ConvexOS dump and restore

Order No. DSW-392

First Edition
October 1991

Field Test

This document is approved for use at Field Test sites and for limited customer distribution with program manager approval. The information contained is for verification under actual working conditions.

CONVEX COMPUTER CORPORATION

ConvexOS `dump` and `restore` Quick Reference

Order No. DSW-392
First Edition
October 1991

Printed in the United States of America.

Contents

Preface

This reference describes how to use ConvexOS `dump` and `restore` commands. These commands can be used, to a limited extent, without root privileges, but most applications require them.

Command descriptions and examples are divided into sections according to function. To find the task you want to perform, find the appropriate section in "Contents," page iii. Look through the appropriate section for the command syntax and description you need.

Using command descriptions

This reference uses these conventions in describing command syntax:

- If a command description instructs to *enter* a command, you need to press the **RETURN** key after typing the command.

  If a command description instructs to *type* a command, do *not* press the **RETURN** key after typing the command.

- Different fonts indicate how to use different parts of commands and utilities:

`Courier`	Signifies command names, system output, and error messages.
`Courier bold`	Type exactly as shown.
Italic	Substitute with the appropriate value.
KEYCAP	Press the indicated key.

- Some special conventions are used to convey special conditions:

  CTRL The control key should be held down while pressing another key associated with it in a command syntax description. For example, in the command syntax description:

  CTRL-d

  you need to press the **CTRL** key and type **d** simultaneously.

[ ] Brackets indicate an optional part of a command description:

`restore x` [*filenames...*]

where *filenames* is optional. The brackets are not typed.

... A horizontal ellipsis indicates part of a syntax can repeat. For example:

`restore x` [*filenames...*]

indicates that values for *filenames* can be repeated.

⋮ A vertical ellipsis indicates part of a description or example has been omitted.

Using example descriptions

With some command descriptions are command examples. Command examples are in the following format:

Example: **`/etc/rdump f serv:/dev/rmt10 \`**
`/dev/rda0g`

Only the terms written in **`courier bold`** are to be entered as shown.

The backslash, which should not be typed, indicates that the example is continued on the next line. Do not press **RETURN** until the entire example is typed.

Finding further information

To find information on the following topics, consult the associated documentation:

Topic	Associated documentation
Dumping and restoring file systems	*Managing ConvexOS: Operations Guide* (DSW-031)
Configuring the ConvexOS tape system	*Manging ConvexOS: Configuration Guide* (DSW-030)
Using the ConvexOS tape system	*ConvexOS Tape System User's Guide* (DSW-018) and *ConvexOS Tape System Quick Reference*, (DSW-391)

Dumping file systems

The dump commands described in the following sections copy files and directories to magnetic tape. Most applications using these commands require superuser authorization.

Identifying terms in dump syntax descriptions

The syntax descriptions that follow use these terms:

level — Single-digit integer [0-9] representing the level of the file system dump according to the following chart:

- 0 Dump the entire file system. Referred to as a *full dump*.
- 1 Dump all files touched since the last level-0 dump.
- 2 Dump all files touched since the last level-1 dump.
- 3 Dump all files touched since the last level-2 dump.

  ⋮

- 9 Dump all files touched since the last level-8 dump.

keys — Set of options for executing dump commands. *keys* consists of the options described in the section, "Using keys and arguments," on page 3.

args — Set of arguments corresponding to the keys that are used. Arguments that must be used with certain keys are described in the section, "Using keys and arguments," on page 3.

target — The file system to be dumped, as shown in the file /etc/fstab.

Syntax descriptions for `dump` commands

dump [*level*][*keys*] [*args*] *target*

Copy files in a specified file system that have changed since a specified day, or an entire file system, to magnetic tape.

`dump` is not a fast as `xdump`, but recovers better from tape system errors.

vdump [*level*][*keys*] [*args*] *target*

Copy files in a specified file system that have changed since a specified day, or an entire file system, to magnetic tape. Upon completion execute `restore -V`, which verifies the format of the `dump` tape to make sure that it is readable.

xdump [*level*][*keys*] [*args*] *target*

Copy files in a specified file system that have changed since a specified day, or an entire file system, to magnetic tape.

`xdump` is faster than `dump`, but does not recover from tape system errors as well as `dump` and cannot be used with labeled tapes.

rdump [*level*][*keys*] [*args*] *target*

Copy files from a specified file system on a remote machine that have changed since a specified day, or an entire file system on a remote machine, to magnetic tape on a tape drive on the local machine.

Using keys and arguments

a Skip any files that have changed between the time that the dump was started and the time when dump actually copies the file. dump does not skip directories. dump displays a short message with the inode number of any files that are skipped.

Recommended for dumping a mounted file system because it prevents invalid files from being written to the dump tape. This option is not valid with xdump.

f *destination*

Dump *target* to the place specified by *destination*, instead of on the dump tape.

If using dump, vdump, or xdump, *destination* is the name of the file the dump is to be written to. To write to standard output, specify a dash (-).

If using rdump, *destination* is *machine*:*device*, where *machine* is the remote machine from which a directory tree is dumped, and *device* is the tape device used for the dump.

In either case, the default for *destination* is /dev/rmt8.

Examples:
```
/etc/dump f /dev/rmt8 /dev/rda0g
/etc/rdump f serv:/dev/rmt8 \
/dev/rda0g
```

u If the dump completes successfully, write the date of the beginning of the dump in /etc/dumpdates.

s *size*

Specify the size of the dump tape in feet. When the specified size is reached, dump will wait for reels to be changed. The default tape size is 2300 feet.

Examples:
```
/etc/dump s 2100 /dev/rda0g
/etc/rdump fs serv:/dev/rmt8 \
100 /dev/rda0g
```

d *density*

Set tape density. *density* is expressed in bytes-per-inch, and is taken from the argument list. This is used in calculating the amount of tape used per reel. The default is 1600 bpi.

Examples:
```
/etc/dump d 6250 /dev/rda0g
/etc/rdump fd serv:/dev/rmt8 \
6250 /dev/rda0g
```

W Display the most recent dump dates in the file /etc/dumpdates for all the file systems and indicate those that should be dumped (according to /etc/fstab). If the W option is set, all other options are ignored.

w Display the file systems that need to be dumped (according to /etc/fstab). If the w option is set, all other options are ignored.

n Whenever dump requires operator attention, notify by means similar to the `wall` utility all operators in the group "operator" who are logged in.

b *factor*
Set blocking factor for tape records. *factor* is an integer that indicates how many kbytes are in a tape record and is taken from the argument list. The default is 5.

Examples: **`/etc/dump 0b 10 /dev/rda0g`**

`/etc/rdump 5fb serv:/dev/rmt8 \`
`64 /dev/rda0g`

G Set defaults for a GCR (6250 bpi) dump. This sets the following:

- Tape device to /dev/rmt16
- Tape density to 6250 bpi
- Blocking factor to 50

I Put the tape drive into 100 ips streaming mode. The default is 50 ips.

E Write to tape until finished or until end-of-tape is reached. If end-of-tape is reached, the operator is prompted to change tapes. Even with this option, dump displays an estimate of how many 9-track tapes (not cartridges) are needed for the dump.

rdump may not support this option unless both machines are CONVEX running ConvexOS V9.0 or greater.

Restoring files and file systems

Syntax for restoring file systems non-interactively

To restore parts of a file system or an entire file system from a dump tape, or tape set, to a disk, use the following syntax:

restore [*options*] [*args*] [*filelist*]

where the following terms are used:

options	Set of options described in the sections that follow.
args	Set of arguments corresponding to the keys that are used. Arguments that must be used with certain keys are described in the sections that follow.
filelist	List of files or directories to be restored. This is used only if the x key is specified.

Options for restoring file systems non-interactively

b *factor*
Set blocking factor for tape records. *factor* indicates how many kbytes are in a tape record. The default is 5.

Example: **restore xb 10**

c Convert old directories to new directory format.

d Display debugging information on the controlling terminal.

f *device*
Specify tape device. *device* is the full path name of your tape device. If *device* is "-", `restore` reads from standard input. Thus, dump and `restore` can be used in a pipeline to dump and `restore` a file system:

Example:
```
dump 0f - /usr | ( cd /mnt;      \
restore xf - )
```

G Set defaults for a GCR (6250 bpi) restoration. The tape device is set to /dev/rmt16, the density to 6250 bpi, and the tape blocking factor to 50.

h Restore directory name, but not the files that are in it. This prevents hierarchical restoration of complete subtrees from the tape.

I Put the tape drive into 100 ips streaming mode. The default is 50 ips.

m Restore by inode number rather than by file name. This is useful only if a few files are being extracted, and you want to avoid regenerating the complete path name to the file.

s *n*
Skip to the *n*th file on the tape before beginning restoration.

Example: **restore xs 3**

t [*filelist*]
Display table of contents. The names of files or directories specified in *filelist* are listed if they are on the tape.

If *filelist* is not specified, then all the files and directories on the dump tape are listed.

Example: **restore t file1 file2 dir1**

v Verbose mode. Display the type and name of each file as it is restored.

V Verify the dump tape can be read without tape errors, and the dump file format is correct. Correctness of the directories and files contained on the dump tape is not verified.

x [*filelist*]
Extract the files and directories specified by *filelist*. Subdirectories of directories in *filelist* are also recursively restored.

The owner, modification time, and mode are restored, if possible.

If *filelist* is not specified the entire contents of the tape is extracted.

Example: **restore x file1 file2 dir/file3**

y Continue upon encountering a tape error. `restore` attempts to skip the bad tape block or blocks and continues.

Syntax for restoring file systems interactively

To restore parts of a file system or an entire file system from a dump tape, or tape set, to a disk, use the following syntax:

restore i

Specify interactive restoration of files or directories from a dump tape. This option can be used with other keys as specified in the section, "Options for restoring file systems non-interactively," on page 5.

restore provides a shell-like interface from which you can move through the directory tree stored on the dump tape, selecting files and directories to be extracted.

restore interactive mode commands

This section describes commands that can only be used with restore while in interactive mode.

For those commands that accept arguments, the default is always your current directory within the interface.

add [*args*]

The directories or files specified by *filelist* are added to the list of files to be extracted.

If a directory is specified, then it and all of its descendants (subdirectories and files) are added to the extraction list, unless the h key was specified with the i key when restore was invoked.

Files on the extraction list are prepended with an asterisk (*) when listed with the ls command.

cd *dir*

Change the current working directory within the restore interface to the directory specified by *dir*.

delete [*filelist*]

The directories or files specified by *filelist* are deleted from the list of files to be extracted.

If a directory is specified, then it and all of its descendants (subdirectories and files) are deleted from the extraction list, unless the h key was specified with the i key when restore was invoked.

extract
All of the files that are on the extraction list are extracted from the dump tape. `restore` requests the volume of the tape set you wish to mount.

The fastest way to extract a few files is to start with the last volume, and work towards the first volume.

verbose
Toggle the v key. When the `verbose` key is set, the `ls` command lists the inode numbers and names of all entries and displays information about each file or directory as it is extracted.

help
List a summary of the available commands.

ls [*dir*]
List the directory specified by *dir*. Directory entries in the listing are appended with a virgule (/).

Entries that have been marked for extraction are prepended with an asterisk (*).

If the `verbose` key is set, the inode number of each entry is listed along with the entry name.

pwd
Display the full path name of the current working directory in the `restore` interface.

quit
Exit from `restore`, even if the extraction list is not empty.

Interpreting error messages

filename: `not found on tape`
: *filename* was listed in the tape directory, but was not found on the tape. This is caused by tape-read errors while looking for the file, and from using a `dump` tape created on an active file system.

`Converting to new file system format`
: A `dump` tape created from an old file system has been loaded. It is automatically converted to the new file system format.

`expected next file` *inumber* `got` *inumber2*
: A file, *inumber2*, that was not listed in the directory showed up instead of the expected file, *inumber*. This can occur when using a `dump` tape created on an active file system.

`Incremental tape too low`
: When doing an incremental restore, a tape that was written before the previous incremental tape, or that has too low an incremental level, has been loaded.

`Incremental tape too high`
: When doing an incremental restore, a tape that does not begin its coverage where the previous incremental tape left off, or that has too high an incremental level, has been loaded.

`resync restore, skipped` *num* `blocks`
: After a tape read error, `restore` may have to resynchronize itself. This message lists the number of blocks, *num*, that were skipped over.

`Tape read error while restoring` *filename*
: A tape read error has occurred while restoring *filename*. The contents of *filename* are probably partially wrong.

`Tape read error while skipping over inode` \ *inumber*

`Tape read error while trying to` \ `resynchronize`
: If an inode is being skipped (*inumber*), or the tape system is trying to resynchronize, then no extracted files have been corrupted, even though files may not be found on the tape.

Document number: 710-017030-000