NAME

zdb - display ZFS storage pool debugging and consistency information

SYNOPSIS

zdb [**-AbcdDFGhikLMNPsTvXYy**] [**-e** [**-V**] [**-p** *path*]<?>] [**-I** *inflight-I/O-ops*] [**-o** *var*=*value*]<?>
[**-t** *txg*] [**-U** *cache*] [**-x** *dumpdir*] [**-K** *key*] [*poolname*[/*dataset*|*objset-ID*]] [*object*|*range*<?>]
zdb [**-AdiPv**] [**-e** [**-V**] [**-p** *path*]<?>] [**-U** *cache*] [**-K** *key*] *poolname*[*/dataset*|*objset-ID*] [*object*|*range*<?>]
zdb -B [**-e** [**-V**] [**-p** *path*]<?>] [**-U** *cache*] [**-K** *key*] *poolname/objset-ID* [*backup-flags*]
zdb -C [**-A**] [**-U** *cache*] [*poolname*]
zdb -E [**-A**] *word0*:*word1*:<?>:*word15*
zdb -l [**-Aqu**] *device*
zdb -m [**-AFLPXY**] [**-e** [**-V**] [**-p** *path*]<?>] [**-t** *txg*] [**-U** *cache*] *poolname* [*vdev* [*metaslab*]<?>]
zdb -O [**-K** *key*] *dataset path*
zdb -r [**-K** *key*] *dataset path destination*
zdb -R [**-A**] [**-e** [**-V**] [**-p** *path*]<?>] [**-U** *cache*] *poolname vdev*:*offset*:[*lsize*/]*psize*[:*flags*]
zdb -S [**-AP**] [**-e** [**-V**] [**-p** *path*]<?>] [**-U** *cache*] *poolname*

DESCRIPTION

The **zdb** utility displays information about a ZFS pool useful for debugging and performs some amount of consistency checking. It is a not a general purpose tool and options (and facilities) may change. It is not a fsck(8) utility.

The output of this command in general reflects the on-disk structure of a ZFS pool, and is inherently unstable. The precise output of most invocations is not documented, a knowledge of ZFS internals is assumed.

If the *dataset* argument does not contain any "/" or "**@**" characters, it is interpreted as a pool name. The root dataset can be specified as "*pool*/".

zdb is an "offline" tool; it accesses the block devices underneath the pools directly from userspace and does not care if the pool is imported or datasets are mounted (or even if the system understands ZFS at all). When operating on an imported and active pool it is possible, though unlikely, that zdb may interpret inconsistent pool data and behave erratically.

OPTIONS

Display options:

-b, **--block-stats**

Display statistics regarding the number, size (logical, physical and allocated) and deduplication of blocks.

-B, **--backup**

Generate a backup stream, similar to **zfs send**, but for the numeric objset ID, and without opening the dataset. This can be useful in recovery scenarios if dataset metadata has become corrupted but the dataset itself is readable. The optional *flags* argument is a string of one or more of the letters **e**, **L**, **c**, and **w**, which correspond to the same flags in zfs-send(8).

-c, **--checksum**

Verify the checksum of all metadata blocks while printing block statistics (see **-b**).

If specified multiple times, verify the checksums of all blocks.

-C, **--config**

Display information about the configuration. If specified with no other options, instead display information about the cache file (*/etc/zfs/zpool.cache*). To specify the cache file to display, see **-U**.

If specified multiple times, and a pool name is also specified display both the cached configuration and the on-disk configuration. If specified multiple times with **-e** also display the configuration that would be used were the pool to be imported.

-d, **--datasets**

Display information about datasets. Specified once, displays basic dataset information: ID, create transaction, size, and object count. See **-N** for determining if *poolname*[/*dataset*|*objset-ID*] is to use the specified *dataset*|*objset-ID* as a string (dataset name) or a number (objset ID) when datasets have numeric names.

If specified multiple times provides greater and greater verbosity.

If object IDs or object ID ranges are specified, display information about those specific objects or ranges only.

An object ID range is specified in terms of a colon-separated tuple of the form <start>:<end>[:<flags>]. The fields *start* and *end* are integer object identifiers that denote the upper and lower bounds of the range. An *end* value of -1 specifies a range with no upper bound. The *flags* field optionally specifies a set of flags, described below, that control which object types are dumped. By default, all object types are dumped. A minus sign (-) negates the effect of the flag that follows it and has no effect unless preceded by the *A* flag. For example, the range 0:-1:A-d will dump all object types except for directories.

A Dump all objects (this is the default)

d	Dump ZFS directory objects
f	Dump ZFS plain file objects
m	Dump SPA space map objects
z	Dump ZAP objects
-	Negate the effect of next flag

-D, **--dedup-stats**
Display deduplication statistics, including the deduplication ratio (**dedup**), compression ratio (**compress**), inflation due to the zfs copies property (**copies**), and an overall effective ratio (**dedup** x **compress** / **copies**).

-DD Display a histogram of deduplication statistics, showing the allocated (physically present on disk) and referenced (logically referenced in the pool) block counts and sizes by reference count.

-DDD Display the statistics independently for each deduplication table.

-DDDD
Dump the contents of the deduplication tables describing duplicate blocks.

-DDDDD
Also dump the contents of the deduplication tables describing unique blocks.

-E, **--embedded-block-pointer**=*word0*:*word1*:<?>:*word15*
Decode and display block from an embedded block pointer specified by the *word* arguments.

-h, **--history**
Display pool history similar to **zpool history**, but include internal changes, transaction, and dataset information.

-i, **--intent-logs**
Display information about intent log (ZIL) entries relating to each dataset. If specified multiple times, display counts of each intent log transaction type.

-k, **--checkpointed-state**
Examine the checkpointed state of the pool. Note, the on disk format of the pool is not reverted to the checkpointed state.

-l, **--label**=*device*
Read the vdev labels and L2ARC header from the specified device. **zdb -l** will return 0 if valid

label was found, 1 if error occurred, and 2 if no valid labels were found. The presence of L2ARC header is indicated by a specific sequence (L2ARC_DEV_HDR_MAGIC). If there is an accounting error in the size or the number of L2ARC log blocks **zdb -l** will return 1. Each unique configuration is displayed only once.

-ll *device*

In addition display label space usage stats. If a valid L2ARC header was found also display the properties of log blocks used for restoring L2ARC contents (persistent L2ARC).

-lll *device*

Display every configuration, unique or not. If a valid L2ARC header was found also display the properties of log entries in log blocks used for restoring L2ARC contents (persistent L2ARC).

If the **-q** option is also specified, don't print the labels or the L2ARC header.

If the **-u** option is also specified, also display the uberblocks on this device. Specify multiple times to increase verbosity.

-L, **--disable-leak-tracking**

Disable leak detection and the loading of space maps. By default, **zdb** verifies that all non-free blocks are referenced, which can be very expensive.

-m, **--metaslabs**

Display the offset, spacemap, free space of each metaslab, all the log spacemaps and their obsolete entry statistics.

-mm Also display information about the on-disk free space histogram associated with each metaslab.

-mmm Display the maximum contiguous free space, the in-core free space histogram, and the percentage of free space in each space map.

-mmmm

Display every spacemap record.

-M, **--metaslab-groups**

Display all "normal" vdev metaslab group information - per-vdev metaslab count, fragmentation, and free space histogram, as well as overall pool fragmentation and histogram.

-MM "Special" vdevs are added to -M's normal output.

-O, **--object-lookups**=*dataset path*

Also display information about the maximum contiguous free space and the percentage of free space in each space map.

-MMM Display every spacemap record.

-N Same as **-d** but force zdb to interpret the [*dataset*|*objset-ID*] in [*poolname*[/*dataset*|*objset-ID*]] as a numeric objset ID.

-O *dataset path*

Look up the specified *path* inside of the *dataset* and display its metadata and indirect blocks. Specified *path* must be relative to the root of *dataset*. This option can be combined with **-v** for increasing verbosity.

-r, **--copy-object**=*dataset path destination*

Copy the specified *path* inside of the *dataset* to the specified destination. Specified *path* must be relative to the root of *dataset*. This option can be combined with **-v** for increasing verbosity.

-R, **--read-block**=*poolname vdev*:*offset*:[*lsize*/]*psize*[:*flags*]

Read and display a block from the specified device. By default the block is displayed as a hex dump, but see the description of the **r** flag, below.

The block is specified in terms of a colon-separated tuple *vdev* (an integer vdev identifier) *offset* (the offset within the vdev) *size* (the physical size, or logical size / physical size) of the block to read and, optionally, *flags* (a set of flags, described below).

b *offset* Print block pointer at hex offset

c Calculate and display checksums

d Decompress the block. Set environment variable **ZDB_NO_ZLE** to skip zle when guessing.

e Byte swap the block

g Dump gang block header

i Dump indirect block

r Dump raw uninterpreted block data

v Verbose output for guessing compression algorithm

-s, **--io-stats**

Report statistics on **zdb** I/O. Display operation counts, bandwidth, and error counts of I/O to the pool from **zdb**.

-S, **--simulate-dedup**
Simulate the effects of deduplication, constructing a DDT and then display that DDT as with **-DD**.

-T, **--brt-stats**
Display block reference table (BRT) statistics, including the size of uniques blocks cloned, the space saving as a result of cloning, and the saving ratio.

-TT Display the per-vdev BRT statistics, including total references.

-TTT Dump the contents of the block reference tables.

-u, **--uberblock**
Display the current uberblock.

Other options:

-A, **--ignore-assertions**
Do not abort should any assertion fail.

-AA Enable panic recovery, certain errors which would otherwise be fatal are demoted to warnings.

-AAA Do not abort if asserts fail and also enable panic recovery.

-e, **--exported**=[**-p** *path*]<?>
Operate on an exported pool, not present in */etc/zfs/zpool.cache*. The **-p** flag specifies the path under which devices are to be searched.

-x, **--dump-blocks**=*dumpdir*
All blocks accessed will be copied to files in the specified directory. The blocks will be placed in sparse files whose name is the same as that of the file or device read. **zdb** can be then run on the generated files. Note that the **-bbc** flags are sufficient to access (and thus copy) all metadata on the pool.

-F, **--automatic-rewind**
Attempt to make an unreadable pool readable by trying progressively older transactions.

-G, **--dump-debug-msg**
Dump the contents of the zfs_dbgmsg buffer before exiting **zdb**. zfs_dbgmsg is a buffer used by ZFS to dump advanced debug information.

-I, **--inflight**=*inflight-I/O-ops*

Limit the number of outstanding checksum I/O operations to the specified value. The default value is 200. This option affects the performance of the **-c** option.

-K, **--key**=*key*

Decryption key needed to access an encrypted dataset. This will cause **zdb** to attempt to unlock the dataset using the encryption root, key format and other encryption parameters on the given dataset. **zdb** can still inspect pool and dataset structures on encrypted datasets without unlocking them, but will not be able to access file names and attributes and object contents. **WARNING:** The raw decryption key and any decrypted data will be in user memory while **zdb** is running. Other user programs may be able to extract it by inspecting **zdb** as it runs. Exercise extreme caution when using this option in shared or uncontrolled environments.

-o, **--option**=*var*=*value*<?>

Set the given global libzpool variable to the provided value. The value must be an unsigned 32-bit integer. Currently only little-endian systems are supported to avoid accidentally setting the high 32 bits of 64-bit variables.

-P, **--parseable**

Print numbers in an unscaled form more amenable to parsing, e.g. **1000000** rather than **1M**.

-t, **--txg**=*transaction*

Specify the highest transaction to use when searching for uberblocks. See also the **-u** and **-l** options for a means to see the available uberblocks and their associated transaction numbers.

-U, **--cachefile**=*cachefile*

Use a cache file other than */etc/zfs/zpool.cache*.

-v, **--verbose**

Enable verbosity. Specify multiple times for increased verbosity.

-V, **--verbatim**

Attempt verbatim import. This mimics the behavior of the kernel when loading a pool from a cachefile. Only usable with **-e**.

-X, **--extreme-rewind**

Attempt "extreme" transaction rewind, that is attempt the same recovery as **-F** but read transactions otherwise deemed too old.

-Y, **--all-reconstruction**

Attempt all possible combinations when reconstructing indirect split blocks. This flag disables the individual I/O deadman timer in order to allow as much time as required for the attempted reconstruction.

-y, **--livelist**

Perform validation for livelists that are being deleted. Scans through the livelist and metaslabs, checking for duplicate entries and compares the two, checking for potential double frees. If it encounters issues, warnings will be printed, but the command will not necessarily fail.

Specifying a display option more than once enables verbosity for only that option, with more occurrences enabling more verbosity.

If no options are specified, all information about the named pool will be displayed at default verbosity.

EXAMPLES

Example 1: Display the configuration of imported pool *rpool*

```
# zdb -C rpool
MOS Configuration:
        version: 28
        name: ’rpool’
 <?>
```

Example 2: Display basic dataset information about *rpool*

```
# zdb -d rpool
Dataset mos [META], ID 0, cr_txg 4, 26.9M, 1051 objects
Dataset rpool/swap [ZVOL], ID 59, cr_txg 356, 486M, 2 objects
 <?>
```

Example 3: Display basic information about object 0 in *rpool/export/home*

```
# zdb -d rpool/export/home 0
Dataset rpool/export/home [ZPL], ID 137, cr_txg 1546, 32K, 8 objects

    Object  lvl   iblk   dblk  dsize  lsize   %full  type
         0    7    16K    16K  15.0K    16K   25.00  DMU dnode
```

Example 4: Display the predicted effect of enabling deduplication on *rpool*

```
# zdb -S rpool
Simulated DDT histogram:

bucket              allocated                       referenced
```

```
_______   ______________________________   ______________________________
refcnt   blocks   LSIZE   PSIZE   DSIZE   blocks   LSIZE   PSIZE   DSIZE
------   ------   -----   -----   -----   ------   -----   -----   -----
     1     694K   27.1G   15.0G   15.0G     694K   27.1G   15.0G   15.0G
     2    35.0K   1.33G    699M    699M    74.7K   2.79G   1.45G   1.45G
 <?>
dedup = 1.11, compress = 1.80, copies = 1.00, dedup * compress / copies = 2.00
```

SEE ALSO

zfs(8), zpool(8)