS File System Hierarchy
The following commands create a file system named *pool/home* and a file system named *pool/home/bob*. The mount point */export/home* is set for the parent file system, and is automatically inherited by the child file system.

   # **zfs create** *pool/home*
   # **zfs set mountpoint**=*/export/home pool/home*
   # **zfs create** *pool/home/bob*

Example 2: Disabling and Enabling File System Compression
The following command disables the **compression** property for all file systems under *pool/home*. The next command explicitly enables **compression** for *pool/home/anne*.

   # **zfs set compression**=**off** *pool/home*
   # **zfs set compression**=**on** *pool/home/anne*

Example 3: Setting a Quota on a ZFS File System

The following command sets a quota of 50 Gbytes for *pool/home/bob*:
    # **zfs set quota**=*50G pool/home/bob*

Example 4: Listing ZFS Properties
The following command lists all properties for *pool/home/bob*:

```
# zfs get all pool/home/bob
NAME            PROPERTY        VALUE           SOURCE
pool/home/bob  type            filesystem       -
pool/home/bob  creation        Tue Jul 21 15:53 2009  -
pool/home/bob  used            21K              -
pool/home/bob  available       20.0G            -
pool/home/bob  referenced      21K              -
pool/home/bob  compressratio   1.00x            -
pool/home/bob  mounted         yes              -
pool/home/bob  quota           20G              local
pool/home/bob  reservation     none             default
pool/home/bob  recordsize      128K             default
pool/home/bob  mountpoint      /pool/home/bob   default
pool/home/bob  sharenfs        off              default
pool/home/bob  checksum        on               default
pool/home/bob  compression     on               local
pool/home/bob  atime           on               default
pool/home/bob  devices         on               default
pool/home/bob  exec            on               default
pool/home/bob  setuid          on               default
pool/home/bob  readonly        off              default
pool/home/bob  zoned           off              default
pool/home/bob  snapdir         hidden           default
pool/home/bob  acltype         off              default
pool/home/bob  aclmode         discard          default
pool/home/bob  aclinherit      restricted       default
pool/home/bob  canmount        on               default
pool/home/bob  xattr           on               default
pool/home/bob  copies          1                default
pool/home/bob  version         4                -
pool/home/bob  utf8only        off              -
pool/home/bob  normalization   none             -
pool/home/bob  casesensitivity sensitive        -
pool/home/bob  vscan           off              default
pool/home/bob  nbmand          off              default
```

```
pool/home/bob  sharesmb              off            default
pool/home/bob  refquota              none           default
pool/home/bob  refreservation        none            default
pool/home/bob  primarycache          all            default
pool/home/bob  secondarycache        all             default
pool/home/bob  usedbysnapshots        0             -
pool/home/bob  usedbydataset         21K             -
pool/home/bob  usedbychildren         0             -
pool/home/bob  usedbyrefreservation  0               -
```

The following command gets a single property value:
>   # **zfs get -H -o value compression** *pool/home/bob*
>   on

The following command lists all properties with local settings for *pool/home/bob*:
>   # **zfs get -r -s local -o name**,**property**,**value all** *pool/home/bob*
>   NAME   PROPERTY   VALUE
>   pool/home/bob quota   20G
>   pool/home/bob compression   on

Example 5: Inheriting ZFS Properties
The following command causes *pool/home/bob* and *pool/home/anne* to inherit the **checksum** property
from their parent.
>   # **zfs inherit checksum** *pool/home/bob pool/home/anne*

Example 6: Setting User Properties
The following example sets the user-defined *com.example*:*department* property for a dataset:
>   # **zfs set** *com.example*:*department=12345 tank/accounting*

Example 7: Setting sharenfs Property Options on a ZFS File System
The following commands show how to set **sharenfs** property options to enable read-write access for a set
of IP addresses and to enable root access for system "neo" on the *tank/home* file system:
>   # **zfs set sharenfs**='*rw=*@123.123.0.0/16:[::1],root=*neo*' tank/home

If you are using DNS for host name resolution, specify the fully-qualified hostname.

SEE ALSO
  zfsprops(7), zfs-list(8)